PGP Desktop Security

for Mac OS

User’s Guide

Version 6.5

PGP*, Version 6.5.1

06-99. Printed in the United States of America.

Portions of this software may use public key algorithms described in U.S. Patent numbers 4,200,770, 4,218,582, 4,405,829, and 4,424,414, licensed exclusively by Public Key Partners; the IDEA(tm) cryptographic cipher described in U.S. patent number 5,214,703, licensed from Ascom Tech AG; and the Northern Telecom Ltd., CAST Encryption Algorithm, licensed from Northern Telecom, Ltd. IDEA is a trademark of Ascom Tech AG. Network Associates Inc. may have patents and/or pending patent applications covering subject matter in this software or its documentation; the furnishing of this software or documentation does not give you any license to these patents. The compression code in PGP is by Mark Adler and Jean-Loup Gailly, used with permission from the free Info-ZIP implementation. LDAP software provided courtesy University of Michigan at Ann Arbor, Copyright © 1992-1996 Regents of the University of Michigan. All rights reserved. This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/). Copyright © 1995-1999 The Apache Group. All rights reserved. See text files included with the software or the PGP web site for further information. This software is based in part on the work of the Independent JPEG Group. Soft TEMPEST font courtesy of Ross Anderson and Marcus Kuhn. Biometric word list for fingerprint verification courtesy of Patrick Juola.

The software provided with this documentation is licensed to you for your individual use under the terms of the End User License Agreement and Limited Warranty provided with the software. The information in this document is subject to change without notice. Network Associates Inc. does not warrant that the information meets you requirements or that the information is free of errors. The information may include technical inaccuracies or typographical errors. Changes may be made to the information and incorporated in new editions of this document, if and when made available by Network Associates Inc.

Export of this software and documentation may be subject to compliance with the rules and regulations promulgated from time to time by the Bureau of Export Administration, United States Department of Commerce, which restrict the export and re-export of certain products and technical data.

Network Associates, Inc.
3965 Freedom Circle
Santa Clara, CA 95054

(408) 988-3832 main
(408) 970-9727 fax
http://www.nai.com
info@nai.com

***** is sometimes used instead of the ® for registered trademarks to protect marks registered outside of the U.S.

LIMITED WARRANTY

Limited Warranty. Network Associates Inc. warrants that the Software Product will perform substantially in accordance with the accompanying written materials for a period of sixty (60) days from the date of original purchase. To the extent allowed by applicable law, implied warranties on the Software Product, if any, are limited to such sixty (60) day period. Some jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

Customer Remedies. Network Associates Inc's and its suppliers' entire liability and your exclusive remedy shall be, at Network Associates Inc's option, either (a) return of the purchase price paid for the license, if any or (b) repair or replacement of the Software Product that does not meet Network Associates Inc's limited warranty and which is returned at your expense to Network Associates Inc. with a copy of your receipt. This limited warranty is void if failure of the Software Product has resulted from accident, abuse, or misapplication. Any repaired or replacement Software Product will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. Outside the United States, neither these remedies nor any product support services offered by Network Associates Inc. are available without proof of purchase from an authorized international source and may not be available from Network Associates Inc. to the extent they subject to restrictions under U.S. export control laws and regulations.

NO OTHER WARRANTIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, AND EXCEPT FOR THE LIMITED WARRANTIES SET FORTH HEREIN, THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” AND NETWORK ASSOCIATES, INC. AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, CONFORMANCE WITH DESCRIPTION, TITLE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM JURISDICTION TO JURISDICTION.

LIMITATION OF LIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NETWORK ASSOCIATES, INC. OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR EXEMPLARY DAMAGES OR LOST PROFITS WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE PRODUCT OR THE FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF NETWORK ASSOCIATES, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, NETWORK ASSOCIATES, INC'S CUMULATIVE AND ENTIRE LIABILITY TO YOU OR ANY OTHER PARTY FOR ANY LOSS OR DAMAGES RESULTING FROM ANY CLAIMS, DEMANDS OR ACTIONS ARISING OUT OF OR RELATING TO THIS AGREEMENT SHALL NOT EXCEED THE PURCHASE PRICE PAID FOR THIS LICENSE. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Table of Contents

Preface

PGP is part of your organization's security toolkit for protecting one of your most important assets: *information*. Corporations have traditionally put locks on their doors and file cabinets and require employees to show identification to prove that they are permitted access into various parts of the business site. PGP is a valuable tool to help you protect the security and integrity of your organization's data and messages. For many companies, loss of confidentiality means loss of business.

Entire books have been written on the subject of implementing network security. The focus of this guide is on implementing PGP as a tool within your overall network security structure. PGP is merely one piece of an overall security system, but it is an extremely important one. PGP provides encryption, which protects data from the eyes of anyone for whom it was not intended, even those who can see the encrypted data. This protects information from both internal and external "outsiders."

This guide describes how to use PGP® Desktop Security for the Macintosh. PGP Desktop Security has many new features, which are described in "What's new in PGP version 6.5.1" on page 14

If you are new to cryptography and would like an overview of the terminology and concepts you will encounter while using PGP, see *An Introduction to Cryptography*.

What’s new in PGP version 6.5.1

This version of PGP includes these new features:

- **PGPnet.** PGPnet is a landmark product in the history of PGP. PGPnet secures all TCP/IP communications between itself and any other machine running PGPnet. It is also fully interoperable with the Gauntlet GVPN firewall/gateway providing a complete solution for corporate remote access VPNs using the industry standard IPSec (Internet Protocol Security) and IKE (Internet Key Exchange) protocols. PGPnet has also been successfully tested with Cisco routers (requires Cisco IOS 12.0(5) or later with IPSec TripleDes Feature Pack), Linux FreeS/WAN 1.0, and many others. PGPnet is also the first IPSec product to fully support the use of OpenPGP keys for authentication in addition to X.509. Refer to Chapter 8, “PGPnet Virtual Private Networking,” for more information and instructions on using PGPnet.

- **Self-Decrypting Archives.** PGP can now encrypt files or folders into Self-Decrypting Archives (SDA) which can be sent to users who do not even have PGP. The archives are completely independent of any application, compressed and protected by PGP's strong cryptography. Using this feature without a passphrase will also allow you to create compact Self-Extracting Archives (SEA) which are not encrypted. The resulting archives run on both PowerPC and 68K Macs, and are encrypted using the CAST algorithm.

- **X.509 Certificate and CA Support.** PGP is now able to interoperate with the X.509 certificate format. This is the format used by most web browsers for securing the transfer of web pages. PGP supports the request of certificates from Network Associates' Net Tools PKI, VeriSign's OnSite, and Entrust certificate authorities. X.509 certificates are analogous to a PGP signature, so you can even request X.509 certificates on your existing PGP key. Using PGPnet, this feature can be used to interoperate with existing VPN solutions based on X.509.

- **Automated Freespace Wiping.** PGP's Freespace Wipe feature now allows you to use AppleScript to automate wiping of the freespace on your disks. The AppleScript dictionary for this is located in PGPtools .

- **PGPmenu Improvements.** PGPmenu is enhanced with many new features. Configurable Command Key support allows you to invoke the **Encrypt/Sign/Encrypt&Sign/Decrypt&Verify** commands in third-party applications without even touching the mouse. The **Empty Trash** command in the Finder can now be turned into a **Wipe Trash** command to ensure that everything you throw away gets securely wiped. The cursor now provides animated progress during PGPmenu operations, and more.

- **Outlook Express support and Enhanced Email Integration.** As part of the new PGPmenu, Outlook Express and Claris Emailer are now recognized as special applications in which PGPmenu can automatically grab the recipient email addresses whenever you invoke PGPmenu on a new email message window. This enhancement cuts out the step of specifying the recipient keys. The old Claris Emailer plug-in has been removed now that PGPmenu directly supports it.
- **Fingerprint word list.** When verifying a PGP public key fingerprint, you can now choose to view the fingerprint as a word list instead of hexadecimal characters. The word list in the fingerprint text box is made up of special authentication words that PGP uses and are carefully selected to be phonetically distinct and easy to understand without phonetic ambiguity.
- **HTTP Proxy Support.** If your Macintosh is behind a corporate firewall with an HTTP proxy server, PGP now supports accessing HTTP certificate servers through the proxy. To use this feature, you must configure the proxy server address in the Internet control panel. This feature requires the installation of Internet Config for users not running Mac OS 8.5 or greater.
- **Smart Word Wrapping.** The word wrapping in PGP now automatically rewraps paragraphs and even quoted paragraphs resulting in much cleaner signed messages.

How to contact Network Associates

Customer service

To order products or obtain product information, contact the Network Associates Customer Care department at (408) 988-3832 or write to the following address:

Network Associates, Inc.
McCandless Towers
3965 Freedom Circle
Santa Clara, CA 95054-1203
U.S.A.

Technical support

Network Associates is famous for its dedication to customer satisfaction. We have continued this tradition by making our site on the World Wide Web a valuable resource for answers to technical support issues. We encourage you to make this your first stop for answers to frequently asked questions, for updates to Network Associates software, and for access to Network Associates news and encryption information.

World Wide Web	http://www.nai.com

Technical Support for your PGP product is also available through these channels:

Phone	(408) 988-3832
Email	PGPSupport@pgp.com

To provide the answers you need quickly and efficiently, the Network Associates technical support staff needs some information about your computer and your software. Please have this information ready before you call:

If the automated services do not have the answers you need, contact Network Associates at one of the following numbers Monday through Friday between 6:00 A.M. and 6:00 P.M. Pacific time.

Phone	(408) 988-3832

To provide the answers you need quickly and efficiently, the Network Associates technical support staff needs some information about your computer and your software. Please have this information ready before you call:

- Product name and version number
- Computer brand and model
- Any additional hardware or peripherals connected to your computer
- Operating system type and version numbers
- Network type and version, if applicable
- Content of any status or error message displayed on screen, or appearing in a log file (not all products produce log files)
- Email application and version (if the problem involves using PGP with an email product, for example, the Eudora plug-in)

- Specific steps to reproduce the problem

Year 2000 compliance

Information regarding NAI products that are Year 2000 compliant and its Year 2000 standards and testing models may be obtained from NAI's Web site at http://www.nai.com/y2k.

For further information, email y2k@nai.com.

Network Associates training

For information about scheduling on-site training for any Network Associates product, call (800) 338-8754.

Comments and feedback

Network Associates appreciates your comments and feedback, but incurs no obligation to you for information you submit. Please address your comments about PGP product documentation to: Network Associates, Inc., 3965 Freedom Circle Santa Clara, CA 95054-1203 U.S.A.. You can also e-mail comments to tns_documentation@nai.com.

Recommended Readings

Non-Technical and beginning technical books

- Whitfield Diffie and Susan Eva Landau, “Privacy on the Line,” *MIT Press*; ISBN: 0262041677
  This book is a discussion of the history and policy surrounding cryptography and communications security. It is an excellent read, even for beginners and non-technical people, but with information that even a lot of experts don't know.

- David Kahn, “The Codebreakers” *Scribner*; ISBN: 0684831309
  This book is a history of codes and code breakers from the time of the Egyptians to the end of WWII. Kahn first wrote it in the sixties, and there is a revised edition published in 1996. This book won't teach you anything about how cryptography is done, but it has been the inspiration of the whole modern generation of cryptographers.

- Charlie Kaufman, Radia Perlman, and Mike Spencer, “Network Security: Private Communication in a Public World,” *Prentice Hall;* ISBN: 0-13-061466-1
  This is a good description of network security systems and protocols, including descriptions of what works, what doesn't work, and why. Published in 1995, so it doesn't have many of the latest advances, but is still a good book. It also contains one of the most clear descriptions of how DES works of any book written.

Intermediate books

- Bruce Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C,” *John Wiley & Sons*; ISBN: 0-471-12845-7
  This is a good beginning technical book on how a lot of cryptography works. If you want to become an expert, this is the place to start.

- Alfred J. Menezes, Paul C. van Oorschot, and Scott Vanstone, “Handbook of Applied Cryptography,” *CRC Press;* ISBN: 0-8493-8523-7
  This is the technical book you should get after Schneier. There is a lot of heavy-duty math in this book, but it is nonetheless usable for those who do not understand the math.

- Richard E. Smith, “Internet Cryptography,” *Addison-Wesley Pub Co*; ISBN: 020192480
  This book describes how many Internet security protocols. Most importantly, it describes how systems that are designed well nonetheless end up with flaws through careless operation. This book is light on math, and heavy on practical information.

- William R. Cheswick and Steven M. Bellovin, “Firewalls and Internet Security: Repelling the Wily Hacker” *Addison-Wesley Pub Co*; ISBN: 0201633574
  This book is written by two senior researcher at AT&T Bell Labs, about their experiences maintaining and redesigning AT&T's Internet connection. Very readable.

Advanced books

- Neal Koblitz, “A Course in Number Theory and Cryptography” *Springer-Verlag*; ISBN: 0-387-94293-9
  An excellent graduate-level mathematics textbook on number theory and cryptography.

- Eli Biham and Adi Shamir, “Differential Cryptanalysis of the Data Encryption Standard,” *Springer-Verlag*; ISBN: 0-387-97930-1
  This book describes the technique of differential cryptanalysis as applied to DES. It is an excellent book for learning about this technique.

Installing PGP 1

This chapter describes how to install and run PGP Desktop Security for Macintosh software. This chapter also provides a quick overview of the procedures you will normally follow in using the product.

Before you begin installing PGP be sure to review the system requirements outlined below.

System requirements

To install PGP on a Macintosh system, you must have:

- Power Macintosh (PowerPC processor required)
- Mac OS 7.6.1 or later
- Open Transport 1.3 or later
- 16 MB RAM
- 10 MB hard disk space

If you plan to run PGPnet on the system, you must also have:

- Compatible LAN / WAN hardware and software (refer to the PGP What's New file for more information about PGPnet compatible software and hardware)

Compatibility with other versions

PGP has gone through many revisions since it was released by Phil Zimmermann as a freeware product in 1991. Although this version of PGP represents a significant rewrite of the original program and incorporates a completely new user interface, it has been designed to be compatible with earlier versions of PGP. This means that you can exchange secure email with people who are still using these older versions of the product:

- PGP 2.6 (Distributed by MIT)
- PGP for Personal Privacy, Version 5.0 - 5.5
- PGP for Business Security or PGP for Email and Files Version 5.5
- PGP Desktop Security or PGP for Personal Privacy Version 6.0

NOTE: PGP desktop products that are version 5.0 and later may require the RSA add-on for backward compatibility.

Upgrading from a previous version

If you are upgrading from a previous version of PGP (from PGP, Inc., Network Associates, Inc. or ViaCrypt), you may want to remove the old program files before installing PGP to free up some disk space. However, you should be careful not to delete the private and public keyring files used to store any keys you have created or collected while using the previous version. When you install PGP, you are given the option of retaining your existing private and public keyrings, so you don't have to go to the trouble of importing all of your old keys. To upgrade from a previous version, follow the appropriate steps listed next.

To upgrade from PGP Version 2.6.2 or 2.7.1

1. Exit all programs or open applications.
2. Make backups of your old PGP keyrings on another volume. In PGP for Windows versions 2.6.2 and 2.7.1, your public keys are stored in "pubring.pgp" and your private keys are stored in "secring.pgp". In versions 5.x - 6.5, your public keys are stored in "pubring.pkr" and your private keys are stored in "secring.skr".

TIP: Make two separate backups of your keyrings onto two different floppy disks just to be safe. Be especially careful not to lose your private keyring; otherwise you will never be able to decrypt any email messages or file attachments encrypted with the lost keys. Store the keyrings in a secure place where only you have access to them.

3. When you have successfully backed up your old keyrings, remove or archive the (old) PGP software. You have two options here:
   - Manually delete the entire old PGP folder and all of its contents; or
   - Manually delete the old PGP program and archive the remaining files, especially the configuration and keyring files**.**
4. Install PGP version 6.5.1 using the provided installer.
5. Restart your computer.

To upgrade from PGP Version 5.x

If you are upgrading from PGP version 4.x or 5.x, follow the installation instructions outlined in "Installing PGP" below.

Installing PGP

You can install the PGP Desktop Security software from a CD-ROM or from your company file server. The Installer program ( ) automatically extracts and steps you through the installationThe self-extracting file, Setup.exe, automatically extracts and steps you through the installation. After you install the software, you can create your private and public key pair and begin using PGP. Refer to the PGPMacUsersGuide.pdf file included with the program for instructions on using PGP.

To install PGP Desktop Security for MacintoshWindows systems, carefully follow the steps outlined below.

To install PGP

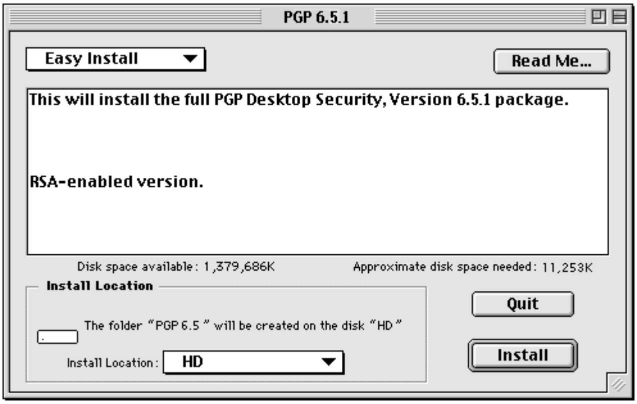
1. Quit all applications running on your computer.
2. **To install from a CD-ROM,** insert it into the CD-ROM drive.

   To install from your company file server, contact your security officer for information about the server from which to download PGP. Log on to the server.
3. Double-click the installation icon ( ) to start the Installer program.

   The Network Associates license agreement appears.
4. Review the license agreement information, then click **Accept** to continue the installation.

   The PGP Release Notes appear.
5. Review the release notes for known issues and export restrictions, then click **Continue**.

The installation screen appears, as shown in Figure 1-1.



This will install the full PGP Desktop Security, Version 6.5.1 package.

RSA-enabled version.

Disk space available: 1,379,686K

Approximate disk space needed: 11,253K

Install Location

The folder "PGP 6.5" will be created on the disk "HD"

Install Location: HD

Figure 1-1. PGP Installation screen

6. Select a type of installation:

   - **Easy Install.** Choose **Easy Install** to perform a full installation of PGP.

   - **Custom Install.** Choose **Custom Install** to install PGP with user-definable options. You are prompted to choose the components that you want to install.

7. Your installation options are:

   - **PGPkeys.** This item installs the PGP program. You must install the Key Management utilities.

   - **PGPtools.** Select this option if you want to install the PGPtools component. PGPtools is a toolbar that allows you to perform PGP functions from within other applications. You can encrypt and sign, decrypt and verify, or securely wipe messages and files directly from PGPtools.

   - **PGPdisk.** Select this option to install the PGPdisk program. PGPdisk is an easy-to-use encryption application that enables you to set aside an area of disk space for storing your sensitive data.

   - **PGPnet.** Select this option to install the PGPnet program. PGPnet, a *Virtual Private Network (VPN)*, is an easy-to-use encryption application that allows you to communicate securely and economically with other PGPnet users on your own corporate intranet and with users throughout the world.

- **PGPmenu.** Select this option to install the PGPmenu component. PGPmenu allows you to performs most PGP functions from the Finder or from within most applications.
- **PGPcontextmenu for MacOS.** Select this option if you want to install the PGPcontextmenu component. The PGPcontextmenu allows you to perform encryption task by holding down the CONTROL KEY and clicking on a file or volume in a window or on your desktop. This feature is available on machines running Mac OS 8 or later and is equivalent to launching and using the features within PGPtools.
- **PGP Qualcomm Eudora Plug-in.** Select this option if you want to integrate PGP functionality with your Qualcomm Eudora email program. PGP version 6.5.1 supports Eudora versions 3.05 and later.
- **PGP Documentation and Apple Guide.** Select this option to install the PGP User's Guide and Apple Guide help.

8. Select a location for your PGP files, then click **Install**.

   A warning screen appears advising you to close all open applications.

9. Close open applications, then click **Continue**.

   The PGP files are copied to the computer.

10. Click **Restart** to reboot the computer.

    The computer restarts. PGP is now installed on the computer.

Using PGP 2

PGP is based on a widely accepted encryption technology known as *public key cryptography* in which two complementary keys, called a *key pair*, are used to maintain secure communications. One of the keys is designated as a *private key* to which only you have access and the other is a *public key* which you freely exchange with other PGP users. Both your private and your public keys are stored in keyring files, which are accessible from the PGPkeys window. It is from this window that you perform all your key management functions.

This section takes a quick look at the procedures you normally follow in the course of using PGP. For details concerning any of these procedures, refer to the appropriate chapters in this book. For a comprehensive overview of PGP encryption technology, refer to *"An Introduction to Cryptography,"* which is included with the product.

Basic steps for using PGP

1. Install PGP on your computer. Refer to "Installing PGP" on page 19 for complete installation instructions.

2. Create a private and public key pair.

   Before you can begin using PGP, you need to generate a key pair. A PGP key pair is composed of a private key to which only you have access and a public key that you can copy and make freely available to everyone with whom you exchange information.

   You have the option of creating a new key pair immediately after you have finished the PGP installation procedure, or you can do so at any time by opening the PGPkeys application.

   For more information about creating a private and public key pair, refer to "Making a key pair" on page 38.

3. Exchange public keys with others.

   After you have created a key pair, you can begin corresponding with other PGP users. You will need a copy of their public key and they will need yours. Your public key is just a block of text, so it's quite easy to trade keys with someone. You can include your public key in an email message, copy it to a file, or post it on a public or corporate key server where anyone can get a copy when they need it.

For more information about exchanging public keys, refer to “Distributing your public key” on page 66 and “Obtaining the public keys of others” on page 70.

4. Validate public keys.

   Once you have a copy of someone’s public key, you can add it to your public keyring. You should then check to make sure that the key has not been tampered with and that it really belongs to the purported owner. You do this by comparing the unique *fingerprint* on your copy of someone’s public key to the fingerprint on that person’s original key. When you are sure that you have a valid public key, you sign it to indicate that you feel the key is safe to use. In addition, you can grant the owner of the key a level of trust indicating how much confidence you have in that person to vouch for the authenticity of someone else’s public key.

   For more information about validating your keys, refer to “Verifying the authenticity of a key” on page 74.

5. Encrypt and sign your email and files.

   After you have generated your key pair and have exchanged public keys, you can begin encrypting and signing email messages and files.

   PGP works on the data generated by other applications. Therefore the appropriate PGP functions are designed to be immediately available to you based on the task you are performing at any given moment. There are several ways to encrypt and sign with PGP:

   - **From within supported email applications (PGP email plug-ins)**. The plug-ins enable you to secure your email from within the supported email application. See “Using PGP within supported email applications” on page 33.

   - **From PGPtools.** PGPtools enables you to perform cryptographic tasks within applications not supported by plug-ins, plus other security tasks, such as wiping files from your disk. See “Using PGPtools” on page 32.

   - **From PGPmenu, or from PGPcontextmenu for Mac OS 8 users.** You can perform most PGP functions from the Finder, or from within most applications by choosing options from PGPmenu. PGPmenu appears as an icon in the menu bar of the Finder and any other applications you have selected. This feature provides immediate access to the PGP functions regardless of which application you are using and is especially useful if you are using an email application that is not supported by the PGP plug-ins.

For more information about encrypting email, refer to "Encrypting and signing email" on page 77. For more information about decrypting files, refer to "Using PGP to encrypt and decrypt files" on page 89.

6. Decrypt and verify your email and files.

   When someone sends you encrypted data, you can unscramble the contents and verify any appended signature to make sure that the data originated with the alleged sender and that it has not been altered.

   - If you are using an email application that is supported by the plug-ins, you can decrypt and verify your messages by selecting the appropriate options from your application's tool bar.

   - If your email application is not supported by the plug-ins, you can copy the message to the clipboard and perform the appropriate functions from there. If you want to decrypt and verify files, you can do so from the clipboard or by using PGPtools. You can also decrypt encrypted files stored on your computer, and verify signed files to ensure that they have not been tampered with.

   For more information about securing email, refer to "Decrypting and verifying email" on page 85. For more information about securing files, refer to "Using PGP to encrypt and decrypt files" on page 89.

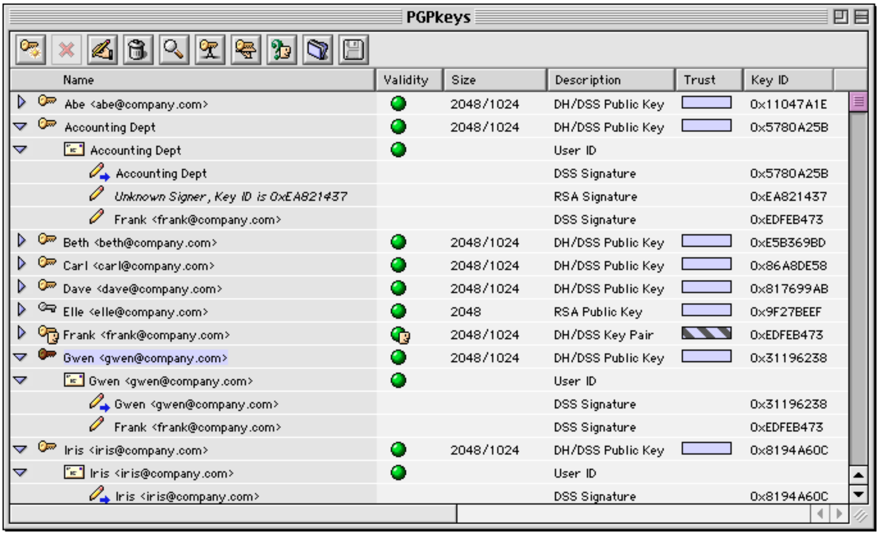
7. Wipe files.

   When you need to permanently delete a file, you can use the Wipe feature to ensure that the file is unrecoverable. The file is immediately overwritten so that it cannot be retrieved using disk recovery software.

   For more information about wiping files, refer to "Using PGP Wipe to delete files" on page 99.

Using PGPkeys

When you choose **PGPkeys** from PGPmenu or from the PGP 6.5 folder, the PGPkeys window opens (Figure 2-1) showing the private and public key pairs you have created for yourself as well as any public keys of other users that you have added to your public keyring.



PGPkeys

Name	Validity	Size	Description	Trust	Key ID
Abe <abe@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x11047A1E
Accounting Dept	●	2048/1024	DH/DSS Public Key	▭	0x5780A25B
Accounting Dept	●		User ID		
Accounting Dept			DSS Signature		0x5780A25B
Unknown Signer, Key ID is 0xEA821437			RSA Signature		0xEA821437
Frank <frank@company.com>			DSS Signature		0xEDFEB473
Beth <beth@company.com>	●	2048/1024	DH/DSS Public Key	▭	0xE5B369BD
Carl <carl@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x86A8DE58
Dave <dave@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x817699AB
Elle <elle@company.com>	●	2048	RSA Public Key	▭	0x9F27BEEF
Frank <frank@company.com>	●	2048/1024	DH/DSS Key Pair	▨	0xEDFEB473
Gwen <gwen@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x31196238
Gwen <gwen@company.com>	●		User ID		
Gwen <gwen@company.com>			DSS Signature		0x31196238
Frank <frank@company.com>			DSS Signature		0xEDFEB473
Iris <iris@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x8194A60C
Iris <iris@company.com>	●		User ID		
Iris <iris@company.com>			DSS Signature		0x8194A60C

Figure 2-1. PGPkeys

(If you have not already created a new key pair, the PGP Key Generation Wizard leads you through the necessary steps. However, before going through the process of creating a new key pair, you should see Chapter 3, “Making and Exchanging Keys,” for complete details about the various options.)

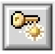
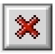
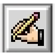
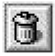
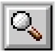
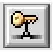
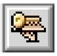
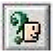
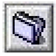
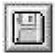
From the PGPkeys window you can create new key pairs and manage all of your other keys. For instance, this is where you examine the attributes associated with a particular key, specify how confident you are that the key actually belongs to the alleged owner, and indicate how well you trust the owner of the key to vouch for the authenticity of other users’ keys. For a complete explanation of the key management functions you perform from the PGPkeys window, see Chapter 6.

PGPkeys icon definitions

PGPkeys menu bar icons

The following table shows all of the icons used in the PGPkeys menu bar, along with a description of their functions.

Table 2-1. PGPkeys menu bar icons

Icon	Function
	Launches the Key Generation Wizard. Click this button to create a new key pair.
	Revokes the currently selected key or signature. Click this button to disable a key or revoke a signature. Revoking a key will prevent anyone from encrypting data to it.
	Allows you to sign the currently selected key. By signing the key, you are certifying that the key and user ID belong to the identified user.
	Deletes the currently selected item. Click this button to remove a key, signature, or photographic ID.
	Opens the **Key Search** window which allows you to search for keys on local keyrings and remote servers.
	Sends the currently selected key to the server. Click this button to upload your key to the Certificate or domain server.
	Updates the currently selected key from a Certificate or domain server. Click this button to import keys from a Certificate or domain server to your keyring.
	Displays the **Properties** dialog box for the currently selected key. Click this button to view the **General** and **Subkey** properties for a key.
	Allows you to import keys from file on to your keyring.
	Allows you to export the selected key to a file.

PGPkeys window icons

The following table shows all of the mini-icons used in the PGPkeys window, along with a description of what they represent.

Table 2-2. PGPkeys window icons

Icon	Description
	A gold key and user represents your Diffie-Hellman/DSS key pair, which consists of your private key and your public key.
	A single gold key represents a Diffie-Hellman/DSS public key.
	A gray key and user represents your RSA key pair, which consists of your private key and your public key.
	A single gray key represents an RSA public key.
	When a key or key pair is dimmed, the keys are temporarily unavailable for encrypting and signing. You can disable a key from the PGPkeys window, which prevents seldom-used keys from cluttering up the Key Selection dialog box.
	This icon indicates that a photographic user ID accompanies the public key.
	A key with a red X indicates that the key has been revoked. Users revoke their keys when they are no longer valid or have been compromised in some way.
	A key with a clock indicates that the key has expired. A key's expiration date is established when the key is created.
	An envelope represents the owner of the key and lists the user names and email addresses associated with the key.
	A gray circle indicates that the key is invalid.
	A green circle indicates that they key is valid. An additional red circle in the ADK column indicates that the key has an associated Additional Decryption Key; an additional gray circle in the ADK column indicates that the key does not have an associated Additional Decryption Key.
	A green circle and user indicates that you own the key, and that it is implicitly trusted.

Table 2-2. PGPkeys window icons

✎ ✒	A pencil or fountain pen indicates the signatures of the PGP users who have vouched for the authenticity of the key. - A signature with a red X through it indicates a revoked signature. - A signature with a dimmed pencil icon indicates a bad or invalid signature. - A signature with a blue arrow next to it indicates that it is exportable.
📄 📄 📄	A certificate represents an X.509 certificate, a recognized electronic document used to prove identity and public key ownership over a communication network. A clock indicates an expired X.509 certificate. A red X indicates a revoked X.509 certificate.
▭	An empty bar indicates an invalid key or an untrusted user.
◧	A half-filled bar indicates a marginally valid key or marginally trusted user.
▧	A striped bar indicates a valid key that you own and is implicitly trusted, regardless of the signatures on the key.
▬	A full bar indicates a completely valid key or a completely trusted user.

Using PGPmenu

You can perform most PGP functions from the Finder or from within most applications by choosing options from PGPmenu, which appears as an icon 🔒 in the menu bar of the Finder and any other applications you have selected. This feature provides immediate access to the PGP functions regardless of which application you are using and is especially useful if you are using an email application that is not supported by the PGP plug-ins.

❒ **NOTE:** PGPmenu now offers direct support for Outlook Express 4.0 or greater and Claris Emailer.

While using email or other text-based applications to which you have added PGPmenu, you can encrypt and sign and decrypt and verify text by choosing options from PGPmenu. While in the Finder, you can encrypt, sign, decrypt, verify, clear passphrase caches and open other PGP applications as shown in Figure 2-2.

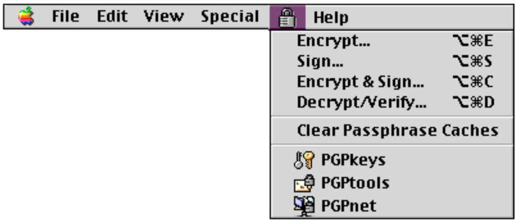


Figure 2-2. PGPmenu

(If you cannot find PGPmenu in one of your applications, you need to add the application to the PGPmenu pane of the **Preferences** dialog box in the PGPkeys or PGPtools application.)

Using PGPtools

If you are using an email application that is not supported by the plug-ins, or if you want to perform PGP functions from within other applications, you can encrypt and sign, decrypt and verify, or securely wipe messages and files directly from PGPtools. You can open PGPtools by:

- Opening the PGP folder and double-clicking the PGPtools icon ( ).
- Storing an alias of PGPtools in the Apple menu, and choosing PGPtools from that menu. You can also store an alias on your desktop.

When PGPtools (Figure 2-3) opens, you can begin your encryption tasks.

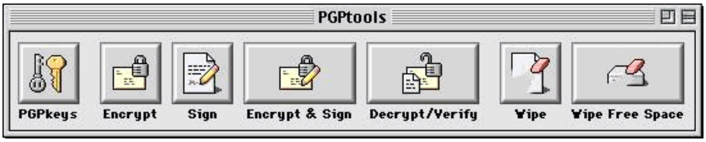


Figure 2-3. PGPtools

If you are working with text or files, you can encrypt, decrypt, sign, and verify by selecting the text or file and then dragging it onto the appropriate button in PGPtools.

If you are working with files, click on the appropriate button in PGPtools to choose a file or select the Clipboard.

Using PGP within supported email applications

One of the most convenient ways to use PGP is through one of the popular email applications supported by the PGP plug-ins. With these plug-ins, you can encrypt and sign if your version of PGP supports the PGP email plug-ins, as well as decrypt and verify your messages while you are composing and reading your mail with a simple click of a button.

If you are using an email application that is not supported by the plug-ins, you can easily encrypt the text of the message using PGPmenu. PGPmenu now offers direct support for Outlook Express 4.0 and Claris Emailer. In addition, if you need to encrypt or decrypt files, you can do so directly from the clipboard or by choosing the appropriate PGP menu option in the Finder. You can also use PGP to encrypt and sign files on the hard disk of your computer for secure storage, to securely wipe files from your hard disk and to wipe free disk space so that sensitive data can't be retrieved with disk recovery software.

If you have Qualcomm Eudora Pro version 3.1 or greater or another popular email application supported by the PGP plug-ins, you can access the necessary PGP functions by clicking the appropriate buttons in your application's toolbar. For example, you click the envelope and lock icon ( ) to indicate that you want to encrypt your message and the pen and paper ( ) to indicate that you want to sign your message. Some applications also have an icon of both a lock and quill, which lets you do both at once.

When you receive email from another PGP user, you decrypt the message and verify the person's digital signature by clicking the opened lock and envelope, or by selecting **Decrypt/Verify** ( ) from PGPtools.

You can also access the PGPkeys window at any time while composing or retrieving your mail by clicking the **PGPkeys** button ( ) in some plug-ins.

Using PGP/MIME

If you are using an email application with one of the plug-ins that supports the PGP/MIME standard, and you are communicating with another user whose email application also supports this standard, both of you can automatically encrypt and decrypt your email messages and any attached files when you send or retrieve your email. All you have to do is turn on the PGP/MIME encryption and signing functions from the **PGP Options** dialog box.

When you receive email from someone who uses the PGP/MIME feature, the mail arrives with an attached icon in the message window indicating that it is PGP/MIME encoded.

To decrypt the text and file attachments in PGP/MIME encapsulated email and to verify any digital signatures, you simply double-click the lock and quill () icon. Attachments are still encrypted if PGP/MIME is not used, but the decryption process is usually more involved for the recipient.

Selecting recipients for encrypted files or email

When you send email to someone whose email application is supported by the PGP plug-ins, the recipient's email address determines which keys to use when encrypting the contents. However, if you enter a user name or email address that does not correspond to any of the keys on your public keyring, or if you are encrypting from PGPmenu or from PGPtools, you must manually select the recipient's public key from the **PGP Key Selection** dialog box.

To select a recipient's public key, drag the icon representing the key into the **Recipients** list box and then click **OK**.

For complete instructions on how to encrypt, sign, decrypt, and verify email, see Chapter 4, "Sending and Receiving Secure Email." For complete instructions on how to encrypt files to store on your hard disk or to send as attachments, see Chapter 5, "Using PGP for Secure File Storage."

Taking shortcuts

Although you will find that PGP is quite easy to use, a number of shortcuts are available to help you accomplish your encryption tasks even quicker. For example, you can perform most PGP functions on files or volumes on your disk using PGPcontextmenu (for Mac OS 8 users), PGPmenu (for System 7 users), or by dragging the file or volume and dropping it onto one of the PGPtools icons.

To access PGP functions using PGPcontextmenu (Mac OS 8)

1. Point to the file or volume, either in a window or on the desktop.
2. Click once while holding down the CONTROL key.
3. The contextual menu appears. PGP appears among the menu options.
4. Choose an option from the PGP menu.

   To close the menu without choosing a command, click outside the menu.

   This feature is available on machines running Mac OS 8 or later and is equivalent to launching and using the features within PGPtools. PGPmenu works similarly for System 7 users.

Getting Help

PGPkeys supports Apple Guide help, which is accessed from the Help menu, and also supports balloon help.

Making and Exchanging Keys 3

This chapter describes how to generate the public and private key pairs that you need to correspond with other PGP users. It also explains how to distribute your public key and obtain the public keys of others so that you can begin exchanging private and authenticated email.

Key concepts

PGP is based on a widely accepted and highly trusted *public key encryption* system, as shown in Figure 3-1, by which you and other PGP users generate a key pair consisting of a private key and a public key. As its name implies, only you have access to your private key, but in order to correspond with other PGP users you need a copy of their public key and they need a copy of yours. You use your private key to sign the email messages and file attachments you send to others and to decrypt the messages and files they send to you. Conversely, you use the public keys of others to send them encrypted email and to verify their digital signatures.

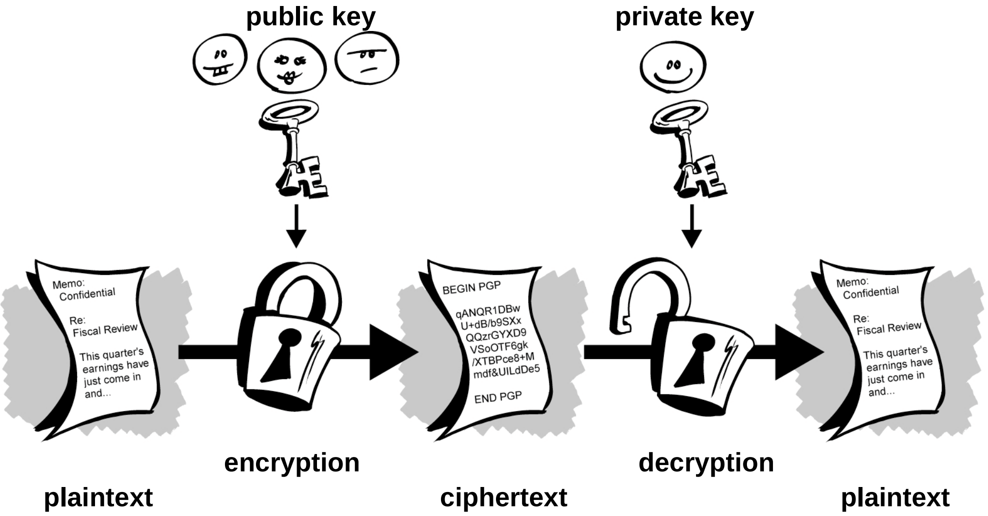


Figure 3-1. Public Key Cryptography diagram

Making a key pair

Unless you have already done so while using another version of PGP, the first thing you need to do before sending or receiving encrypted and signed email is create a new key pair. A key pair consists of two keys: a private key that only you possess and a public key that you freely distribute to those with whom you correspond. You generate a new key pair from PGPkeys using the PGP Key Generation Wizard, which guides you through the process.

NOTE: If you are upgrading from an earlier version of PGP, you have probably already generated a private key and have distributed its matching public key to those with whom you correspond. In this case you don't have to make a new key pair (as described in the next section). Instead, you specify the location of your keys when you run the PGPkeys application. You can go to the **Files** panel of the **Preferences** dialog box and locate your keyring files at any time.

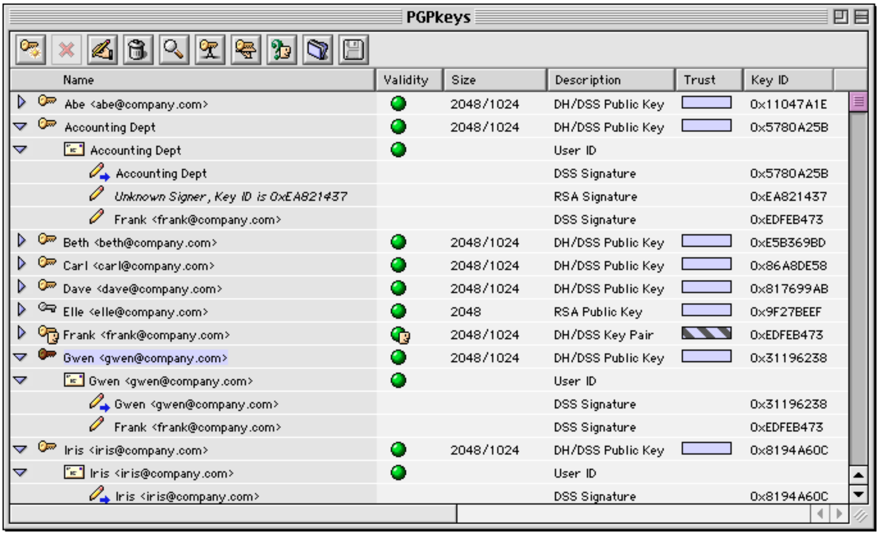
To create a new key pair

1. Open PGPkeys.

   You can open PGPkeys by:

   - double-clicking the PGPkeys icon in the PGP folder.
   - clicking the PGPmenu icon (🔒) in the Finder then clicking PGPkeys

   PGPkeys appears, as shown in Figure 3-2.



PGPkeys

Name	Validity	Size	Description	Trust	Key ID
Abe <abe@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x11047A1E
Accounting Dept	●	2048/1024	DH/DSS Public Key	▭	0x5780A25B
Accounting Dept	●		User ID		
Accounting Dept			DSS Signature		0x5780A25B
Unknown Signer, Key ID is 0xEA821437			RSA Signature		0xEA821437
Frank <frank@company.com>			DSS Signature		0xEDFEB473
Beth <beth@company.com>	●	2048/1024	DH/DSS Public Key	▭	0xE5B369BD
Carl <carl@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x86A8DE58
Dave <dave@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x817699AB
Elle <elle@company.com>	●	2048	RSA Public Key	▭	0x9F27BEEF
Frank <frank@company.com>	●	2048/1024	DH/DSS Key Pair	▨	0xEDFEB473
Gwen <gwen@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x31196238
Gwen <gwen@company.com>	●		User ID		
Gwen <gwen@company.com>			DSS Signature		0x31196238
Frank <frank@company.com>			DSS Signature		0xEDFEB473
Iris <iris@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x8194A60C
Iris <iris@company.com>	●		User ID		
Iris <iris@company.com>			DSS Signature		0x8194A60C

Figure 3-2. PGPkeys

2. Click 🔑 in the PGPkeys menu bar.

   The PGP Key Generation Wizard provides some introductory information on the first screen.

3. When you are finished reading this information, click **Next** to advance to the next pane.

   The PGP Key Generation Wizard asks you to enter your name and email address.

4. Enter your name on the first line and your email address on the second line.

   It's not absolutely necessary to enter your real name or even your email address. However, using your real name makes it easier for others to identify you as the owner of your public key. Also, by using your correct email address, you and others can take advantage of the plug-in feature that automatically looks up the appropriate key on your current keyring when you address mail to a particular recipient. Some Corporate Signing Keys and Additional Decryption Keys have no use for an email address because they do not represent individuals.

5. Click **Next** to advance to the next dialog box.

   The Key Generation Wizard asks you to select a key type.

6. Select a key type, either Diffie-Hellman / DSS or RSA and then click **Next**.

❑ **NOTE:** If your version of PGP does not support RSA, this step may not be available to you. For more information about RSA support, see the WhatsNew file that accompanies the product.

Earlier versions of PGP use an older technology referred to as RSA to generate keys. With PGP Version 5.0 and above, you have the option of creating a new type of key based on the improved Elgamal variant of Diffie-Hellman technology.

- If you plan to correspond with people who are still using RSA keys, you might want to generate an RSA key pair that is compatible with older versions of the program.
- If you plan to correspond with people who have PGP Version 5.0 or later, you can take advantage of the new technology and generate a pair of Diffie-Hellman / DSS keys.
- If you want to exchange email with all PGP users, make an RSA key pair and a Diffie-Hellman / DSS key pair, then use the appropriate pair depending on the version of PGP used by the recipient. You must create a separate key pair for each type of key that you need.

7. The PGP Key Generation Wizard asks you to specify a size for your new keys.

Select a key size from 1024 to 3072 bits, or enter a custom key size from 1024 to 4096 bits.

❑ **NOTE:** A custom key size may take a long time to generate, depending on the speed of the computer you are using. The time required to generate a key is non-deterministic, but should not take more than a few minutes using standard key sizes.

The key size corresponds to the number of bits used to construct your digital key. The larger the key, the less chance that someone will be able to crack it, but the longer it takes to perform the decryption and encryption process. You need to strike a balance between the convenience of performing PGP functions quickly with a smaller key and the increased level of security provided by a larger key. Unless you are exchanging extremely sensitive information that is of enough interest that someone would be willing to mount an expensive and time-consuming cryptographic attack in order to read it, you are safe using a key composed of 1024 bits.

❑ **NOTE:** When creating a Diffie-Hellman / DSS key pair, the size of the DSS portion of the key is less than or equal to the size of the Diffie-Hellman portion of the key, and is limited to a maximum size of 1024 bits.

8. Click **Next** to advance to the next pane.

   The PGP Key Generation Wizard asks you to indicate when the key pair will expire.

9. Indicate when you want your keys to expire. You can either use the default selection, which is **Never**, or you can enter a specific date after which the keys will expire.

   Once you create a key pair and have distributed your public key to the world, you will probably continue to use the same keys from that point on. However, under certain conditions you may want to create a special key pair that you plan to use for only a limited period of time. In this case, when the public key expires, it can no longer be used by someone to encrypt mail for you but it can still be used to verify your digital signature. Similarly, when your private key expires, it can still be used to decrypt mail that was sent to you before your public key expired but can no longer be used to sign mail for others.

10. Click **Next** to advance to the next pane.

    The PGP Key Generation Wizard asks you to enter a passphrase.

11. In the **Passphrase** dialog box, enter the string of characters or words you want to use to maintain exclusive access to your private key. To confirm your entry, press the TAB key to advance to the next line, then enter the same passphrase again.

    Normally, as an added level of security, the characters you enter for the passphrase do not appear on the screen. However, if you are sure that no one is watching, and you would like to see the characters of your passphrase as you type, clear the **Hide Typing** checkbox.

NOTE: Your passphrase should contain multiple words and may include spaces, numbers, and punctuation characters. Choose something that you can remember easily but that others won't be able to guess. The passphrase is case sensitive, meaning that it distinguishes between uppercase and lowercase letters. The longer your passphrase, and the greater the variety of characters it contains, the more secure it is. Strong passphrases include upper and lowercase letters, numbers, punctuation, and spaces but are more likely forgotten. See "Creating a passphrase that you will remember" on page 43, for more information about choosing a passphrase.

WARNING: No one, including Network Associates, can recover a forgotten passphrase.

12. Click **Next** to begin the key generation process.

The PGP Key Generation Wizard indicates that it is busy generating your key.

If you have entered an inadequate passphrase, a warning message appears before the keys are generated and you have the choice of accepting the bad passphrase or entering a more secure one before continuing. For more information about passphrases, see "Creating a passphrase that you will remember" on page 43.

If there is not enough random information upon which to build the key, the **PGP Random Data** dialog box appears. As instructed in the dialog box, move your mouse around and enter a series of random keystrokes until the progress bar is completely filled in. Your mouse movements and keystrokes generate random information that is needed to create a unique key pair.

NOTE: PGPkeys continually gathers random data from many sources on the system, including mouse positions, timings, and keystrokes. If the Random Data dialog box does not appear, it indicates that PGP has already collected all the random data that it needs to create the key pair.

After the key generation process begins, it may take a while to generate the keys. In fact, if you specify a size other than the default values for a Diffie-Hellman / DSS key, the fast key generation option is not used and it may take hours to generate your key at larger sizes. Eventually the PGP Key Generation Wizard indicates that the key generation process is complete.

13. Click **Next** to advance to the next pane.

    The PGP Key Generation Wizard indicates that you have successfully generated a new key pair and asks if you want to send your public key to a certificate server.

14. Specify whether you want your new public key to be sent to the server, and then click **Next** (the default server is specified in the **Server Preferences** dialog box).

    When you send your public key to the certificate server, anyone who has access to that certificate server can get a copy of your key when they need it. For complete details, see "Distributing your public key" on page 62.

    When the key generation process is complete, the final panel appears.

15. Click **Finish**.

    A key pair representing your newly created keys appears in the PGPkeys window. At this point you can examine your keys by checking their properties and the attributes associated with the keys; you may also want to add other email addresses that belong to you. See "Adding and removing information in your key pair" on page 45, for details about modifying the information in your keypair.

Creating a passphrase that you will remember

Encrypting a file and then finding yourself unable to decrypt it is a painful lesson in learning how to choose a passphrase you will remember. Most applications require a password between three and eight letters. A single word password is vulnerable to a dictionary attack, which consists of having a computer try all the words in the dictionary until it finds your password. To protect against this manner of attack, it is widely recommended that you create a word that includes a combination of upper and lowercase alphabetic letters, numbers, punctuation marks, and spaces. This results in a stronger password, but an obscure one that you are unlikely to remember easily. We do not recommend that you use a single-word passphrase.

A passphrase is less vulnerable to a dictionary attack. This is accomplished easily by using multiple words in your passphrase, rather than trying to thwart a dictionary attack by arbitrarily inserting a lot of funny non-alphabetic characters, which has the effect of making your passphrase too easy to forget and could lead to a disastrous loss of information because you can't decrypt your own files. However, unless the passphrase you choose is something that is easily committed to long-term memory, you are unlikely to remember it verbatim. Picking a phrase on the spur of the moment is likely to result in forgetting it entirely. Choose something that is already residing in your long-term memory. Perhaps a silly saying you heard years ago that has somehow stuck in your mind all this time. It should not be something that you have repeated to others recently, nor a famous quotation, because you want it to be hard for a sophisticated attacker to guess. If it's already deeply embedded in your long-term memory, you probably won't forget it.

Of course, if you are reckless enough to write your passphrase down and tape it to your monitor or to the inside of your desk drawer, it won't matter what you choose.

Backing up your keys

Once you have generated a key pair, it is wise to put a copy of it in a safe place in case something happens to the original. PGP prompts you to save a backup copy when you close the PGPkeys application after creating a new key pair.

Your private keys and your public keys are stored in separate keyring files, which you can copy just like any other files to another location on your hard drive or to a floppy disk. By default, the private keyring (PGP Private Keys) and the public keyring (PGP Public Keys) are stored along with the other program files in the "PGP Keyrings" folder in your PGP folder, but you can save your backups in any location you like.

PGP periodically prompts you to backup your keys. When you specify that you want to save a backup copy of your keys, the **Save As** dialog box appears, asking you to specify the location of the backup private and public keyring files that are to be created.

Protecting your keys

Besides making backup copies of your keys, you should be especially careful about where you store your private key. Even though your private key is protected by a passphrase that only you should know, it is possible that someone could discover your passphrase and then use your private key to decipher your email or forge your digital signature. For instance, somebody could look over your shoulder and watch the keystrokes you enter or intercept them on the network or even over the airwaves.

To prevent anyone who might happen to intercept your passphrase from being able to use your private key, you should store your private key only on your own computer. If your computer is attached to a network, you should also make sure that your files are not automatically included in a system-wide backup where others might gain access to your private key. Given the ease with which computers are accessible over networks, if you are working with extremely sensitive information, you may want to keep your private key on a floppy disk, which you can insert like an old-fashioned key whenever you want to read or sign private information.

As another security precaution, consider assigning a different name to your private keyring file and then storing it somewhere other than in the default PGP folder where it will not be so easy to locate. Use the **Files** panel of the **PGPkeys Preferences** dialog box to specify a name and location for your private and public keyring files.

Adding and removing information in your key pair

At any time you can add, change, or remove these items in your key pair:

- a photographic ID
- additional subkeys
- a user name and address
- designated revokers
- an X.509 certificate
- your passphrase

Adding a photographic ID to your key

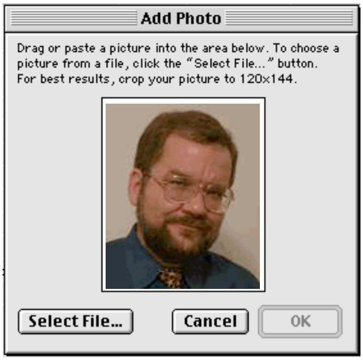
You can include a photographic user ID with your Diffie-Hellman / DSS key.

WARNING: Although you can view the photographic ID accompanied with someone’s key for verification, you should always check and compare the digital fingerprints. See “Verifying someone’s public key” on page 109 for more information about authentication.

To add your photograph to your key

1. Open PGPkeys.
2. Select your key pair and then choose **Add/Photo** from the **Keys** menu.

The **Add Photo** dialog box opens, as shown in Figure 3-3.



Drag or paste a picture into the area below. To choose a picture from a file, click the "Select File..." button. For best results, crop your picture to 120x144.

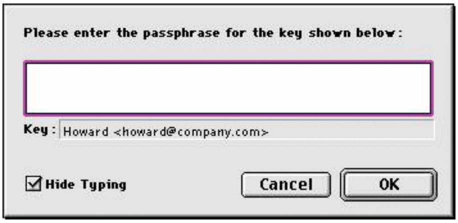
Figure 3-3. Add Photo dialog box

3. Drag or paste your photograph onto the **Add Photo** dialog box or browse to it by clicking **Select File**.

> **NOTE:** The photograph must be a PICT file. For maximum picture quality, crop the picture to 120x144 pixels before adding it to the **Add Photo** dialog box. If you do not do this, PGP will scale the picture for you.

4. Click **OK**.

The **Passphrase** dialog box opens, as shown in Figure 3-4.



Please enter the passphrase for the key shown below:

Key: Howard <howard@company.com>

☑ Hide Typing

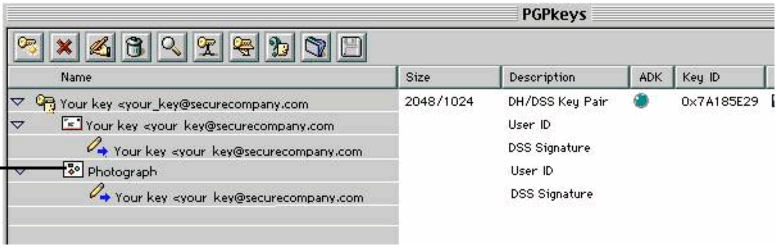
Figure 3-4. Passphrase dialog box

5. Enter your passphrase in the space provided, then click **OK**.

Your photographic user ID is added to your public key and is listed in the PGPkeys window. You can now send your key to the server. See "To send your public key to a certificate server" on page 63, for additional instructions.

To replace your photographic ID

1. Open PGPkeys.
2. Select your key pair from the **Keys** menu, then click ▷ to view the list of items associated with your key.
3. Select the photograph that you want to replace.



Name	Size	Description	ADK	Key ID
Your key <your_key@securecompany.com	2048/1024	DH/DSS Key Pair	●	0x7A185E29
Your key <your key@securecompany.com		User ID		
Your key <your key@securecompany.com		DSS Signature		
Photograph		User ID		
Your key <your key@securecompany.com		DSS Signature		

Your photograph

Figure 3-5. PGPkeys (Example: Photographic User ID)

4. Choose **Delete** from the **Edit** menu.
5. Add your new photographic ID using the instructions outlined in "To add your photograph to your key" on page 45.

Creating new subkeys

Every Diffie-Hellman / DSS key is actually two keys: a DSS signing key and a Diffie-Hellman encryption subkey. PGP Version 6.5 provides the ability to create and revoke new encryption keys without sacrificing your master signing key and the signatures collected on it. One of the most common uses for this feature is to create multiple subkeys that are set to be used during different periods of the key's lifetime. For example, if you create a key that will expire in three years, you might also create 3 subkeys and use each of them for one of the years in the lifetime of the key. This can be a useful security measure and provides an automatic way to periodically switch to a new encryption key without having to recreate and distribute a new public key.

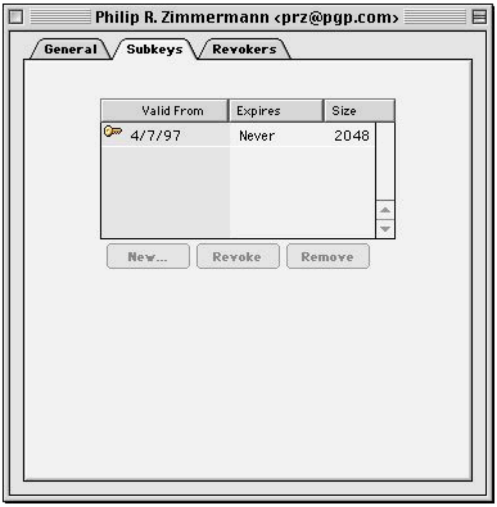
To create new subkeys

1. Open PGPkeys.
2. Select your key pair and then choose **Properties** from the **Keys** menu, or click .

The **Properties** dialog box opens.

3. Click the **Subkeys** tab.

   The **Subkeys** dialog box opens, as shown in Figure 3-6.



Philip R. Zimmermann <prz@pgp.com>

Valid From	Expires	Size
4/7/97	Never	2048

Figure 3-6. PGP key property page (Subkeys dialog box)

4. To create a new subkey, click **New**.

   The **New Subkey** dialog box opens.

5. Enter a key size from 1024 to 3072 bits, or enter a custom key size from 1024 to 4096 bits.

6. Indicate the start date on which you want your subkey to activate.

7. Indicate when you want your subkey to expire. You can either use the default selection, which is **Never**, or you can enter a specific date after which the subkey will expire.

8. Click **OK**.

   The **Passphrase** dialog box appears.

9. Enter your passphrase and then click **OK**.

   Your new subkey is listed in the Subkey window.

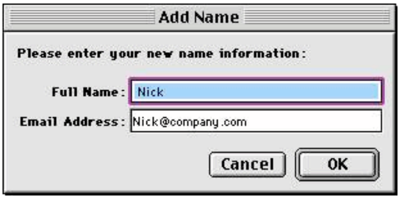
Adding a new user name or address to your key pair

You may have more than one user name or email address for which you want to use the same key pair. After creating a new key pair, you can add alternate names and addresses to the keys. You can only add a new user name or email address if you have both the private and public keys.

To add a new user name or address to your key

1. Open PGPkeys.
2. Select the key pair for which you want to add another user name or address.
3. Choose **Add/Name** from the **Keys** menu.

   The **PGP New User Name** dialog box appears (Figure 3-7).



Please enter your new name information:

Full Name: Nick

Email Address: Nick@company.com

Figure 3-7. PGP New User Name dialog box

4. Enter the new name and email address in the appropriate fields, and then click **OK**.

   The PGP **Enter Passphrase** dialog box appears.
5. Enter your passphrase, then click **OK**.

   The new name is added to the end of the user name list associated with the key. If you want to set the new user name and address as the primary identifier for your key, select the name and address and then choose **Set as Primary Name** from the **Keys** menu.

Adding a designated revoker

It is possible that you might forget your passphrase someday or lose your private key. In this case, you would never be able to use your key again, and you would have no way of revoking your old key when you create a new one. To safeguard against this possibility, you can appoint a third-party key revoker on your public keyring to revoke your key. The third-party you designate will be able to revoke your DH / DSS key, send it to the server and it will be just as if you had revoked it yourself.

To add a designated revoker to your key

1. Open PGPkeys.
2. Select the key pair for which you want to designate a revoker.
3. Select **Add/Revoker** from the **Keys** menu.

   A dialog box opens and displays a list of keys.
4. Select the key(s) in the User ID list that you want to appoint as a designated revoker.
5. Click **OK**.

   A confirmation dialog box appears.
6. Click **OK** to continue.

   The **Passphrase** dialog box appears.
7. Enter your passphrase, then click **OK**.
8. The selected key(s) is now authorized to revoke your key. For effective key management, distribute a current copy of your key to the revoker(s) or upload your key to the server. See “Distributing your public key” on page 62 for instructions.

Adding an X.509 certificate to your PGP key

NOTE: The instructions in this section describe how to add an X.509 certificate to your keypair if you are using the Net Tools PKI Server.

An X.509 digital certificate is a recognized electronic document used to prove identity and public key ownership over a communication network.

You can request an X.509 digital certificate and add it to your keypair using PGP menu preferences and your company's *Certificate Authority* (CA) or a public CA (for example, VeriSign).

There are four main steps to adding an X.509 certificate to your keypair. First, retrieve the Root CA certificate from the CA and add it to your PGP keyring. Next, enter information about the CA in the CA Options panel. Request a certificate from the CA. Your X.509 certificate request is verified and signed by the CA. (The CA's signature on the certificate makes it possible to detect any subsequent tampering with the identifying information or the public key, and it implies that the CA considers the information in the certificate valid.) Finally, retrieve the certificate issued by the CA and add it to your keypair.

To add an X.509 certificate to your PGP keypair

1. **Obtain and add the Root CA certificate to your PGP keyring.**

   To do this, follow these steps:

   1. Open your Web browser and connect to the CA's enrollment site. If you do not know the URL, consult your company's PGP or PKI administrator.
   2. Click the **Download a CA Certificate** link. From the drop-down list, select a certificate authority and the appropriate certificate.
   3. Click **Examine this Certificate** and copy the key block for the Root CA certificate and paste it into PGPkeys.

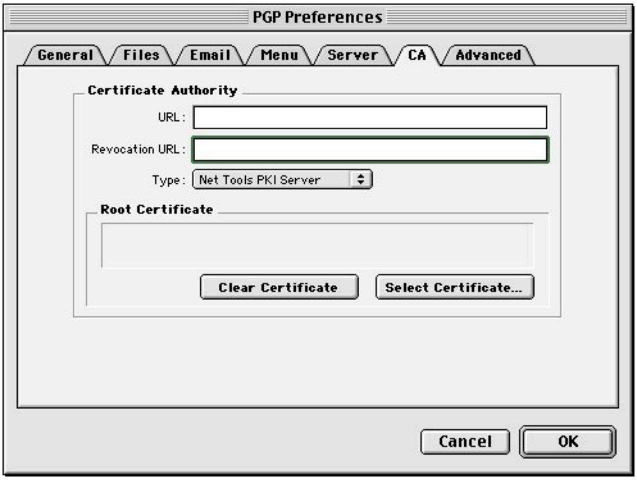
      The **Import Key** dialog box appears and imports the Root CA certificate into your keyring.
   4. Sign the Root CA certificate with your key to make it valid, then open the Key Properties and set the trust level. Trust must be set on the Root CA.

2. **Configure CA Options panel.**

   To do this, follow these steps:

5. Select **Preferences** from the PGPkeys **Edit** menu, then click on the **CA** tab.

   The **CA** panel appears, as shown in Figure 3-8.



Type : Net Tools PKI Server

Figure 3-8. PGP Preferences dialog box (CA Panel)

6. Enter the CA's URL in the **Certificate Authority URL** text box, for example, https://nnn.nnn.nnn.nnn:nnnnn (this is the same URL you used to retrieve the Root CA).

7. If there is a separate URL for retrieving certificate revocation lists (CRLs), enter it in the corresponding text box.

   If you do not know the URL for Revocation, leave this field blank or consult your company's PGP or PKI administrator.

8. In the **Type** box, specify the name of certificate authority you are using. Your options are:

   - Net Tools PKI Server
   - VeriSign OnSite
   - Entrust

9. Click the **Select Certificate** button, then select the Root CA certificate you just retrieved.

The **Root Certificate** text box displays information on the selected root CA certificate. The terminology for the certificate is a policy decision. Typically, the following terminology is true for X.509 certificates:

CN **(Common Name)**	Often a description of the type of certificate (e.g., "Root").
EMAIL	The email address for the certificate holder.
OU **(Organizational Unit)**	The organization to which the certificate belongs (e.g.,"Accounting").
O **(Organization)**	Typically the name of the company to which the certificate belongs (e.g.,"Secure Company").
L **(Locality)**	The location of the holder of the certificate (e.g., "Santa Clara").

10. Click **OK**.

3. Make a certificate request.

To do this, follow these steps:

1. Select PGP keypair and choose **Add/Certificate** from the **Keys** menu.
2. The **Certificate Attributes** dialog box appears.Verify the certificate attributes; use the **Add**, **Edit**, and **Remove** buttons to make any required changes, and click **OK**. The **PGP Enter Passphrase** dialog box appears.
3. Enter the passphrase for your keypair, then click **OK**.

   The **PGP Server Progress** bar appears.

   The certificate request is sent to the CA server. The server authenticates itself to your computer and accepts your request.

   Your company's PGP or PKI administrator verifies your information in the request. The identifying information and public key are assembled and then digitally signed with the CA's own certificate to create your new certificate.

   The administrator sends you an email message stating that your certificate is ready for retrieval.

4. Retrieve your certificate and add it to your keypair.

To do this, follow these steps:

1. In PGPkeys, select the PGPkey for which you made the certificate request.

2. On the **Server** menu, select **Retrieve Certificate**.

   PGP contacts the CA server and automatically retrieves your new X.509 certificate and adds it to your PGPkey.

3. If you are running PGPnet, set this certificate as your X.509 authentication key in PGPnet (**View -> Preferences -> Authentication**).

Changing your passphrase

It's a good practice to change your passphrase at regular intervals, perhaps every three months. More importantly, you should change your passphrase the moment you think it has been compromised, for example, by someone looking over your shoulder as you typed it in.

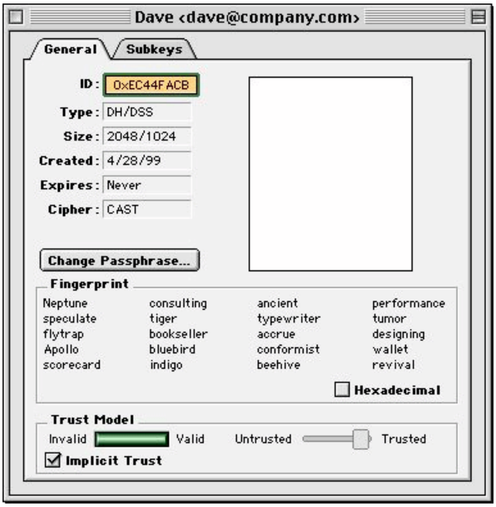
To change your passphrase

1. Open PGPkeys.

2. Select the key for which you want to change the passphrase.

3. Choose **Properties** from the **Keys** menu or click to open the **Properties** dialog box.

The **Properties** dialog box appears, as shown in Figure 3-9.



Dave <dave@company.com>

ID: 0xEC44FACB

Type: DH/DSS

Size: 2048/1024

Created: 4/28/99

Expires: Never

Cipher: CAST

Fingerprint

Neptune	consulting	ancient	performance
speculate	tiger	typewriter	tumor
flytrap	bookseller	accrue	designing
Apollo	bluebird	conformist	wallet
scorecard	indigo	beehive	revival

□ Hexadecimal

Trust Model

Invalid — Valid   Untrusted — Trusted

☑ Implicit Trust

Figure 3-9. Properties dialog box (General panel)

4. Click **Change Passphrase**.

   The **Passphrase** dialog box appears.

NOTE: If you want to change the passphrase for a split key, you must first rejoin the key shares. Click Join to collect the key shares. See "Signing and decrypting files with a split key" on page 94 for information about collecting key shares.

5. Enter your current passphrase in the space provided, then click **OK**.

   The **Change Passphrase** dialog box appears.

6. Enter your new passphrase in the first text box. Press the TAB key to advance to the next text box and confirm your entry by entering your new passphrase again.

7. Click **OK**.

WARNING: If you are changing your passphrase because you feel that your passphrase has been compromised, you should wipe all backup keyrings and wipe your freespace.

Deleting a key or signature on your PGP keyring

At some point you may want to remove a key or a signature from your PGP keyring. When you delete a key or signature from a key, it is removed and not recoverable. Signatures and user IDs can be re-added to a key, and an imported public key can be re-imported to your keyring. However, a private key that exists only on that keyring cannot be recreated, and all messages encrypted to its public key copies can no longer be decrypted.

NOTE: If you want to delete a signature or user ID associated with your public key on a certificate server, see "Updating your key on a certificate server" on page 64 for instructions.

To delete a key or signature from your PGP keyring

1. Open PGPkeys.
2. Select the key or signature you want to delete.
3. Choose **Clear** from the **Edit** menu or click 🗑 in the PGPkeys menu bar.

   The **Confirmation** dialog box appears.
4. Click the **OK** button.

Splitting and rejoining keys

Any private key can be split into shares among multiple "shareholders" using a cryptographic process known as Blakely-Shamir key splitting. This technique is recommended for extremely high security keys. For example, Network Associates keeps a corporate key split between multiple individuals. Whenever we need to sign with that key, the shares of the key are rejoined temporarily.

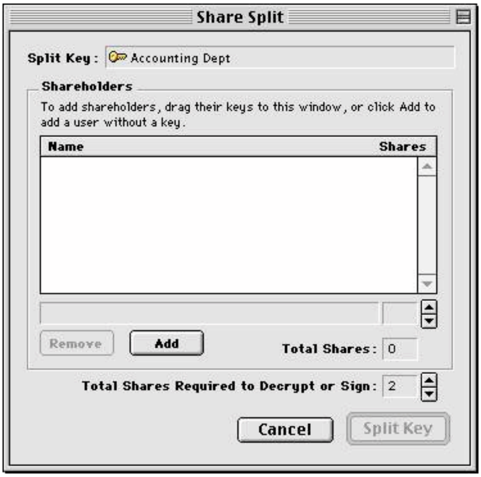
Creating a split key

To split a key, select the key pair to be split and choose **Share Split** from the **Keys** menu. You are then asked to set up how many different shares will be required to rejoin the key. The shares are saved as files either encrypted to the public key of a shareholder or encrypted conventionally if the shareholder has no public key. After the key has been split, attempting to sign with it or decrypt with it will automatically attempt to rejoin the key. For information about rejoining a split key, see "Signing and decrypting files with a split key" on page 94.

To create a split key with multiple shares

1. Open PGPkeys.
2. In PGPkeys, create a new key pair or select an existing key pair that you want to split.
3. On the **Keys** menu, choose Share Split.

   The **Share Split** dialog box opens (Figure 3-10) on top of PGPkeys.



Split Key : Accounting Dept

To add shareholders, drag their keys to this window, or click Add to add a user without a key.

Total Shares : 0

Total Shares Required to Decrypt or Sign : 2

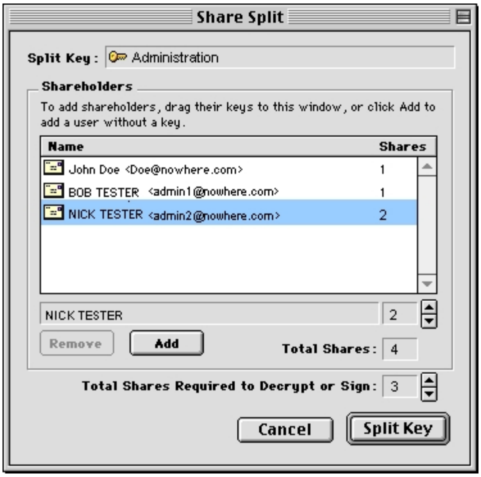
Figure 3-10. Share Split dialog box

4. Add shareholders to the key pair by dragging their keys from PGPkeys to the **Shareholder** list in the **Share Split** dialog box.

   To add a shareholder that does not have a public key, click **Add** in the **Share Split** dialog box, enter the persons name and then allow the person to type in their passphrase.

5. When all of the shareholders are listed, you can specify the number of key shares that are necessary to decrypt or sign with this key.

   In Figure 3-11, for example, the total number of shares that make up the Accounting Dept key is four and the total number of shares required to decrypt or sign is three. This provides a buffer in the event that one of the shareholders is unable to provide their key share or forgets the passphrase.



Share Split

Split Key : Administration

Shareholders

To add shareholders, drag their keys to this window, or click Add to add a user without a key.

Name	Shares
John Doe <Doe@nowhere.com>	1
BOB TESTER <admin1@nowhere.com>	1
NICK TESTER <admin2@nowhere.com>	2

NICK TESTER | 2

Total Shares : 4

Total Shares Required to Decrypt or Sign : 3

Figure 3-11. Share Split dialog box (Example)

By default, each shareholder is responsible for one share. To increase the number of shares a shareholder possesses, click the name in the shareholder's list to display it in the text field below. Type the new number of key shares or use the arrows to select a new amount.

6. Click **Split Key**.

   A dialog box opens and prompts you to select a directory in which to store the shares.

7. Select a location to store the key shares.

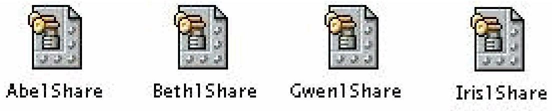
   The **Passphrase** dialog box appears.

8. Enter the passphrase for the key you want to split and then click **OK**.

   A confirmation dialog box opens.

9. Click **Yes** to split the key.

   The key is split and the shares are saved in the location you specified. Each key share is saved with the shareholder's name as the file name and, as shown in the example below:



10. Distribute the key shares to the owners, then delete the local copies.

Once a key is split among multiple shareholders, attempting to sign or decrypt with it will cause PGP to automatically attempt to rejoin the key. To learn how to rejoin a split key to sign or decrypt files, "Signing and decrypting files with a split key" on page 94.

Rejoining split keys

Once a key is split among multiple shareholders, attempting to sign or decrypt with it will cause PGP to automatically attempt to rejoin the key. There are two ways to rejoin the key, locally and remotely.

Rejoining key shares locally requires the shareholders presence at the rejoining computer. Each shareholder is required to enter the passphrase for their key share.

Rejoining key shares remotely requires the remote shareholders to authenticate and decrypt their keys before sending them over the network. PGP's Transport Layer Security (TLS) provides a secure link to transmit key shares which allows multiple individuals in distant locations to securely sign or decrypt with their key share.

IMPORTANT: Before receiving key shares over the network, you should verify each shareholder's fingerprint and sign their public key to ensure that their authenticating key is legitimate. To learn how to verify a keypair, see "Verify with a digital fingerprint" on page 71.

To rejoin a split key

1. Contact each shareholder of the split key. To rejoin key shares locally, the shareholders of the key must be present.

   To collect key shares over the network, ensure that the remote shareholders are prepared to send their key share file. Remote shareholders must have:

   – their key share file and password

   – a keypair (for authentication to the computer that is collecting the key shares)

   – a network connection

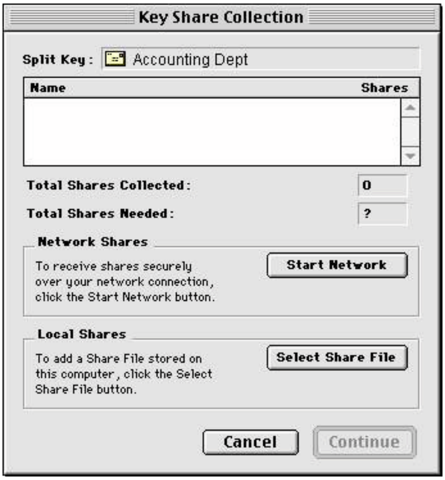
   – the IP address or Domain Name of the computer that is collecting the key shares

2. At the rejoining computer, use the Finder to select the file(s) that you want to sign or decrypt with the split key.

The **PGP Enter Passphrase for Selected Key** dialog box appears with the split key selected.

3. Click **OK** to reconstitute the selected key.

   The **Key Share Collection** dialog box appears, as shown in Figure 3-12.



Key Share Collection

Split Key: Accounting Dept

Name	Shares

Total Shares Collected: 0

Total Shares Needed: ?

Network Shares

To receive shares securely over your network connection, click the Start Network button.

Local Shares

To add a Share File stored on this computer, click the Select Share File button.

Figure 3-12. Key Share Collection dialog box

4. Do one of the following:

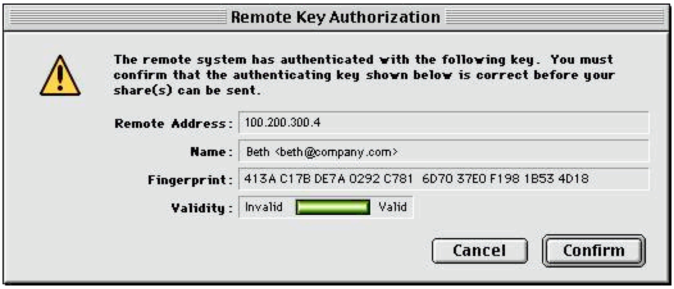
   - **If you are collecting the key shares locally**, click **Select Share File** and then locate the share files associated with the split key. The share files can be collected from the hard drive, a floppy disk, or a mounted drive. Continue with Step 5.
   - **If you are collecting key shares over the network**, click **Start Network**.

     The **Passphrase** dialog box opens. In the **Signing Key** box, select the keypair that you want to use for authentication to the remote system and enter the passphrase. Click **OK** to prepare the computer to receive the key shares.

     The status of the transaction is displayed in the **Network Shares** box. When the status changes to “Listening,” the PGP application is ready to receive the key shares.

     At this time, the shareholders must send their key shares. To learn how to send key shares to the rejoining computer, see “To send your key share over the network” on page 61.

When a share is received, the **Remote Authentication** dialog box appears, as shown in Figure 3-13.



Remote Key Authorization

The remote system has authenticated with the following key. You must confirm that the authenticating key shown below is correct before your share(s) can be sent.

Remote Address: 100.200.300.4

Name: Beth <beth@company.com>

Fingerprint: 413A C17B DE7A 0292 C781 6D70 37E0 F198 1B53 4D18

Validity: Invalid 🟩 Valid

Figure 3-13. Remote Authentication dialog box

If you have not signed the key that is being used to authenticate the remote system, the key will be considered invalid. Although you can rejoin the split key with an invalid authenticating key, it is not recommended. You should verify each shareholder’s fingerprint and sign their public key to ensure that the authenticating key is legitimate.

Click **Confirm** to accept the share file.

5. Continue collecting key shares until the value for **Total Shares Collected** matches the value for **Total Shares Needed** in the **Key Shares Collection** dialog box.

6. Click **OK**.

   The file is signed or decrypted with the split key.

To send your key share over the network

1. When you are contacted by the person who is rejoining the split key, make sure that you have these items:

   – your key share file and password

   – your keypair (for authentication to the computer that is collecting the key shares)

   – a network connection

   – the IP address or Domain Name of the rejoining computer collecting the key shares

2. Choose Send Share File from the PGPkeys File menu.

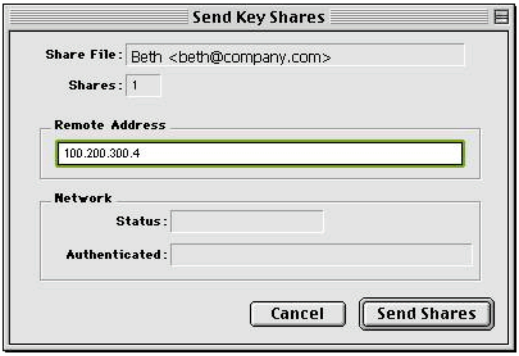
   The **Select Share File** dialog box appears.

3. Locate your key share and then click **Open**.

   The **PGP Enter Passphrase** dialog box appears.

4. Enter your passphrase and then click **OK**.

   The **Send Key Shares** dialog box appears, as shown in Figure 3-14.



Send Key Shares

Share File: Beth <beth@company.com>

Shares: 1

Remote Address

100.200.300.4

Network

Status:

Authenticated:

Figure 3-14. Send Key Shares dialog box

5. Enter the IP address or the Domain Name of the rejoining computer in the **Remote Address** text box, then click **Send Shares**.

   The status of the transaction is displayed in the **Network Status** box. When the status changes to "Connected," you are asked to authenticate yourself to the rejoining computer.

   The **Remote Authentication** dialog box appears asking you to confirm that the remote computer is the one to whom you want to send your key share.

6. Click **Confirm** to complete the transaction.

   After the remote computer receives your key shares and confirms the transaction, a message box appears stating that the shares were successfully sent.

7. Click **OK**.

8. Click **Done** in the **Key Shares** window when you have completed sending your key share.

Distributing your public key

After you create your keys, you need to make them available to others so that they can send you encrypted information and verify your digital signature. There are three ways in which you can distribute your public key:

- Make your public key available through a public certificate server,
- Include your public key in an email message,

  Or
- Export your public key or copy it to a text file.

Your public key is basically composed of a block of text, so it is quite easy to make it available through a public certificate server, include it in an email message, or export or copy it to a file. The recipient can then use whatever method is most convenient to add your public key to their public keyring.

Making your public key available through a certificate server

The best method for making your public key available is to place it on a public certificate server where anyone can access it. That way, people can send you email without having to explicitly request a copy of your key. It also relieves you and others from having to maintain a large number of public keys that you rarely use. There are a number of certificate servers worldwide, including those offered by Network Associates, Inc., where you can make your key available for anyone to access. Your Security Officer will usually pre-configure your keyserver settings so that everything works correctly for your site.

To send your public key to a certificate server

1. Connect to the Internet.
2. Open PGPkeys.
3. Select the icon that represents the public key you want to post on the certificate server.
4. Open the **Server** menu, then select the certificate server you want to post on from the **Send To** submenu. PGP lets you know that the keys are successfully uploaded to the server.

Once you place a copy of your public key on a certificate server, you can tell people who want to send you encrypted data or to verify your digital signature to get a copy of your key from the server. Even if you don’t explicitly point them to your public key, they can get a copy by searching the certificate server for your name or email address. Many people include the Web address for their public key at the end of their email messages; in most cases the recipient can just double-click the address to access a copy of your key on the server. Some people even put their PGP fingerprint on their business cards for easier verification.

Updating your key on a certificate server

If you ever need to change your email address, or if you acquire new signatures, all you have to do to replace your old key is send a new copy to the server; the information is automatically updated. However, you should keep in mind that public certificate servers are only capable of updating new information and will not allow removal of user names or signatures from your key. To remove signatures or user names from your key, see "Removing signatures or user names associated with your key" for instructions. If your key is ever compromised, you can revoke it, which tells the world to no longer trust that version of your key. See Chapter 6, "Managing Keys and Setting PGP Options" for more details on how to revoke a key.

Removing signatures or user names associated with your key

At some point you may want to remove a key, a signature, or a user ID associated with a particular key.

Public certificate servers are only capable of updating new information and will not allow removal of user names or signatures from your key. To remove signatures or user names associated with your public key, you must first remove your key from the server, make the required change, then post your key back on the server.

If your PGP Server settings are configured to synchronize keys with the server upon adding names / photos / revokers to your key, your key is automatically updated on the server. If, however, your keys do not automatically synchronized with the server, follow the instructions outlined below to manually update your key on the certificate server.

NOTE: When you delete a key, signature, or user ID from a key, it is removed and not recoverable. Signatures and user IDs can be re-added to a key, and an imported public key can be re-imported to your keyring. However, a private key that exists only on that keyring cannot be recreated, and all messages encrypted to its public key copies can no longer be decrypted.

To remove signatures or user names associated with your key on a certificate server

IMPORTANT: This procedure is for removing signatures or user names associated with your key on LDAP certificate servers only. Additionally, the certificate server must be configured to allow this action. If you do not know the type server or its configuration settings, consult the certificate server administrator for your company before updating your key.

1. Open PGPkeys.
2. Choose **Search** from the **Server** menu or click 🔍 in the PGPkeys menu.

   The **PGPkeys Search** window appears.
3. Choose the server you want to search from the **Search for Keys On** menu.
4. Specify your search criteria to locate your public key:

   The default is **User ID**, but you can click the arrows to select **Key ID**, **Key Status**, **Key Type**, **Key Size**, **Creation Date**, **or Expiration Date**. For example, you might search for all keys with the User ID of Fred.
5. To begin the search, click **Search**.

   The results of the search appear in the window.
6. Select the key that you want to remove from the server, then select **Delete** from the Server menu.

   The **Passphrase** dialog box appears.
7. Enter the passphrase for the key you want to remove from the server and then click **OK**.

   Confirmation dialog appears and the key is removed.
8. Update your key (remove the unwanted signatures or user names).
9. Copy the updated key to the server (see "Making your public key available through a certificate server" on page 63 for instructions).

   If the server on which you are updating your public key is configured to synchronize keys with other public certificate servers, your key will be updated on the other servers automatically upon synchronization.

IMPORTANT: If you delete your key from a certificate server, you should be aware that someone who has your public key on their keyring can upload it to the server again. You should check the server periodically to see if the key has reappeared - you may have to delete your key from the server more than once.

Including your public key in an email message

Another convenient method of delivering your public key to someone is to include it along with an email message.

To include your public key in an email message

1. Open PGPkeys.
2. Select your key pair and then choose Copy from the Edit menu.
3. Open the editor you use to compose your email messages, place the cursor in the desired area, and then choose Paste from the Edit menu. In newer email applications, you can simply drag your key from PGPkeys into the text of your email message to transfer the key information.

When you send someone your public key, be sure to sign the email. That way, the recipient can verify your signature and be sure that no one has tampered with the information along the way. Of course, if your key has not yet been signed by any trusted introducers, recipients of your signature can only truly be sure the signature is from you by verifying the fingerprint on your key.

Exporting your public key to a file

Another method of distributing your public key is to copy it to a file and then make this file available to the person with whom you want to communicate.

To export your public key to a file

There are three ways to export or save your public key to a file:

- Select the icon representing your key pair from PGPkeys, then choose **Export** on the **Keys** menu and enter the name of the file where you want the key to be saved,
- Drag the icon representing your key pair from PGPkeys to the folder that you want the key to be saved,

  Or

- Select the icon representing your key pair in PGPkeys, choose **Copy** on the **Edit** menu, then choose **Paste** to insert the key information into a text document.

❑ **NOTE:** If you are sending your key to colleagues who are using PCs, enter a name of up to eight initial characters and three additional characters for the file type extension (for example, MyKey.txt).

Obtaining the public keys of others

Just as you need to distribute your public key to those who want to send you encrypted mail or to verify your digital signature, you need to obtain the public keys of others so you can send them encrypted mail or verify their digital signatures.

To obtain someone's public key

There are three ways you can obtain someone's public key:

- Get the key from a public certificate server,

  Or

- Import the public key from an exported file.

Public keys are just blocks of text, so they are easy to add to your keyring by importing them from a file or and then pasting them into your public keyring.

Getting public keys from a certificate server

If the person to whom you want to send encrypted mail is an experienced PGP user, chances are that they have placed a copy of their public key on a certificate server. This makes it very convenient for you to get a copy of their most up-to-date key whenever you want to send them mail and also relieves you from having to store a lot of keys on your public keyring.

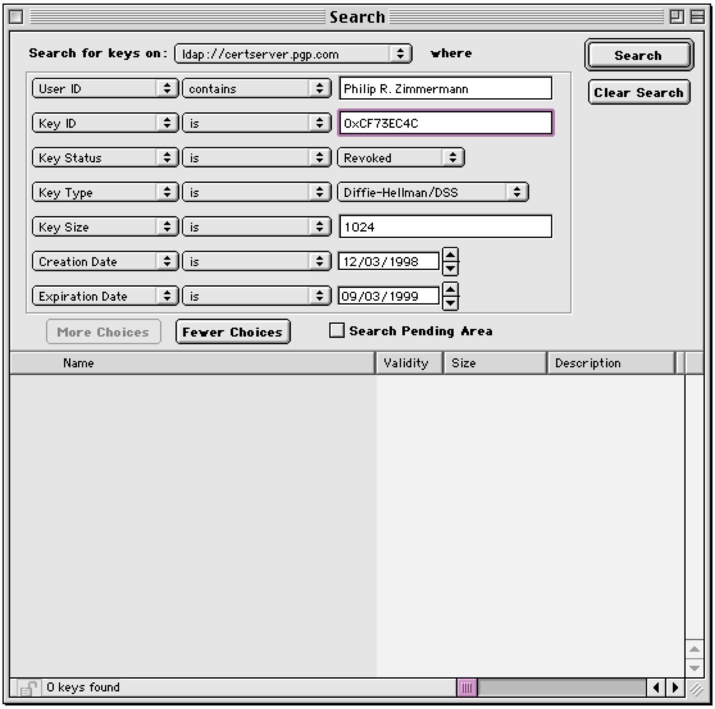
Your security officer may direct you to use a corporate certificate server that holds all of your organization's frequently used keys. In this case, your PGP software is probably already configured to access the appropriate server.

There are a number of public certificate servers, such as the one maintained by Network Associates, Inc., where you can locate the keys of most PGP users. If the recipient has not pointed you to the Web address where his or her public key is stored, you can access any certificate server and do a search for the user's name or email address, because all certificate servers are regularly updated to include the keys stored on all the other servers.

To get someone's public key from a certificate server

1. Open PGPkeys.
2. Choose **Search** from the **Server** menu or click the **Search** button (🔍) in PGPkeys.

   The **PGPkeys Search** window appears as in Figure 3-15.



Search for keys on: ldap://certserver.pgp.com where

User ID contains Philip R. Zimmermann

Key ID is 0xCF73EC4C

Key Status is Revoked

Key Type is Diffie-Hellman/DSS

Key Size is 1024

Creation Date is 12/03/1998

Expiration Date is 09/03/1999

☐ Search Pending Area

0 keys found

Figure 3-15. PGPkeys Search window
(*More Choices view*)

3. Choose the server you wish to search from the **Search for Keys On** menu.
4. Specify your search criteria.

You can search for keys on a certificate server by specifying values for these key characteristics:

- User ID
- Key ID
- Key Status (Revoked or Disabled)
- Key Type (Diffie-Hellman or RSA)
- Creation date
- Expiration date
- Revoked keys
- Disabled keys
- Key size
- Keys signed by a particular key

The inverse of most of these operations is also available. For example, you may search using “User ID is not Bob” as your criteria.

5. Enter the value you want to search for.
6. Click **More Choices** to add additional criteria to your search; for example, Key IDs with the name Fred created on or before October 6, 1997.
7. To begin the search, click **Search**.

   A progress bar appears displaying the status of the search.

   ❑ **NOTE:** To cancel a search in progress, click Cancel.

   The results of the search appear in the window.
8. To import the keys, drag them to the PGPkeys main window.
9. Click **Clear Search** to clear your search criteria.

Importing keys

You can import public keys and PKCS-12 X.509 private keys to your PGP public keyring. To import from your browser by copying and pasting into your public keyring.

Another method for obtaining someone’s public key is to have that person save it to a file from which you can import, or it or copy and paste it into your public keyring.

To import a public key from a file

There are three methods of extracting someone’s public key and adding it to your public keyring:

- Choose **Import from the Keys** menu and then navigate to the file where the public key is stored,
- Drag the file containing the public key onto the main PGPkeys window,

  Or

- Open the text document where the public key is stored, select the block of text representing the key, and then choose **Copy** from the **Edit** menu. Go to PGPkeys and choose **Paste** from the **Edit** menu to copy the key. The key then shows up as an icon in PGPkeys.

You can also obtain PKCS-12 X.509 private keys by exporting them from your browser and dropping them into PGPkeys, or by choosing **Import** from the **Keys** menu.

Verifying the authenticity of a key

When you exchange keys with someone, it is sometimes hard to tell if the key really belongs to that person. PGP software provides a number of safeguards that allow you to check a key’s authenticity and to certify that the key belongs to a particular owner (that is, to *validate* it). The PGP program also warns you if you attempt to use a key that is not valid and also defaults to warn you when you are about to use a marginally valid key.

Why verify the authenticity of a key?

One of the major vulnerabilities of public key encryption systems is the ability of sophisticated eavesdroppers to mount a “man-in-the-middle” attack by replacing someone’s public key with one of their own. In this way they can intercept any encrypted email intended for that person, decrypt it using their own key, then encrypt it again with the person’s real key and send it on to them as if nothing had ever happened. In fact, this could all be done automatically through a sophisticated computer program that stands in the middle and deciphers all of your correspondence.

Based on this scenario, you and those with whom you exchange email need a way to determine whether you do indeed have legitimate copies of each others' keys. The best way to be completely sure that a public key actually belongs to a particular person is to have the owner copy it to a floppy disk and then physically hand it to you. However, you are seldom close enough to personally hand a disk to someone; you generally exchange public keys via email or get them from a public certificate server.

Verify with a digital fingerprint

You can determine if a key really belongs to a particular person by checking its digital fingerprint, a unique series of numbers or words generated when the key is created. By comparing the fingerprint on your copy of someone's public key to the fingerprint on their original key, you can be absolutely sure that you do in fact have a valid copy of their key. To learn how to verify with a digital fingerprint, see "Verifying someone's public key" on page 114.

Validating the public key

Once you are absolutely convinced that you have a legitimate copy of someone's public key, you can then sign that person's key. By signing someone's public key with your private key, you are certifying that you are sure the key belongs to the alleged user. For instance, when you create a new key, it is automatically certified with your own digital signature. By default, signatures you make on other keys are not exportable, which means they apply only to the key when it is on your local keyring. For detailed instructions on signing a key, see "Signing someone's public key" on page 116.

Working with trusted introducers

PGP users often have other trusted users sign their public keys to further attest to their authenticity. For instance, you might send a trusted colleague a copy of your public key with a request that he or she certify and return it so you can include the signature when you post your key on a public certificate server. Using PGP, when someone gets a copy of your public key, they don't have to check the key's authenticity themselves, but can instead rely on how well they trust the person(s) who signed your key. PGP provides the means for establishing this level of validity for each of the public keys you add to your public keyring and shows the level of trust and validity associated with each key PGPkeys. This means that when you get a key from someone whose key is signed by a trusted introducer, you can be fairly sure that the key belongs to the purported user. For details on how to sign keys and validate users, see "Signing someone's public key" on page 116.

Your Security Officer can act as a trusted introducer, and you may then trust any keys signed by the corporate key to be valid keys. If you work for a large company with several locations, you may have regional introducers, and your Security Officer may be a meta-introducer, or a trusted introducer of trusted introducers.

What is a trusted introducer?

PGP uses the concept of a trusted introducer, someone who you trust to provide you with keys that are valid. This concept may be familiar to you from Victorian novels, in which people gave letters of introduction to one another. For example, if your uncle knew someone in a faraway city with whom you might want to do business, he might write a letter of introduction to his acquaintance. With PGP, users can sign one another's keys to validate them. You sign someone's key to indicate that you are sure that their key is valid, which means that it truly is their key. There are several ways to do this. When a trusted introducer signs another person's key, you trust that the keys they sign are valid, and you do not feel that you must verify their keys before using them.

What is a meta-introducer

PGP also supports the concept of a meta-introducer--a trusted introducer of trusted introducers. If you work in a very large company, you might have a regional security officer, a trusted introducer, who would sign users' keys. You could trust that these keys were valid because the regional security officer had performed the actions to ensure validity. The organization may also have a head security officer who works with the local security officers, so that a person in a West Coast office could trust a person in an East Coast office, because both their keys had been signed by their respective regional security officers, who in turn had their keys signed by the head security officer, who is a meta-introducer. This allows the establishment of a trust hierarchy in the organization.

Sending and Receiving Secure Email 4

This chapter explains how to encrypt and sign the email you send to others and decrypt and verify the email others send to you.

Encrypting and signing email

There are three ways to encrypt and sign email messages. The quickest and easiest way to encrypt and sign email is with an application supported by the PGP email plug-ins. Although the procedure varies slightly between different email applications, you perform the encryption and signing process by clicking the appropriate buttons in the application's toolbar.

If you are using an email application that is not supported by the PGP plug-ins, you can encrypt and sign your email messages via PGPmenu, which is compatible with most popular text-based applications. When accessing this menu from the Finder, you can encrypt and sign or decrypt and verify files and even entire folders.

TIP: If you are sending sensitive email, consider leaving your subject line blank or creating a subject line that does not reveal the contents of your encrypted message.

If you do not have one of the email applications that is supported by PGP, see Chapter 5 for information about how to encrypt files.

As an alternative to using the plug-ins, you can use PGPtools to encrypt and sign your email text and attachments before sending them, see "To encrypt and sign text using PGPtools" on page 77.

Encrypting and signing with supported email applications

When you encrypt and sign with an email application that is supported by the PGP plug-ins, you have two choices, depending on what type of email application the recipient is using. If you are communicating with other PGP users who have an email application that supports the PGP/MIME standard, you can take advantage of a PGP/MIME feature to encrypt and sign your email messages and any file attachments automatically when you send them. If you are communicating with someone who does not have a PGP/MIME-compliant email application, you should encrypt your email with PGP/MIME turned off to avoid any compatibility problems. Refer to Table 4-1, “PGP Plug-in Features,” for a list of plug-ins and their features.

Table 4-1. PGP Plug-in Features

	Eudora 3 - 4.0	Claris Emailer 2.0	Outlook Express 4.0
PGP/MIME	Yes	No	No
Auto-decrypt	No	decryption supported through PGPmenu	decryption supported through PGPmenu
Encrypt HTML	Yes	No	No
View decrypted HTML as an HTML document	Yes	No	No
Encrypt attachments	Yes	No	No
Encrypt/Sign defaults	No	No	No
Recipient matching	Yes	Yes	Yes

To encrypt and sign with supported email applications

1. Use your email application to compose your email message as you normally would.

2. When you have finished composing the text of your email message, open the PGPmenu by clicking the PGP lock icon (🔒) on your email applications menu bar. Choose one of these options from the PGPmenu.

- **Encrypt.** Select this option to only encrypt the text of your message before sending,
- **Sign.** Select this option to only sign your message before sending,

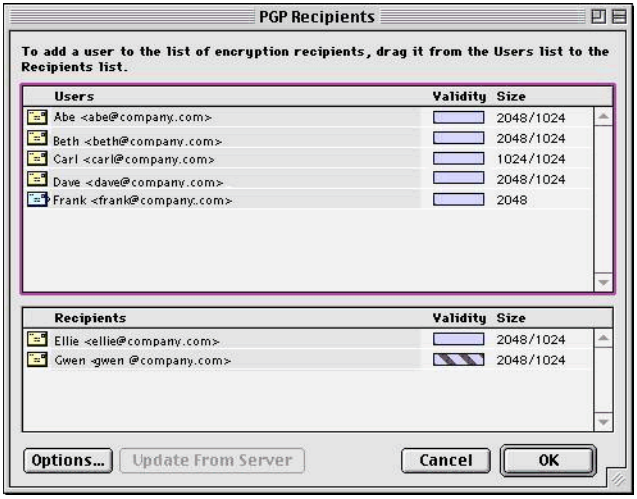
  Or

- **Encrypt and Sign.** Select this option to both encrypt and sign your message before sending.

NOTE: If you know that you are going to use PGP / MIME regularly, you can leave this turned on by selecting the appropriate settings in the **Email** panel of the **Preferences** dialog box.

3. Send your message as you normally do.

   If you have a copy of the public keys for every one of the recipients, the appropriate keys are used. However, if you specify a recipient for whom there is no corresponding public key or one or more of the keys have insufficient validity, the **PGP Key Selection** dialog box appears (Figure 4-1) so that you can specify the correct key.



To add a user to the list of encryption recipients, drag it from the Users list to the Recipients list.

Users	Validity	Size
Abe <abe@company.com>	▭	2048/1024
Beth <beth@company.com>	▭	2048/1024
Carl <carl@company.com>	▭	1024/1024
Dave <dave@company.com>	▭	2048/1024
Frank <frank@company.com>	▭	2048

Recipients	Validity	Size
Ellie <ellie@company.com>	▭	2048/1024
Gwen <gwen @company.com>	▨	2048/1024

Figure 4-1. PGP Recipient Selection window

4. Drag the public keys for those who are to receive a copy of the encrypted email message into the Recipients list box. You can also double-click any of the keys to move them from one area of the screen to the other.

   The **Validity** icon indicates the minimum level of confidence that the public keys in the **Recipient** list are valid. This validity is based on the signatures associated with the key. See Chapter 6, “Managing Keys and Setting PGP Options,” for details.

5. You can choose from the following encryption options depending on the type of data you are encrypting:

   - **Secure Viewer.** Select this option to protect the data from TEMPEST attacks upon decryption. If you select this option, the decrypted data is displayed in a special TEMPEST attack prevention font that is unreadable to radiation capturing equipment. For more information about TEMPEST attacks, see “Vulnerabilities” on page 246.

     NOTE: The Secure Viewer option may not be compatible with previous versions of PGP. Files encrypted with this option enabled can be decrypted by previous versions of PGP, however this feature may be ignored.

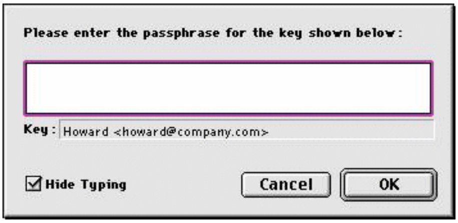
   - **Conventional Encrypt.** Select this option to use a common passphrase instead of public key encryption. If you select this option, the file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you will be asked to choose.

   - **Self Decrypting Archive.** Select this option to create a self decrypting executable file. If you select this option, the file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you are asked to choose. The resulting executable file can be decrypted by simply double-clicking on it and entering the appropriate passphrase. This option is especially convenient for users who are sending encrypted files to people who do not have PGP software installed. Note that sender and recipient must be on the same platform.

     You can also use this feature without a passphrase to create compact Self-Extracting Archives (SEA) which are not encrypted. The resulting archives run on both PowerPC and 68K Macs.

6. Click **OK** to encrypt and sign your mail.

   If you have elected to sign the encrypted data, the **Signing Key Passphrase** dialog box appears, as shown in Figure 4-2, requesting your passphrase before the mail is sent.


Please enter the passphrase for the key shown below:

Key: Howard <howard@company.com>

☑ Hide Typing

Figure 4-2. Signing Key Passphrase dialog box

7. Enter your passphrase and then click **OK**.

WARNING: If you do not send your email immediately but instead store it in your outbox, you should be aware that when using some email applications the information is not encrypted until the email is actually transmitted. Before queuing encrypted messages you should check to see if your application does in fact encrypt the messages in your outbox. If it does not, you can use PGPmenu option to encrypt your messages before queuing them in the outbox.

To encrypt and sign text using PGPtools

1. Copy the text that you want to encrypt and sign to the clipboard.
2. Click on the **Encrypt**, **Sign**, or **Encrypt and Sign** button in PGPtools.

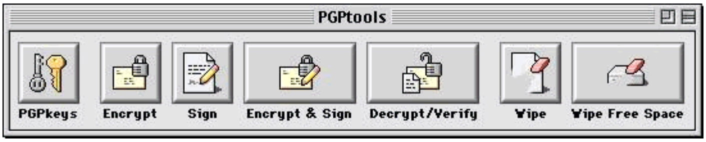


Figure 4-3. PGPtools window

The **PGP Key Select File(s)** dialog box appears.

3. Click the **Clipboard** button.

   The **PGP Key Recipients** dialog box appears (Figure 4-1).

4. Drag the public keys for those who are to receive a copy of the encrypted email message into the **Recipients** list box. You can also double-click any of the keys to move them from one area of the screen to the other.

The **Validity** icon indicates the minimum level of confidence that the public keys in the **Recipient** list are valid. This validity is based on the signatures associated with the key. See Chapter 6, "Managing Keys and Setting PGP Options," for details.

5. You can choose from the following encryption options depending on the type of data you are encrypting:

   - **Secure Viewer.** Select this option to protect the data from TEMPEST attacks upon decryption. If you select this option, the decrypted data is displayed in a special TEMPEST attack prevention font that is unreadable to radiation capturing equipment. For more information about TEMPEST attacks, see "Vulnerabilities" on page 246.

     NOTE: The Secure Viewer option may not be compatible with previous versions of PGP. Files encrypted with this option enabled can be decrypted by previous versions of PGP, however this feature may be ignored.

   - **Conventional Encrypt.** Select this option to use a common passphrase instead of public key encryption. If you select this option, the file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you will be asked to choose.

   - **Self Decrypting Archive.** Select this option to create a self decrypting executable file. If you select this option, the file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you are asked to choose. The resulting executable file can be decrypted by simply double-clicking on it and entering the appropriate passphrase. This option is especially convenient for users who are sending encrypted files to people who do not have PGP software installed. Note that sender and recipient must be on the same platform.

     You can also use this feature without a passphrase to create compact Self-Extracting Archives (SEA) which are not encrypted. The resulting archives run on both PowerPC and 68K Macs.

6. Click **OK** to encrypt and sign your mail.

   If you have elected to sign the encrypted data, the **Signing Key Passphrase** dialog box appears, as shown in Figure 4-2, requesting your passphrase before the mail is sent.

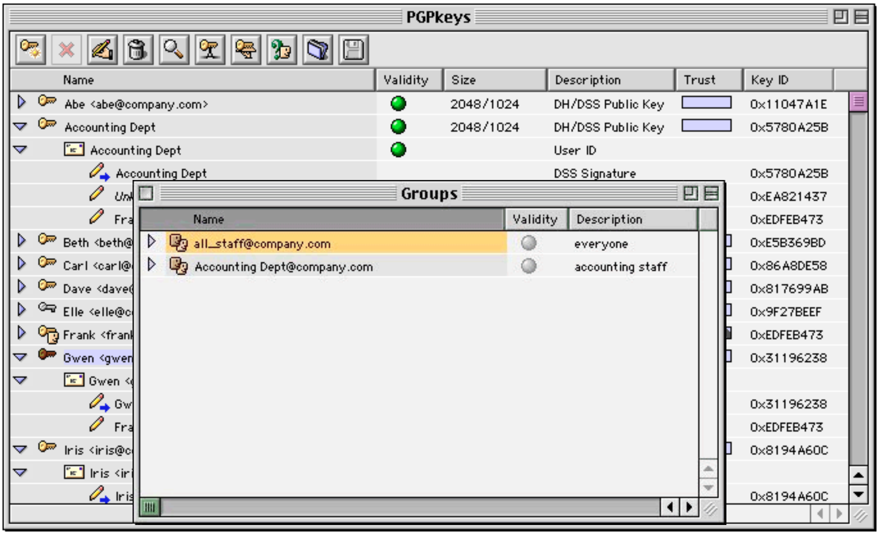
7. Enter your passphrase and then click **OK**.

8. Paste the text into your email message, then send the message.

Encrypting email to groups of recipients

You can use PGP to create group distribution lists. For example, if you want to send encrypted mail to 10 people at engineering@company.com, you could create a distribution list with that name. The **Groups** menu in PGPkeys contains the **Show Groups** option that toggles the display of the **Groups** window in PGPkeys. The **Groups List** window is displayed as in Figure 4-4.

☐ **NOTE:** If you intend to encrypt information to all members of an existing email distribution list, you must create a PGP group by the same name as, and including the same members as, the email distribution list. For example, if there is a staff@company.comlist set up in your email application, you must create a staff@company.com group in PGP.



PGPkeys

Name	Validity	Size	Description	Trust	Key ID
Abe <abe@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x11047A1E
Accounting Dept	●	2048/1024	DH/DSS Public Key	▭	0x5780A25B
Accounting Dept	●		User ID		
Accounting Dept			DSS Signature		0x5780A25B
Unk					0xEA821437
Fra					0xEDFEB473
Beth <beth@					0xE5B369BD
Carl <carl@					0x86A8DE58
Dave <dave@					0x817699AB
Elle <elle@c					0x9F27BEEF
Frank <frank					0xEDFEB473
Gwen <gwen					0x31196238
Gwen <					
Gw					0x31196238
Fra					0xEDFEB473
Iris <iris@c					0x8194A60C
Iris <iri					
Iris					0x8194A60C

Groups

Name	Validity	Description
all_staff@company.com	○	everyone
Accounting Dept@company.com	○	accounting staff

Figure 4-4. PGPkeys with Groups window

Working with distribution lists

Use the Groups feature to create distribution lists and to edit the list of people to whom you want to send encrypted email.

To create a group (distribution list)

1. Choose **New Group** from the **Groups** menu.
2. Enter a name for the group distribution list. Optionally, enter a group description. For example, you can name the group "everyone@company.com" with a description of "All employees."
3. Click **OK** to create the distribution list.

   The group distribution list is added to your keyring and can be viewed in the **Groups** window.

To add members to a distribution list

1. In the PGPkeys window, select the users or lists you want to add to your distribution list.
2. Drag the users from the PGPkeys window to the desired distribution list in the **Groups** window.

   ❐ **NOTE:** Members in a distribution list can be added to other distribution lists.

To delete members from a distribution list

1. Within the distribution list, select the member to be deleted.
2. Press the DELETE key.

   PGP asks you to confirm your choice.

To delete a distribution list

1. Select the distribution list to be deleted from the **Groups** window.
2. Press the DELETE key.

To add a distribution list to another distribution list

1. Select the distribution list that you want to add to another list.
2. Drag the selected list into the list to which it will be added.

Sending encrypted and signed email to distribution lists

You can send encrypted email to groups of recipients once your PGP distribution lists are created. See “Working with distribution lists” on page 80 for more information about creating and editing distribution lists.

To send encrypted and signed email to a distribution list

1. Address the mail to your mail distribution list.

   The name of your encryption distribution list must correspond to the name of the email distribution list.

2. Use your email application to compose your email message just as you normally would.
3. When you have finished composing the text of your email message, open the PGPmenu by clicking the PGP lock icon (🔒) on your email applications menu bar. Choose one of these options from the PGPmenu

   - **Encrypt.** Select this option to only encrypt the text of your message before sending.
   - **Sign.** Select this option to only sign your message before sending.

4. **Encrypt and Sign.** Select this option to both encrypt and sign your message before sending.

   The **PGP Key Recipients** dialog box appears (Figure 4-1). You can select the recipient’s public keys for the text you are encrypting or signing. The options available are described in “To encrypt and sign with supported email applications” on page 74.

5. Send the message.

Decrypting and verifying email

The quickest and easiest way to decrypt and verify the email sent to you is with an application supported by the PGP plug-ins. Although the procedure varies slightly between different email applications, when you are using an email application supported by the plug-ins, you can perform the decryption and verification operations by clicking the envelope icon in the message or your application’s toolbar. In some cases you may need to select **Decrypt/Verify** from the menu in your email application. In addition, if you are using an application that supports the PGP / MIME standard, you can decrypt and verify your email messages as well as any file attachments by clicking an icon attached to your message.

If you are using an email application that is not supported by the PGP plug-ins, you will decrypt and verify your email messages via PGPmenu. In addition, if your email includes encrypted file attachments, you must decrypt them separately via PGPtools or PGPmenu.

To decrypt and verify from supported email applications

1. Open your email message just as you normally do.

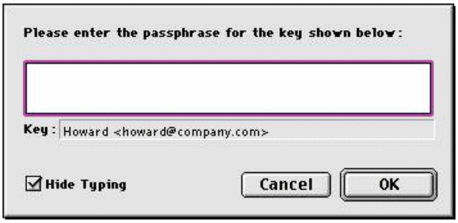
   You will see a block of unintelligible ciphertext in the body of your email message.

2. Copy the cipher text to the clipboard.

3. To decrypt and verify the message, click the locked envelope icon (✉).

   To decrypt and verify attached files, decrypt them separately using PGPtools or PGPmenu.

   The **PGP Enter Passphrase** dialog box appears, as shown in Figure 4-5, asking you to enter your passphrase.



Please enter the passphrase for the key shown below:

Key: Howard <howard@company.com>

☑ Hide Typing

Figure 4-5. Signing Key Passphrase dialog box

4. Enter your passphrase, then click **OK**.

The message is decrypted. If it has been signed and you have the senders public key, a message appears indicating whether the signature is valid.

If the message is encrypted with the **Secure Viewer** option enabled, an advisory message appears. Click **OK** to continue. The decrypted message appears on a secure PGP screen in a special TEMPEST attack prevention font.

5. You can save the message in its decrypted state, or you can save the original encrypted version so that it remains secure.

□ **NOTE:** Messages encrypted with the **Secure Viewer** option enabled cannot be saved in their decrypted state.

To decrypt and verify from non-supported email applications

1. Open your email message just as you normally do.

   You will see a block of unintelligible ciphertext in the body of your email message.

2. In PGPmenu, select **Decrypt/Verify**.

   If the email message includes encrypted file attachments, decrypt them separately with PGPtools or PGPmenu.

   The **PGP Enter Passphrase** dialog box appears, as shown in Figure 4-5, asking you to enter your passphrase.

3. Enter your passphrase, then click **OK**.

   The message is decrypted. If it has been signed, a message appears indicating whether the signature is valid.

   If the message is encrypted with **Secure Viewer** enabled, an advisory message appears. Click **OK** to continue. The decrypted message appears on a secure PGP screen in a special TEMPEST attack prevention font.

4. You can save the message in its decrypted state, or you can save the original encrypted version so that it remains secure.

□ **NOTE:** Messages encrypted with the **Secure Viewer** option enabled cannot be saved in their decrypted state.

Using PGP for Secure File Storage 5

This chapter describes how to use PGP to securely maintain files. It describes how to use PGP to encrypt, decrypt, sign and verify files either for email or for secure storage on your computer. It also describes the PGP Wipe and Free Space Wiper functions, which delete files by erasing their contents completely from your computer.

Using PGP to encrypt and decrypt files

You can use PGP to encrypt and sign files to use as email attachments. You can also use the techniques described in this chapter to encrypt and sign files so that you can store them securely on your computer.

Using the PGPmenu to encrypt and sign

Use the PGPmenu to send an encrypted file as an attachment with your email message, or to encrypt a file to protect it on your computer.

To encrypt and sign using PGPmenu

1. In the Finder select the file or files that you want to encrypt.

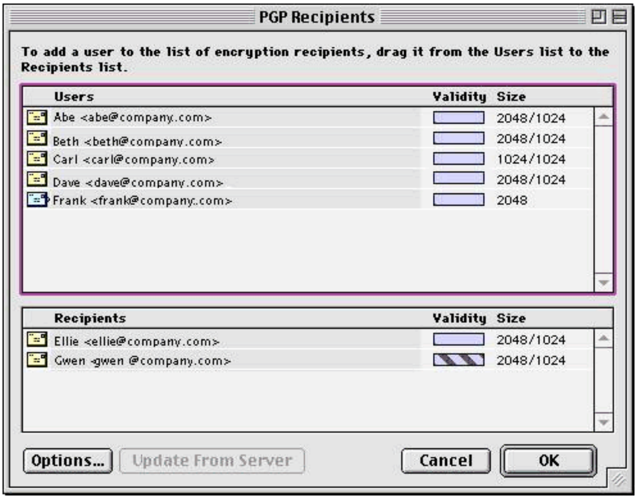
2. Choose one of the following options from PGPmenu or for Mac OS 8 users, choose from the PGPcontextmenu, which is accessed by holding down the Control key while you select a file:

   - **Encrypt.** Select this option to only encrypt the file or files you selected.

   - **Sign.** Select this option to only sign the file or files you selected.

   - **Encrypt and Sign.** Select this option to both encrypt and sign the file or files you selected.

   The **PGP Key Selection** dialog box appears, as shown in Figure 5-1.



PGP Recipients

To add a user to the list of encryption recipients, drag it from the Users list to the Recipients list.

Users	Validity	Size
Abe <abe@company.com>	▭	2048/1024
Beth <beth@company.com>	▭	2048/1024
Carl <carl@company.com>	▭	1024/1024
Dave <dave@company.com>	▭	2048/1024
Frank <frank@company.com>	▭	2048

Recipients	Validity	Size
Ellie <ellie@company.com>	▭	2048/1024
Gwen <gwen @company.com>	▨	2048/1024

Figure 5-1. PGP Recipients dialog box

You can select the recipient’s public keys for the file you are encrypting or signing.

3. Select the public keys by dragging them to the **Recipients** list, then click **Options** to specify encryption settings **OK**.

   You can choose from the following encryption options depending on the type of data you are encrypting:

   - **Text Output.** When sending files as attachments with some email applications, you may need to select the **Text Output** checkbox to save the file as ASCII text. This is sometimes necessary in order to send a binary file using older email applications. Selecting this option increases the size of the encrypted file by about 30 percent.
   - **MacBinary**.
     - **Yes.** This is the recommended option for all encryptions when sending to another user of PGP Version 5.5 or above on any platform. This means that Mac OS users will receive the exact file that was intended.
     - **No.** Select this option when sending encrypted files to a PC using an older version of PGP if you know that the file you are sending can be read by Windows applications when no MacBinary is used.
     - **Smart.** Select this option when communicating with users who are not using PGP versions 5.5 or above.

- **Wipe Original.** Select this checkbox to overwrite the original document that you are encrypting, so that your sensitive information is not readable by anyone who can access your hard disk.
- **Secure Viewer.** Select this checkbox to protect text from TEMPEST attacks upon decryption. If you select this option, the data is displayed in a special TEMPEST attack prevention font that is unreadable to radiation capturing equipment upon decrypting. For more information about TEMPEST attacks, see "Vulnerabilities" on page 234.

  NOTE: This option is only available when encrypting text or text files.

- **Conventional Encrypt.** Select this checkbox to rely on a common passphrase rather than on public key cryptography. The file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you are asked to choose.
- **Self Decrypting Archive.** Select this checkbox to create a self decrypting executable file. If you select this option, the file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you are asked to choose. The resulting executable file can be decrypted by simply double-clicking on it and entering the appropriate passphrase. This option is especially convenient for users who are sending encrypted files to people who do not have PGP software installed. Note that sender and recipient must be on the same platform.

  If you select this checkbox, you can also create non-encrypted self-extracting archive. To create a self-extracting archive with PGP, do not provide passphrase and click **OK**. The resulting executable file will have a .SEA extension.

If you are signing the files, you are asked to supply your passphrase.

After encryption, if you look in the folder where the original file was located, you will find a file with the specified name represented by one of four icons:

encrypted with standard output

encrypted with text output

self decrypting archive output

self extracting archive output

If you are encrypting or signing a folder, the output may be in a new folder, depending on the options you selected.

Using PGPtools to encrypt and sign

To encrypt and sign using PGPtools

1. Open PGPtools.
2. In the Finder, select the file or files that you want to encrypt.

   You can select multiple files, but you must encrypt and sign each of them individually.
3. Drag the file(s) onto the **Encrypt**, **Sign**, or **Encrypt** and **Sign** button PGPtools.

   The **PGP Recipients** dialog box appears, as shown in Figure 5-1.
4. Select the public keys by dragging them to the **Recipients** list.
5. You can choose from the following encryption options depending on the type of data you are encrypting:

   - **Text Output.** When sending files as attachments with some email applications, you may need to select the **Text Output** checkbox to save the file as ASCII text. This is sometimes necessary in order to send a binary file using older email applications. Selecting this option increases the size of the encrypted file by about 30 percent.
   - **MacBinary**.
     - **Yes.** This is the recommended option for all encryptions when sending to another user of PGP Version 5.5 or above on any platform. This means that Mac OS users will receive the exact file that was intended.
     - **No.** Select this option when sending encrypted files to a PC using an older version of PGP if you know that the file you are sending can be read by Windows applications when no MacBinary is used.
     - **Smart.** Select this option when communicating with users who are not using PGP versions 5.5 or above.
   - **Wipe Original.** Select this checkbox to overwrite the original document that you are encrypting, so that your sensitive information is not readable by anyone who can access your hard disk.

- **Secure Viewer.** Select this checkbox to protect text from TEMPEST attacks upon decryption. If you select this option, the data is displayed in a special TEMPEST attack prevention font that is unreadable to radiation capturing equipment upon decrypting. For more information about TEMPEST attacks, see "Vulnerabilities" on page 246.

  NOTE: This option is only available when encrypting text or text files.

- **Conventional Encrypt.** Select this checkbox to rely on a common passphrase rather than on public key cryptography. The file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you will be asked to choose.

- **Self Decrypting Archive.** Select this checkbox to create a self decrypting executable file. If you select this option, the file is encrypted using a session key, which encrypts (and decrypts) using a passphrase that you are asked to choose. The resulting executable file can be decrypted by simply double-clicking on it and entering the appropriate passphrase. This option is especially convenient for users who are sending encrypted files to people who do not have PGP software installed. Note that sender and recipient must be on the same platform.

  If you select this checkbox, you can also create a non-encrypted self-extracting archive. To create a self-extracting archive with PGP, do not provide a passphrase and click **OK**. The resulting executable file will have a .SEA extension.

6. Click **OK**.

   If you are signing the files, you are asked to supply your passphrase.

   After encryption, if you look in the folder where the original file was located, you will find a file with the specified name represented by one of four icons:

encrypted with standard output

encrypted with text output

self decrypting archive output

self extracting archive output

   If you are encrypting or signing a folder, the output may be in a new folder, depending on the options you selected.

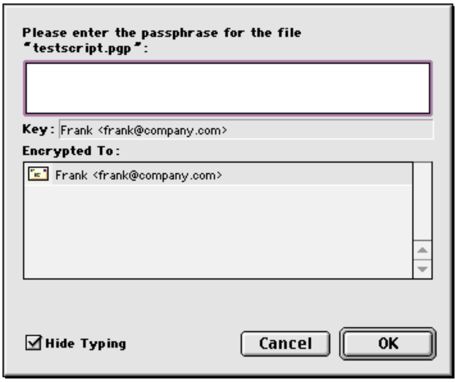
Using PGPmenu to decrypt and verify

If the email you receive has file attachments, and you are not using a PGP / MIME-compliant email application, you must decrypt them from the Finder.

To decrypt and verify files using PGPmenu

1. In the Finder, select the file or files that you want to decrypt and verify.
2. Choose **Decrypt/Verify** from PGPmenu.

   The passphrase dialog box appears, as shown in Figure 5-2.



Please enter the passphrase for the file "testscript.pgp":

Key: Frank <frank@company.com>

Encrypted To:

Frank <frank@company.com>

☑ Hide Typing

Figure 5-2. Passphrase dialog box

3. Enter your passphrase and then click **OK**.

   The file is decrypted. If it has been signed, a message appears indicating whether the signature is valid.

   If the text file is encrypted with **Secure Viewer** enabled, an advisory message appears. Click **OK** to continue. The decrypted text appears on a secure PGP screen in a special TEMPEST attack prevention font.

4. You can save the message in its decrypted state, or you can save the original encrypted version so that it remains secure.

   NOTE: Messages encrypted with the **Secure Viewer** option enabled cannot be saved in their decrypted state. They are only viewable on the secure PGP screen after decryption.

Using PGPtools to decrypt and verify

To decrypt and verify using PGPtools

1. In Finder, select the file or files that you want to decrypt.
2. Drag the file onto the **Decrypt/Verify** button in PGPtools.

   The **PGP Enter Passphrase** dialog box appears, as shown in Figure 5-2, asking you to enter your passphrase.
3. Enter your passphrase and then click **OK**.

   If the file is signed, a message appears indicating whether the signature is valid.

   If the text file is encrypted with **Secure Viewer** enabled, an advisory message appears. Click **OK** to continue. The decrypted text appears on a secure PGP screen in a special TEMPEST attack prevention font.
4. You can save the message in its decrypted state, or you can save the original encrypted version so that it remains secure.

   NOTE: Messages encrypted with the **Secure Viewer** option enabled cannot be saved in their decrypted state. They are only viewable on the secure PGP screen after decryption.

Signing and decrypting files with a split key

Once a key is split among multiple shareholders, attempting to sign or decrypt with it will cause PGP to automatically attempt to rejoin the key. There are two ways to rejoin the key, locally and remotely.

To rejoin key shares locally requires the shareholders presence at the rejoining computer. Each shareholder is required to enter the passphrase for their key share.

To rejoin key shares remotely requires the remote shareholders to authenticate and decrypt their keys before sending them over the network. PGP’s Transport Layer Security (TLS) provides a secure link to transmit key shares which allows multiple individuals in distant locations to securely sign or decrypt with their key share.

IMPORTANT: Before receiving key shares over the network, you should verify each shareholder’s fingerprint and sign their public key to ensure that their authenticating key is legitimate. To learn how to verify a keypair, see “Verify with a digital fingerprint” on page 74.

To rejoin a split key

1. Contact each shareholder of the split key. To rejoin a key shares locally, the shareholders of the key must be present.

   To collect key shares over the network, ensure that the remote shareholders are prepared to send their key share file. Remote shareholders must have:

   – their key share file and password

   – a public key (for authentication to the computer that is collecting the key shares)

   – a network connection

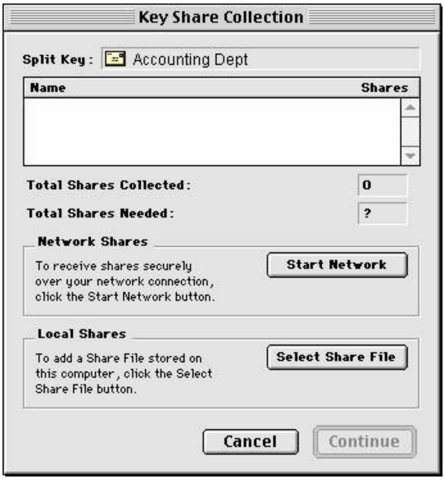
   – the IP address or Domain Name of the computer that is collecting the key shares

2. At the rejoining computer, use the Finder to select the file(s) that you want to sign or decrypt with the split key.

   The **PGP Enter Passphrase for Selected Key** dialog box appears with the split key selected.

3. Click **OK** to reconstitute the selected key.

   The **Key Share Collection** dialog box appears, as shown in Figure 5-3.



Key Share Collection

Split Key : Accounting Dept

Total Shares Collected : 0

Total Shares Needed : ?

To receive shares securely over your network connection, click the Start Network button.

To add a Share File stored on this computer, click the Select Share File button.

Figure 5-3. Key Share Collection dialog box

4. Do one of the following:

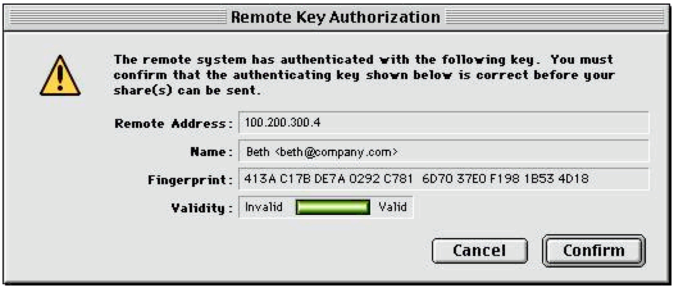
- **If you are collecting the key shares locally**, click **Select Share File** and then locate the share files associated with the split key. The share files can be collected from the hard drive, a floppy disk, or a mounted drive. Continue with Step 5.
- **If you are collecting key shares over the network**, click **Start Network**.

  The **Passphrase** dialog box opens. In the **Signing Key** box, select the keypair that you want to use for authentication to the remote system and enter the passphrase. Click **OK** to prepare the computer to receive the key shares.

  The status of the transaction is displayed in the **Network Shares** box. When the status changes to “Listening,” the PGP application is ready to receive the key shares.

  At this time, the shareholders must send their key shares. To learn how to send key shares to the rejoining computer, see “To send your key share over the network” on page 94.

  When a key is received, the **Remote Authentication** dialog box appears, as shown in Figure 5-4.



Remote Key Authorization

The remote system has authenticated with the following key. You must confirm that the authenticating key shown below is correct before your share(s) can be sent.

Remote Address: 100.200.300.4

Name: Beth <beth@company.com>

Fingerprint: 413A C17B DE7A 0292 C781 6D70 37E0 F198 1B53 4D18

Validity: Invalid Valid

Figure 5-4. Remote Authentication dialog box

If you have not signed the key that is being used to authenticate the remote system, the key will be considered invalid. Although you can rejoin the split key with an invalid authenticating key, it is not recommended. You should verify each shareholder’s fingerprint and sign their public key to ensure that the authenticating key is legitimate.

Click **Confirm** to accept the share file.

5. Continue collecting key shares until the value for **Total Shares Collected** matches the value for **Total Shares Needed** in the **Key Shares Collection** dialog box.
6. Click **OK**.

The file is signed or decrypted with the split key.

To send your key share over the network

1. When you are contacted by the person who is rejoining the split key, make sure that you have these items:

   – your key share file and password

   – your keypair (for authentication to the computer that is collecting the key shares)

   – a network connection

   – the IP address or Domain Name of the rejoining computer collecting the key shares

2. Choose Send Share File from the PGPkeys File menu.

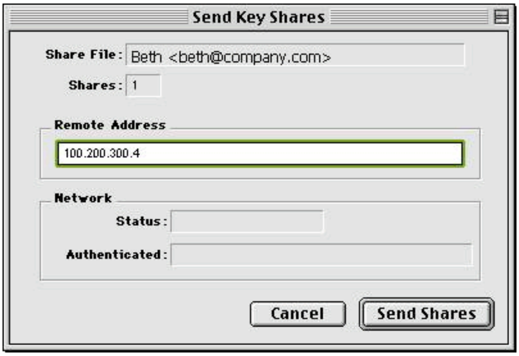
   The **Select Share File** dialog box appears.

3. Locate your key share and then click **Open**.

   The **PGP Enter Passphrase** dialog box appears.

4. Enter your passphrase and then click **OK**.

   The **Send Key Shares** dialog box appears, as shown in Figure 5-5.



Send Key Shares

Share File: Beth <beth@company.com>

Shares: 1

Remote Address

100.200.300.4

Network

Status:

Authenticated:

Figure 5-5. Send Key Shares dialog box

5. Enter the IP address or the Domain Name of the rejoining computer in the **Remote Address** text box, then click **Send Shares**.

   The status of the transaction is displayed in the **Network Status** box. When the status changes to “Connected,” you are asked to authenticate yourself to the rejoining computer.

The **Remote Authentication** dialog box appears asking you to confirm that the remote computer is the one to whom you want to send your key share.

6. Click **Confirm** to complete the transaction.

   After the remote computer receives your key shares and confirms the transaction, a message box appears stating that the shares were successfully sent.

7. Click **OK**.

8. Click **Done** in the **Key Shares** window when you have completed sending your key share.

Using PGP Wipe to delete files

The **Wipe** option on PGPmenu deletes files and their contents. The **Wipe** feature is a secure way of permanently removing a file and its contents from the hard drive of your computer. When you delete a file normally by placing it in the Trash, the name of the file is removed from the file directory, but the data in the file stays on the disk. **Wipe** removes all traces of a file's data so that no one can use a software tool to recover the file.

To permanently delete a file

1. In the Finder, select the file or files that you want to wipe.

2. Choose **Wipe** from PGPmenu or for Mac OS 8 users, choose from the PGPcontextmenu, which is accessed by holding down the CONTROL KEY while you select a file.

3. Click **OK** to permanently erase the file.

   To stop wiping the file before the task is completed, click **Cancel**.

   ❑ **NOTE:** Clicking **Cancel** during file wipe can leave remnants of the file behind.

To permanently delete a file using PGPtools

1. In the Finder, select the file or files that you want to wipe.

2. Drag the file onto the **Wipe** button ( ) in PGPtools.

   A confirmation dialog box appears.

3. Click **OK** to permanently erase the file.

   To stop wiping the file before the task is completed, click **Cancel**.

   NOTE: Clicking **Cancel** during file wipe can leave remnants of the file behind.

Even on systems with virtual memory, PGP correctly writes over all the contents of the file. It is worth noting that some application programs save the file prior to encrypting it and may have leave fragments of the file on your disk in locations which are no longer considered part of the file. For more information, see “Swap files or virtual memory” on page 237. You can use PGP Free Space Wiper to wipe all free space on your disk to solve this problem. See the next section for information about Free Space Wiper. Also, be aware that many programs automatically save files in progress, so there may be back-up copies of the file that you want to delete.

Using the PGP Free Space Wiper to clean free space on your disks

As you create and delete files on your computer, the data contained in those files remains on the drive. PGPtools can be used to securely wipe the data in a file before it is deleted to negate the possibility of the data ever being recovered.

Many programs create temporary files while you edit the contents of the documents. These files are deleted when you close the documents but the actual document data is left scattered about your drive. To help reduce the chance that your document’s data can later be recovered, Network Associates recommends that you securely wipe the free space on your drives as well as securely deleting sensitive documents.

To wipe free space on your disks

WARNING: Before running the PGP Free Space Wiper, file sharing must be turned off and all applications on the volume or disk that you want to wipe must be closed.

1. Open PGPtools.
2. Click the **Wipe Free Space** button ( ) in PGPtools.

   The **PGP Free Space Wiper Welcome** screen appears.

3. Read the information carefully, then click **Next** to advance to the next dialog box.

   The PGP Free Space Wiper prompts you to select the volume you want to wipe and the number of passes you want to perform.

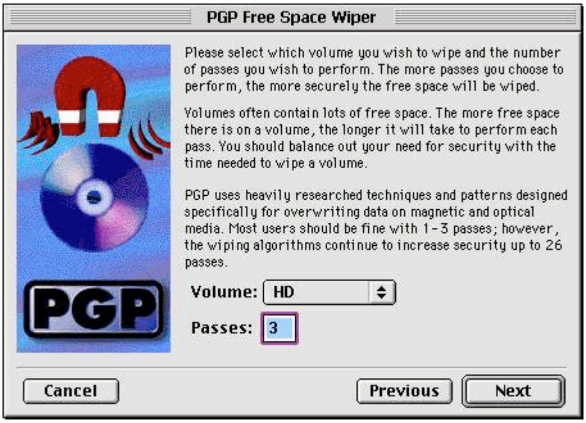
4. In the **Volume** box, select the disk or volume that you want PGP to wipe. Then, select the number of passes that you want PGP to perform. The recommended guidelines are:

   - 3 passes for personal use.
   - 10 passes for commercial use.
   - 18 passes for military use.
   - 26 passes for maximum security.

   NOTE: Commercial data recovery companies have been known to recover data that has been over written up to 9 times. PGP uses highly sophisticated patterns during each wipe to ensure that your sensitive data cannot be recovered.

5. Click **Next** to continue.

   The **Perform Wipe** dialog box opens, as shown in Figure 5-6, and displays statistical information about the drive or volume you selected



Please select which volume you wish to wipe and the number of passes you wish to perform. The more passes you choose to perform, the more securely the free space will be wiped.

Volumes often contain lots of free space. The more free space there is on a volume, the longer it will take to perform each pass. You should balance out your need for security with the time needed to wipe a volume.

PGP uses heavily researched techniques and patterns designed specifically for overwriting data on magnetic and optical media. Most users should be fine with 1-3 passes; however, the wiping algorithms continue to increase security up to 26 passes.

Volume: HD

Passes: 3

Figure 5-6. Free Space Wiper (Perform Wipe dialog box)

6. Click the **Begin Wipe** button to start freespace wiping your disk or volume.

   The PGP Free Space Wiper scans and then wipes leftover fragments from your disk or volume.

7. When the wipe session ends, click **Finish**.

WARNING: Clicking **Cancel** during file wipe can leave remains of the file on your computer.

Scheduling Free Space Wiper

You can use AppleScript to schedule periodic secure wiping of freespace on your disks The AppleScript dictionary for this is located in PGPtools.

To schedule freespace wiping

1. Open the Script Editor and choose **New Script** from the **File** menu.

   TIP: The Script Editor is located in the **AppleScript** folder in the **Apple Extras** folder on your startup disk.

   The new script window appears.

2. Enter a description for the script in the **Description** text box—for example, “Scheduled Free Space Wipe - 3 passes.”

3. Enter the following information into the AppleScript text box:

```
with timeout of X seconds

tell application “PGPtools”

activate

wipe free space volume “Your Macintosh HD” passes Y

end tell

end timeout
```

   Where “*Your Macintosh HD*” is, enter the name of the hard disk in which you want to wipe freespace.

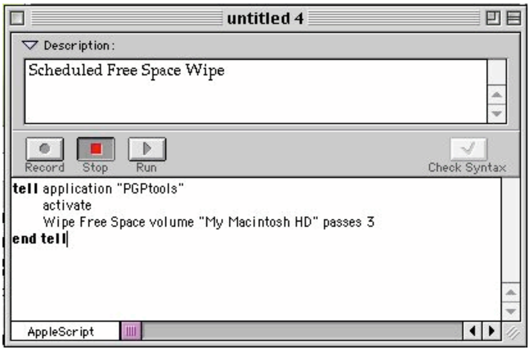
   Where *X* is, enter the amount of time you want to allocate for the Free Space Wipe session. Wiping may take a long time to complete, depending on the amount of passes you specified and the speed of the computer you are using. If you are unsure, enter a high timeout number (for example, 60,000 seconds).

   Where *Y* is, passes that you want Free Space Wipe to perform. For guidelines, see Step 4 on page 97.

4. Click the **Check Syntax** button in the script window and navigate to the PGPtools application.

> **TIP:** Use the PGPtools Dictionary to view a list of acceptable syntax. To open the PGPtools Dictionary, choose **Open Dictionary** from the **File** menu and navigate to PGPtools in your PGP folder.

When the syntax check is complete, the Script Editor formats the script, as shown in Figure 5-7.



Scheduled Free Space Wipe

```
tell application "PGPtools"
     activate
     Wipe Free Space volume "My Macintosh HD" passes 3
end tell
```

Figure 5-7. Script Editor window

5. Choose **Save** from the **File** menu.
6. When the **Save** dialog box appears select a location and specify a name for the script.
7. Open the **Kind** pop-up menu and choose **Application**.
8. Click **Save**.

> **TIP:** You can customize your script to wipe additional hard disks or to run different or multiple wipe schedules.

You can ensure periodic wiping by saving the compiled script in your Shutdown Items folder in the System folder.

Managing Keys and Setting PGP Preferences 6

This chapter explains how to examine and manage the keys stored on your keyrings. It also describes how to set your options to suit your particular computing environment.

Managing your keys

The keys you create, as well as those you collect from others, are stored on keyrings, which are essentially files stored on your hard drive or on a floppy disk. Normally your private keys are stored in a file named PGP Private Keys and your public keys are stored in another file named PGP Public Keys. These files are usually located in the PGP Keyrings folder.

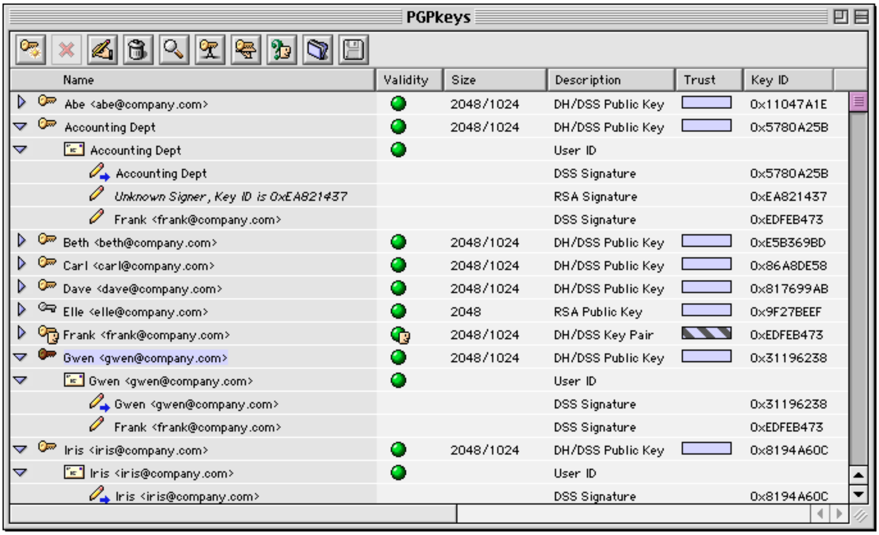
NOTE: As a result of your private key being encrypted automatically and your passphrase being uncompromised, there is no danger in leaving your keyrings on your computer. However, if you are not comfortable storing your keys in the default location, you can choose a different filename or location. For details, see "Setting PGP preferences," later in this chapter.

Occasionally, you may want to examine or change the attributes associated with your keys. For instance, when you obtain someone's public key, you might want to identify its type (either RSA or Diffie-Hellman / DSS), check its fingerprint, or determine its validity based on any digital signatures included with the key. You may also want to sign someone's public key to indicate that you believe it is valid, assign a level of trust to the key's owner, or change a passphrase for your private key. You may even want to search a key server for someone's key. You perform all of these key-management functions from PGPkeys.

The PGPkeys window

To open the PGPkeys window, choose PGPkeys from PGPmenu or double-click 🔑 in the PGP program folder.

The **PGPkeys** window, as shown in Figure 6-1, displays the keys you have created for yourself, as well as any public keys you have added to your public keyring.



PGPkeys

Name	Validity	Size	Description	Trust	Key ID
Abe <abe@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x11047A1E
Accounting Dept	●	2048/1024	DH/DSS Public Key	▭	0x5780A25B
Accounting Dept	●		User ID		
Accounting Dept			DSS Signature		0x5780A25B
Unknown Signer, Key ID is 0xEA821437			RSA Signature		0xEA821437
Frank <frank@company.com>			DSS Signature		0xEDFEB473
Beth <beth@company.com>	●	2048/1024	DH/DSS Public Key	▭	0xE5B369BD
Carl <carl@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x86A8DE58
Dave <dave@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x817699AB
Elle <elle@company.com>	●	2048	RSA Public Key	▭	0x9F27BEEF
Frank <frank@company.com>	●	2048/1024	DH/DSS Key Pair	▨	0xEDFEB473
Gwen <gwen@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x31196238
Gwen <gwen@company.com>	●		User ID		
Gwen <gwen@company.com>			DSS Signature		0x31196238
Frank <frank@company.com>			DSS Signature		0xEDFEB473
Iris <iris@company.com>	●	2048/1024	DH/DSS Public Key	▭	0x8194A60C
Iris <iris@company.com>	●		User ID		
Iris <iris@company.com>			DSS Signature		0x8194A60C

Figure 6-1. PGPkeys window

A key and user icon (🔑) represent the private and public key pairs you have created for yourself, and single keys (🔑) represent the public keys you have collected from others. If you have more than one type of key, you will notice that RSA-type keys are silver keys and Diffie-Hellman / DSS keys are gold keys.

By clicking on the triangle at the left side of the key icon, you can expand the entries to reveal the user ID and email address for the owner of the key as represented by the envelope icons (✉). By clicking the triangle next to an envelope icon, you can see the signatures of any users who have certified the user ID. If you don't want to expand each key individually, simply select the keys of interest and then choose **Expand Selection** from the **Edit** menu.

PGPkeys attribute definitions

Some of the attributes associated with keys can be displayed in the main PGPkeys window. You can choose which attributes you want to make visible by selecting them in the **View** menu. For each selected item in the **View** menu, PGPkeys displays a column in the main window. If you want to change the order of these columns, click and drag the header of the column you want to move.

Table 6-1. PGPkeys attribute overview

Name

Shows an iconic representation of the key along with the user name and email address of the owner, and the names of the key's signers.

Validity

Indicates the level of confidence that the key actually belongs to the alleged owner. The validity is based on who has signed the key and how well you trust the signer(s) to vouch for the authenticity of a key. The public keys you sign yourself have the highest level of validity, based on the assumption that you only sign someone's key if you are totally convinced that it is valid. The validity of any other keys, which you have not personally signed, depends on the level of trust you have granted to any other users who have signed the key. If there are no signatures associated with the key, then it is not considered valid, and a message indicating this fact appears whenever you encrypt to the key.

Validity is indicated by either circle or bar icons, depending upon your Advanced **Preferences** "Display marginal validity level" setting (see "Setting advanced preferences" later in this chapter). If set, then validity appears as:

▭, an empty bar for invalid keys

◧, a half-filled bar for marginally valid keys

▬, a filled bar for valid keys that you do not own

▨, a striped bar for valid keys that you do own

If not set, then validity appears as:

●, a gray circle for invalid keys and marginally valid keys if the Advanced **Preferences** "Treat marginally valid keys as invalid" is set

🟢, a green circle for valid keys that you do not own

In a corporate environment, your security officer may sign users' keys with the Corporate Signing Key. Keys signed with the Corporate Signing Key are usually assumed to be completely valid. See Chapter 2, "Using PGP," for more information.

Size

Shows the number of bits used to construct the key. Generally, the larger the key, the less chance that it will ever be compromised. However, larger keys require slightly more time to encrypt and decrypt data than do smaller keys. When you create a Diffie-Hellman/DSS key, there is one number for the Diffie-Hellman portion and another number for the DSS portion. The DSS portion is used for signing, and the Diffie-Hellman portion for encryption.

Description Describes the type of information displayed in the **Name** column: key type, type of ID, or signature type.

Additional Decryption Key Shows whether the key has an associated Additional Decryption Key.

Key ID A unique identifying number associated with each key. This identification number is useful for distinguishing between two keys that share the same user name and email address.

Trust Indicates the level of trust you have granted to the owner of the key to serve as an introducer for the public keys of others. This trust comes into play when you are unable to verify the validity of someone’s public key for yourself and instead rely on the judgment of other users who have signed the key. When you create a key pair, they are considered implicitly trustworthy, as shown by the striping in the trust and validity bars, or by a green dot and user icon.

When you receive a public key that has been signed by another of the user’s keys on your public keyring, the level of authenticity is based on the trust you have granted to the signer of that key. You assign a level of trust, either Trusted, Marginal, or Untrusted, in the **Key Properties** dialog box.

Expiration Shows the date when the key will expire. Most keys are set to Never; however, there may be instances when the owner of a key wants it to be used for only a fixed period of time.

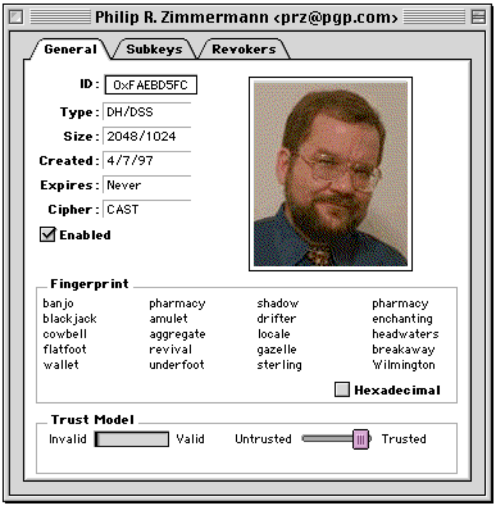
Creation Shows the date when the key was originally created. You can sometimes make an assumption about the validity of a key based on how long it has been in circulation. If the key has been in use for a while, it is less likely that someone will try to replace it because there are many other copies in circulation. Never rely on creation dates as the sole indicator of validity.

Examining a key’s properties

In addition to the general attributes shown in the **PGPkeys** window, you can also examine and change other key and subkey properties.

The **Key Properties** window includes the **General** panel, **Subkey** panel, and **Revokers** panel, each of which gives you necessary information about a person’s public key, or the ability to create, configure, edit, or delete attributes in your own public key. The following sections describe each element in more detail.

To access the properties for a particular key, select the desired key and then choose **Properties** from the **Keys** menu. The **Key Property** dialog box appears as shown in Figure 6-2.



ID: 0xFAEBD5FC
Type: DH/DSS
Size: 2048/1024
Created: 4/7/97
Expires: Never
Cipher: CAST
☑ Enabled

Fingerprint

banjo	pharmacy	shadow	pharmacy
blackjack	amulet	drifter	enchanting
cowbell	aggregate	locale	headwaters
flatfoot	revival	gazelle	breakaway
wallet	underfoot	sterling	Wilmington

☐ Hexadecimal

Trust Model

Figure 6-2. Key Property dialog box (General panel)

General Key Properties panel

To access the **General Key Properties** panel for a particular key, select the desired key and then choose **Properties** from the Keys menu.

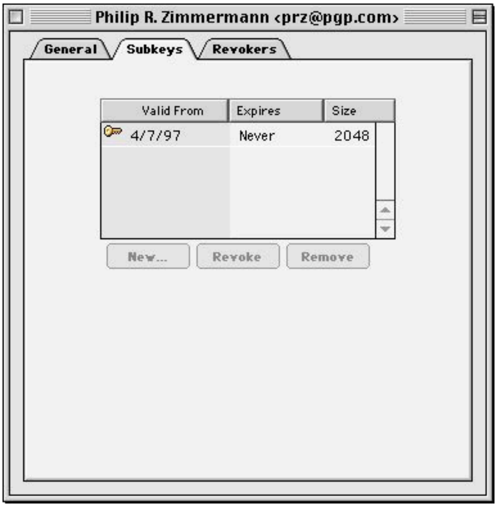
Refer to Table 6-2, “General Key Properties panel attributes,”for a description of each attribute available in the **General Key Properties** panel.

Table 6-2. General Key Properties panel attributes

Key ID	A unique identifying number associated with each key. This identification number is useful for distinguishing between two keys that share the same user name and email address.
Key Type	The key type, either RSA or Diffie-Hellman/DSS.
Key Size	The size of the key.
Created	The date when the key was created.
Expires	The date when the key expires. Owners specify this date when they create their keys, and the value is usually set to Never. However, some keys are set to expire on a particular date if the owner wants them to be used for a limited period of time.
Cipher	CAST, Triple DES, or IDEA. This is the “preferred” encryption algorithm by which the owner of the key requests that you encrypt to his public key. If this algorithm is allowed in your **Advanced Preferences**, it will be used whenever encrypting to this key.
Join	Opens the **Key Share Collection** dialog box. Available for split keys only. See “Signing and decrypting files with a split key” on page 94 for information about rejoining split keys.
Enabled	Indicates whether the key is currently enabled. When a key is disabled, it is dimmed in the PGPkeys window and is not available for performing any PGP functions except **Decrypt** and **Verify**. However, the key remains on your keyring and you can enable it again at any time. To enable or disable a key, select or clear the **Enabled** checkbox. (The checkbox is not visible for implicitly trusted keys.) This feature is useful for preventing seldom-used keys from cluttering up the **Key Selection** dialog box when you are sending encrypted email.
Change Passphrase	Changes the passphrase for a private key. If you ever think that your passphrase is no longer a secret, click this button to enter a new passphrase. It is a good idea to change your passphrase every 6 months or so. For instructions on changing your passphrase, see “Changing your Passphrase” later in this chapter.
Fingerprint	A unique identification number that is generated when the key is created. This is the primary means by which you can check the authenticity of a key. The best way to check a fingerprint is to have the owner read their fingerprint to you over the phone so that you can compare it with the fingerprint shown for your copy of their public key. The fingerprint can be viewed in two ways, in a unique list of words or in its hexadecimal format.
Hexadecimal	Displays the fingerprint as a unique series of hexadecimal numbers. By default, this option is disabled and the fingerprint is displayed as a unique series of words.
Trust Model	Indicates the validity of the key based on its certification and the level of trust you have in the owner to vouch for the authenticity of someone else’s public key. You set the trust level by sliding the bar to the appropriate level (Trusted, Marginal, or Untrusted). The bar is disabled for revoked, expired, and implicitly trusted keys.

Subkey properties window

To access the **Subkey Properties** panel for a particular key, select the desired key and then choose **Properties** from the **Keys** menu. The **Key Properties** dialog box appears, as shown in Figure 6-2 on page 105. Click the **Subkey** tab. The **Subkey** panel appears as shown in Figure 6-3.



Valid From	Expires	Size
4/7/97	Never	2048

Figure 6-3. Key Property dialog box (Subkey panel)

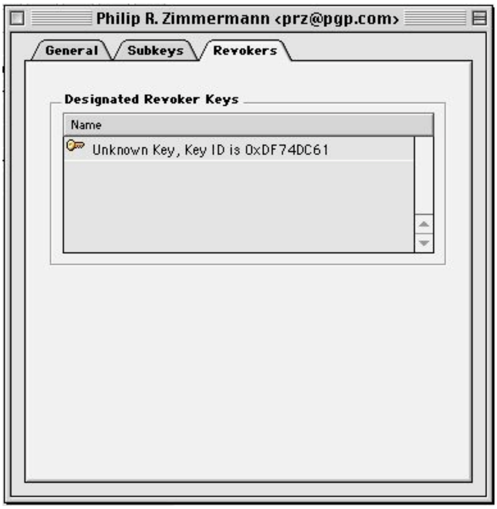
Refer to Table 6-2, "General Key Properties panel attributes," for a description of each attribute and task available in the **Subkey** panel.

Table 6-3. Subkey properties panel

Valid From	The date when the subkey becomes active.
Expires	The date when the subkey expires. Owners specify this date when they create their subkeys. Subkeys are usually active for a limited period of time.
Key Size	The size of the subkey.
New	Creates a new subkey. For information about creating a new subkey, see "Creating new subkeys" on page 49.
Revoke	Revokes the currently selected encryption subkey. After you revoke the subkey and redistribute your key, others will no longer be able to encrypt data to this subkey.
Remove	Permanently removes the currently selected encryption subkey. This procedure cannot be undone. Any data that is encrypted to the selected subkey can longer be decrypted. **TIP:** Use the Revoke option (described above) if you want to disable the subkey and update the key server. Once a subkey has been sent to the server, it cannot be removed.

Designated revoker window

To access the **Revokers** panel for a particular key, select the desired key and then choose **Properties** from the **Keys** menu. The **Key Properties** dialog box appears, as shown in Figure 6-2 on page 105. Click the **Revokers** tab. The **Revokers** panel appears as shown in Figure 6-3.



Philip R. Zimmermann <prz@pgp.com>

Unknown Key, Key ID is 0xDF74DC61

Figure 6-4. Key Property dialog box (Revokers panel)

The Revokers panel lists any keys that have the ability to revoke your PGP key. For instructions on adding a revoker to your key, "Adding a designated revoker" on page 52.

Specifying a default key pair

When encrypting messages or files, PGP gives you the option to additionally encrypt to a key pair that you specify as your default key pair. When you sign a message or someone's public key, PGP will use this key pair by default. Your default key pair is displayed in bold type to distinguish them from your other keys. If you have only one key pair on your keyring, it is automatically designated as your default key pair. If you have more than one key pair, you may want to specifically designate one pair as your default pair.

To specify your default key pair

1. Open PGPkeys.
2. Highlight the key pair you want to designate as your default key.
3. Choose **Set Default** from the **Keys** menu.

   The selected key pair is displayed in bold type, indicating that it is now designated as your default key pair.

Verifying someone's public key

In the past it was difficult to know for certain whether a key belonged to a particular individual unless that person physically handed the key to you on a floppy disk. Exchanging keys in this manner is not usually practical, especially for users who are located many miles apart.

There are several ways to check a key's fingerprint, but the safest is to call the person and have them read the fingerprint to you over the phone. Unless the person is the target of an attack, it is highly unlikely that someone would be able to intercept this random call and imitate the person you expect to hear on the other end. You can also compare the fingerprint on your copy of someone's public key to the fingerprint on their original key on a public server.

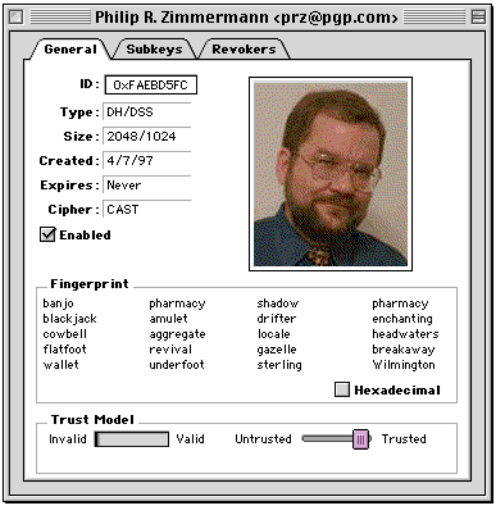
The fingerprint can be viewed in two ways, in a unique list of words or in its hexadecimal format

To check a public key with its digital fingerprint

1. Open PGPkeys.
2. Highlight the public key that you want to verify.

3. Choose **Properties** from the **Keys** menu or click to open the **Properties** dialog box.

   The **Properties** dialog box opens, as shown in Figure 6-5.



Philip R. Zimmermann <prz@pgp.com>

ID : 0xFAEBD5FC
Type : DH/DSS
Size : 2048/1024
Created : 4/7/97
Expires : Never
Cipher : CAST
☑ Enabled

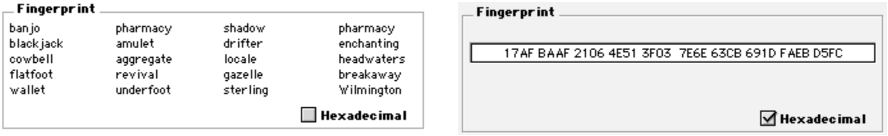
Fingerprint
banjo pharmacy shadow pharmacy
blackjack amulet drifter enchanting
cowbell aggregate locale headwaters
flatfoot revival gazelle breakaway
wallet underfoot sterling Wilmington
□ Hexadecimal

Trust Model
Invalid Valid Untrusted Trusted

Figure 6-5. PGP Properties dialog box

4. Use the series words or characters displayed in the **Fingerprint** text box to compare with the original fingerprint.

   By default, a word list is displayed in the **Fingerprint** text box (example shown in Figure 6-6). However, you can select the **Hexadecimal** checkbox to view the fingerprint in 20 hexadecimal characters (example shown in Figure 6-6).



Fingerprint
banjo pharmacy shadow pharmacy
blackjack amulet drifter enchanting
cowbell aggregate locale headwaters
flatfoot revival gazelle breakaway
wallet underfoot sterling Wilmington
□ Hexadecimal

Fingerprint
17AF BAAF 2106 4E51 3F03 7E6E 63CB 691D FAEB D5FC
☑ Hexadecimal

Word list view *Hexadecimal view*

Figure 6-6. Fingerprint text box

The word list in the fingerprint text box is made up of special authentication words that PGP uses and are carefully selected to be phonetically distinct and easy to understand without phonetic ambiguity.

The word list serves a similar purpose as the military alphabet, which allows pilots to convey information distinctly over a noisy radio channel. If you'd like to know more about the word hash technique and view the word list, see Appendix D, "Biometric Word Lists."

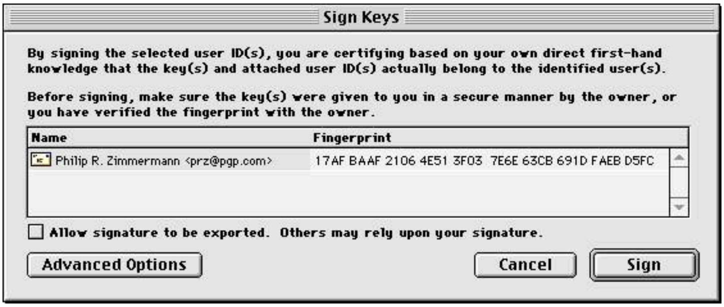
Signing someone's public key

When you create a set of keys, the keys are automatically signed using your public key. Similarly, once you are sure that a key belongs to the proper individual, you can sign that person's public key, indicating that you are sure it is a valid key. When you sign someone's public key, an icon associated with your user name is shown for that key.

To sign someone's public key

1. Open the PGPkeys window.
2. Highlight the public key that you want to sign.
3. Choose **Sign** from the **Keys** menu or click ✍ to open the **Sign Keys** dialog box.

   The **Sign Keys** dialog box appears (Figure 6-5) with the public key and fingerprint displayed in the text box.



Sign Keys

By signing the selected user ID(s), you are certifying based on your own direct first-hand knowledge that the key(s) and attached user ID(s) actually belong to the identified user(s).

Before signing, make sure the key(s) were given to you in a secure manner by the owner, or you have verified the fingerprint with the owner.

Name	Fingerprint
Philip R. Zimmermann <prz@pgp.com>	17AF BAAF 2106 4E51 3F03 7E6E 63CB 691D FAEB D5FC

□ Allow signature to be exported. Others may rely upon your signature.

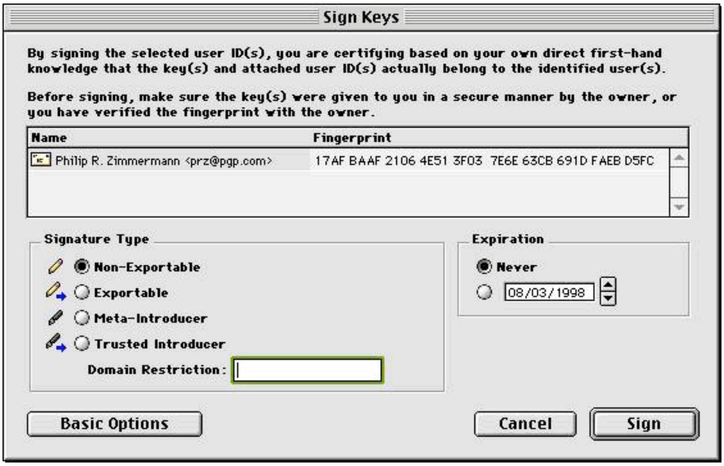
Figure 6-7. PGP Sign Keys dialog box (Basic Options)

4. Click the **Allow signature to be Exported** checkbox, to allow your signature to be exported with this key.

   An exportable signature is one that is allowed to be sent to servers and travels with the key whenever it is exported, such as by dragging it to an email message. The checkbox provides a shorthand means of indicating that you wish to export your signature.

Or

Click the Advanced Options button to configure options, such as signature type and signature expiration (Figure 6-8).



Sign Keys

By signing the selected user ID(s), you are certifying based on your own direct first-hand knowledge that the key(s) and attached user ID(s) actually belong to the identified user(s).

Before signing, make sure the key(s) were given to you in a secure manner by the owner, or you have verified the fingerprint with the owner.

Name	Fingerprint
Philip R. Zimmermann <prz@pgp.com>	17AF BAAF 2106 4E51 3F03 7E6E 63CB 691D FAEB D5FC

Signature Type

- ● Non-Exportable
- ○ Exportable
- ○ Meta-Introducer
- ○ Trusted Introducer

Domain Restriction:

Expiration

- ● Never
- ○ 08/03/1998

Figure 6-8. PGP Sign Keys dialog box (Advanced Options)

Choose a signature type to sign the public key with. Your options are:

- **Non-exportable.** Use this signature when you believe the key is valid but you don’t want others to rely on your certification. This signature type cannot be sent with the associated key to a key server, or exported in any way.

- **Exportable.** Use exportable signatures in situations where your signature is sent with the key to the key server so that others can rely on your signature and trust your keys as a result. This is equivalent to checking the **Allow signature to be exported** checkbox on the **Sign Keys** menu.

- **Meta-Introducer Non-Exportable.** Certifies that this key and any keys signed by this key with a Trusted Introducer Validity Assertion are fully trusted introducers to you. This signature type is non-exportable.

- **Trusted Introducer Exportable.** Use this signature in situations where you certify that this key is valid, and that the owner of the key should be completely trusted to vouch for other keys. This signature type is exportable. You can restrict the validation capabilities of the trusted introducer to a particular email domain.

5. If you want to limit the Trusted Introducer’s certificate validation capabilities to a single domain, enter the domain name in the Domain text box.
6. If you want to assign an expiration date to this signature, enter the date on which you want this signature to expire in the Date text box. Otherwise, the signature will never expire.
7. Click **OK**.

   The **Passphrase** dialog box appears.
8. Enter your passphrase, then click **OK**.

   An icon associated with your user name is now included with the public key that you just signed.

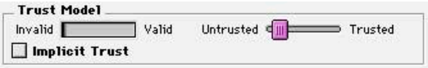
Granting trust for key validations

Besides certifying that a key belongs to someone, you can assign a level of trust to the user of the keys indicating how well you trust them to act as an introducer to others whose keys you may get in the future. This means that if you ever get a key from someone that has been signed by an individual whom you have designated as trustworthy, the key is considered valid even though you have not done the check yourself.

To grant trust for a key

1. Open PGPkeys.
2. Select the key for which you want to change the trust level.

   ❑ **NOTE:** You must sign the key before you can set the trust level for it. If you have not already signed the key, see “Validating the public key” on page 75 for instructions.
3. Choose **Properties** from the **Keys** menu or click to open the **Properties** dialog box, as shown in Figure 6-5.
4. Use the Trust Level sliding bar to choose the appropriate level of trust for the key pair.



□ Implicit Trust

Figure 6-9. Trust Level dialog box

5. Close the dialog box to accept the new setting.

If you give a key with a photo a high level of trust, PGP removes the red question mark from the photograph.

Disabling and enabling keys

Sometimes you may want to temporarily disable a key. The ability to disable keys is useful when you want to retain a public key for future use, but you don't want it cluttering up your recipient list every time you send mail.

To disable a key

1. Open PGPkeys.
2. Select the key you want to disable.
3. Select **Disable** in the **Keys** menu.

   The key is dimmed and is temporarily unavailable for use.

To enable a key

1. Open PGPkeys.
2. Select the key you want to enable.
3. Select **Enable** in the **Keys** menu.

   The key becomes visible and can be used as before.

Importing and Exporting Keys

Although you often distribute your public key and obtain the public keys of others by cutting and pasting the raw text from a public or corporate key server, you can also exchange keys by importing and exporting them as separate text files. For instance, someone could hand you a disk containing their public key, or you might want to make your public key available over an FTP server.

To import a key from a file

1. Open PGPkeys.
2. Choose **Import** from the **Keys** menu.

The **Import** dialog box appears.

3. Select the file that contains the key you want to import, then click **Open**.

   The **Import Selection** dialog box appears.

4. Select the key(s) that you want to import to your keyring, then click the **Import** button.

5. The imported key(s) appears in PGPkeys, where you can use it to encrypt data or to verify someone's digital signature.

To add a key from an email message

If a colleague sends you an email message with their key enclosed (as a block of text) you can add it to your keyring.

1. While the email message window is open, open PGPkeys.

2. Tile the two windows so that you can see part of PGPkeys behind the message window.

3. Select the key text, including the `BEGIN PGP PUBLIC KEY BLOCK` and `END PGP PUBLIC KEY BLOCK` text, and drag the text onto the PGPkeys window.

   The **Import Selection** dialog box appears.

4. Select the key(s) that you want to import to your keyring, then click the **Import** button.

5. The imported key(s) appears in PGPkeys, where you can use it to encrypt data or to verify someone's digital signature.

To export a key to a file

1. Open the PGPkeys window.

2. Select the key you want to export to a file.

3. Choose **Export** from the **Keys** menu.

   The **Export** dialog box appears.

4. Enter the name of the file or navigate to the file which you want the key to be exported and then click **Save**.

   The exported key is saved to the named file in the specified folder location.

You can also obtain your Pkcs-12 X.509 private keys by exporting them from your browser and dropping them into PGPkeys, or by choosing **Import** from the **Keys** menu.

Revoking a key

If the situation ever arises that you no longer trust your personal key pair, you can issue a revocation to the world telling everyone to stop using your public key. The best way to circulate a revoked key is to place it on a public key server.

To revoke a key

1. Open PGPkeys.
2. Select the key pair you want to revoke.
3. Choose **Revoke** from the **Keys** menu.

   The **Revocation Confirmation** dialog box appears.
4. Click **OK** to confirm your intent to revoke the selected key.

   The **PGP Enter Passphrase** dialog box appears.
5. Enter your passphrase, then click **OK**.

   When you revoke a key, it is crossed out with a red line to indicate that it is no longer valid.
6. Send the revoked key to the server so everyone will know not to use your old key.

Appointing a designated revoker

It is possible that you might forget your passphrase someday or lose your private key. In which case, you would never be able to use your key again, and you would have no way of revoking your old key when you create a new one. To safeguard against this possibility, you can appoint a third-party key revoker on your public keyring to revoke your key. The third-party you designate will be able to revoke your DH / DSS key, send it to the server and it will be just as if you had revoked it yourself.

To appoint a designated revoker

1. Open PGPkeys.
2. Select the key pair for which you want to designate a revoker.

3. Select **Add/Revoker** from the **Keys** menu.

   A dialog box opens and displays a list of keys.

4. Select the key(s) in the User ID list that you want to appoint as a designated revoker.

5. Click **OK**.

   A confirmation dialog box appears.

6. Click **OK** to continue.

   The **Passphrase** dialog box appears.

7. Enter your passphrase, then click **OK**.

8. The selected key(s) is now authorized to revoke your key. For effective key management, distribute a current copy of your key to the revoker(s) or upload your key to the server. See "Distributing your public key" on page 62 for instructions.

Setting PGP preferences

PGP is configured to accommodate the needs of most users, but you have the option of adjusting some of the settings to suit your particular computing environment. You specify these settings through the **Preferences** dialog box, which you can access by choosing **Preferences** from the PGPkeys **Edit** menu.

Setting general preferences

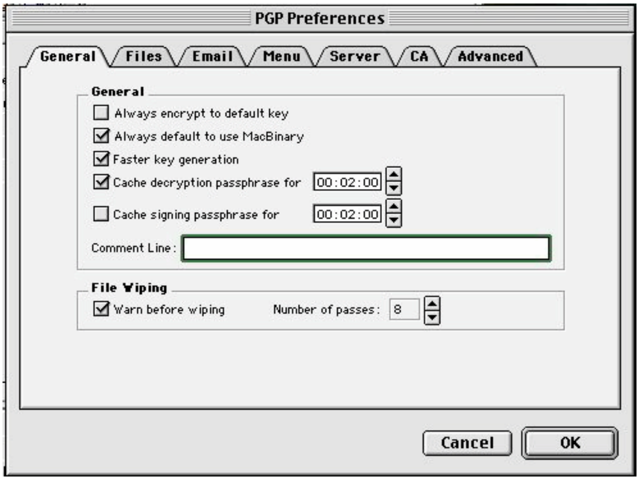
Use the General panel to specify your encrypting, signing, and file wiping preferences.

To set general PGP preferences

1. Open PGPkeys.

2. In the PGPkeys **Edit** menu, select **Preferences**.

   The **Preferences** menu opens with the **General** panel showing (Figure 6-10).



General

☐ Always encrypt to default key

☑ Always default to use MacBinary

☑ Faster key generation

☑ Cache decryption passphrase for 00:02:00

☐ Cache signing passphrase for 00:02:00

Comment Line:

File Wiping

☑ Warn before wiping   Number of passes: 8

Figure 6-10. PGP Preferences dialog box (General panel)

3. Select general encryption settings from the **General** panel. Your options are:

   - **Always Encrypt to Default Key.** When this setting is selected, all the email messages and file attachments you encrypt with a recipient's public key are also encrypted to you using your default public key. It is useful to leave this setting turned on so that you have the option of decrypting the contents of any email or files you have previously encrypted.

   - **Always Default to Use MacBinary.** When this setting is selected, Mac OS users will receive the exact file that was intended, and the Windows version will automatically decode the MacBinary and even append the appropriate file extension, such as .doc for Microsoft Word or .ppt for Microsoft PowerPoint. PGP recommends that you always select this setting. For more details, see Appendix B, "Transferring Files Between the Mac OS and Windows."

- **Faster Key Generation.** When this setting is selected, less time is required to generate a new Diffie-Hellman / DSS key pair. This process is speeded up by using a previously calculated set of prime numbers rather than going through the time-consuming process of creating them from scratch each time a new key is generated. However, remember that fast key generation is only implemented for the fixed key sizes above 1024 and below 4096 provided as options when you create a key, and is not used if you enter some other value. Although it would be unlikely for anyone to crack your key based on their knowledge of these canned prime numbers, some may want to spend the extra time to create a key pair with the maximum level of security.

  The general belief in the cryptographic community is that using canned primes provides no decrease in security for the Diffie-Hellman / DSS algorithms. If this feature makes you uncomfortable, you may turn it off.

- **Cache Decryption Passphrases for...** When this setting is selected, your decryption passphrase is automatically stored in your computer's memory. Specify the frequency (in hours: minutes: seconds) in which you want to save your passphrase. The default setting is 2 minutes.

- **Cache Signing Passphrases for...** When this setting is selected, your signing passphrase is automatically stored in your computer's memory. Specify the frequency (in hours: minutes: seconds) in which you want to save your signing passphrase. The default setting is 2 minutes.

- **Comment** Line**.** You can add your comment text in this area. The text you enter hear is always included in messages and files that you encrypt or sign. Comments entered in this field appear below the --BEGIN PGP MESSAGE BLOCK-- text header and PGP version number of each message.

- **Warn Before Wiping.** When this setting is selected, a dialog box appears before you wipe a file to give you one last chance to change your mind before PGP securely overwrites the contents of the file and deletes it from your computer.

- **Number of Passes**. This setting controls how many times the wipe utilities pass over the disk.

4. Click **OK** to save your changes and return to the PGPkeys main window or choose another tab to continue configuring your PGP preferences.

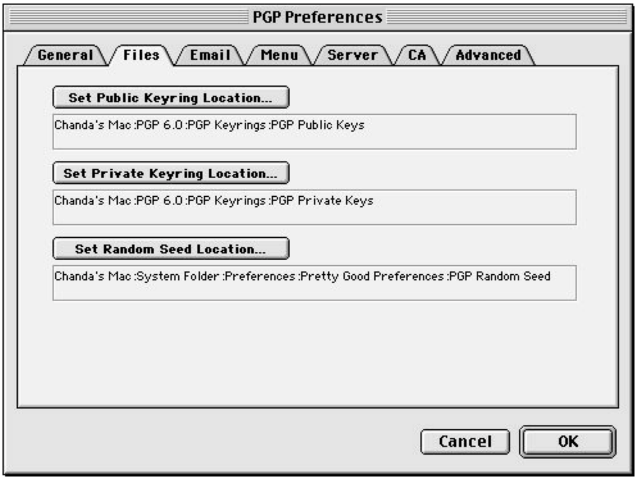
Setting file preferences

Use the **Files** panel to specify the location of the keyrings used to store your private and public keys.

To set PGP file preferences

1. Open PGPkeys.
2. Choose **Preferences** from the PGPkeys **Edit** menu, then click the **Files** tab.

   The **Preferences** menu opens with the **Files** panel showing (Figure 6-11).


PGP Preferences
General
Files
Email
Menu
Server
CA
Advanced
Set Public Keyring Location...
Set Private Keyring Location...
Set Random Seed Location...
Cancel
OK

Chanda's Mac:PGP 6.0:PGP Keyrings:PGP Public Keys

Chanda's Mac:PGP 6.0:PGP Keyrings:PGP Private Keys

Chanda's Mac:System Folder:Preferences:Pretty Good Preferences:PGP Random Seed

Figure 6-11. PGP Preferences dialog box (Files panel)

3. Use the buttons listed in the **Files** panel to set the appropriate location for your public and private keyrings, and / or random seed file:

   - **Set Public Keyring Location.** Shows the current location and name of the file where the PGP program expects to find your public keyring file. If you plan to store your public keys in a file with a different name or in some other location, you specify this information here. The location you specify will also be used to store all automatic backups of the public keyring.

- **Set Private Keyring Location.** Shows the current location and name of the file where the PGP program expects to find your private keyring file. If you plan to store your private keys in a file with a different name or in some other location, you specify this information here. Some users like to keep their private keyring on a floppy disk, which they insert like a key whenever they need to sign or decrypt mail. The location you specify will also be used to store all automatic backups of the public keyring.

- **Random Seed File.** Shows the location of the Random Seed file. Some users may wish to keep their Random Seed file in a secure location to prevent tampering. Given that this method of attack is very difficult, and has been anticipated by PGP, moving the Random Seed file from its default location is of marginal benefit.

4. Click **OK** to save your changes and return to the PGPkeys main window or choose another tab to continue configuring your PGP preferences.

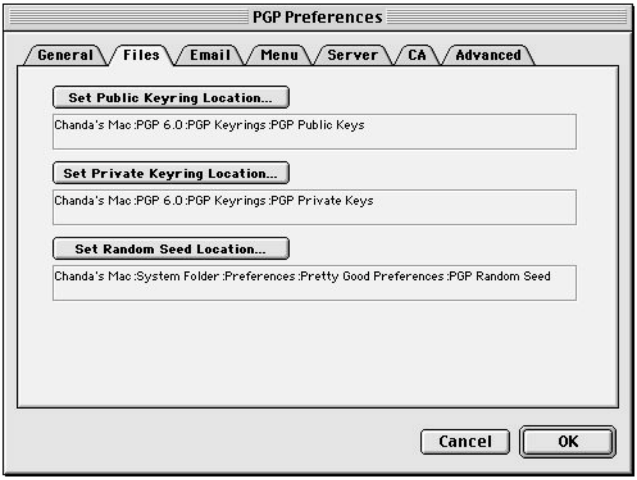
Setting email preferences

Use the **Email** panel to specify the options that affect the way PGP functions are implemented for your particular email application. Remember that not all of the selections may apply to your particular email application.

To set email preferences

1. Open PGPkeys.
2. Choose **Preferences** from the PGPkeys **Edit** menu, then click the **Email** tab.

The **Preferences** menu opens with the **Email** panel showing (Figure 6-12).


PGP Preferences
General
Files
Email
Menu
Server
CA
Advanced
Set Public Keyring Location...
Set Private Keyring Location...
Set Random Seed Location...
Cancel
OK

Chanda's Mac:PGP 6.0:PGP Keyrings:PGP Public Keys

Chanda's Mac:PGP 6.0:PGP Keyrings:PGP Private Keys

Chanda's Mac:System Folder:Preferences:Pretty Good Preferences:PGP Random Seed

Figure 6-12. PGP Preferences dialog box (Email panel)

3. Select your email encryption preferences from the **Email** panel. Your options are:

    - **Use PGP/MIME when sending mail.** If you are using Eudora and you enable this setting, all of your email messages and file attachments are automatically encrypted to the intended recipient. This setting has no effect on other encryptions you perform from PGPmenu and should not be used if you plan to send email to recipients who use email applications that are not supported by the PGP/MIME standard. Using Eudora, attachments will always be encrypted regardless of this setting, but if the recipient does not have PGP/MIME, the decryption process will be more manual.

    - **Encrypt new messages by default.** If you enable this setting, all of your email messages and file attachments are automatically encrypted. Some email applications cannot support this feature.

    - **Sign new messages by default.** If you enable this setting, all of your email messages and file attachments are automatically signed. Some email applications cannot support this feature. This setting has no effect on other signatures you add from PGPmenu.

- **Automatically decrypt/verify when opening messages.** If you enable this setting, all of your email messages and file attachments that are encrypted and / or signed are automatically decrypted and verified. Some email applications cannot support this feature.

- **Always use Secure Viewer when decrypting.** If you enable this setting, all of your decrypted email messages are displayed in the Secure Viewer window with a special TEMPEST attack prevention font. For more information about TEMPEST attacks, see “Vulnerabilities” on page 246.

- **Word wrap clear-signed messages at column [ ].** This setting specifies the column number where a hard carriage return is used to wrap the text in your digital signature to the next line. This feature is necessary because not all applications handle word wrapping in the same way, which could cause the lines in your digitally signed messages to be broken up in a way that cannot be easily read. The default setting is 70, which prevents problems with most applications.

> **WARNING:** If you change the word-wrap setting in PGP, make sure that it is less than the word-wrap settings in your email application. If you set it to be the same or a greater length, carriage returns may be added that invalidate your PGP signature.

4. Click **OK** to save your changes and return to the PGPkeys window or choose another tab to continue configuring your PGP preferences

.Setting PGPmenu preferences

From the PGPmenu panel you can perform these three tasks:

- **Add and remove PGPmenu for various applications.** To ensure easy access to PGP on your computer, you need to specify which applications on your computer that you want to integrate with PGP. For instructions, see “To add and remove PGPmenu for various applications” on page 124.

- **Enable the Wipe Trash option.** When you delete a file normally by placing it in the Trash, the name of the file is removed from the file directory, but the data in the file stays on the disk and is recoverable. When this option is enabled, the **Empty Trash** option in the Finder changes to **Wipe Trash** and wipes the Trash so that your deleted items can no longer be recovered. To enable this option, check the **Wipe Trash** checkbox in the upper right-hand side of the **Menu** panel.

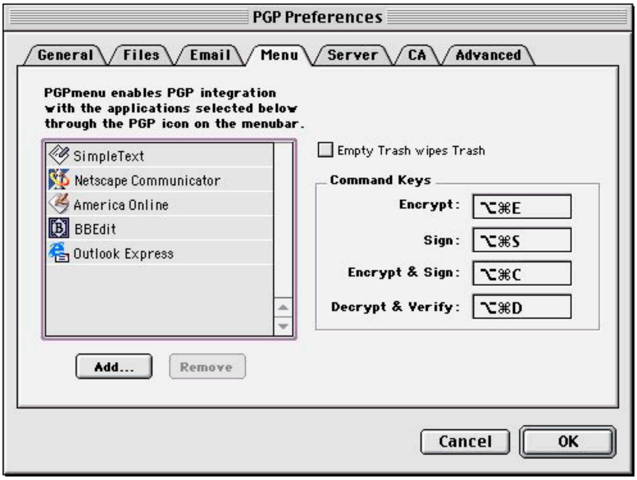
TIP: When this option is enabled, you can revert to the **Empty Trash** option in the Finder by holding down the COMMAND key.

- **View keyboard shortcuts for PGP functions.** PGPmenu also includes a set of keyboard shortcuts that gives you quick access to PGP's most common commands. The common commands include encrypting, signing, encrypting and signing, and decrypting and verifying data in the current open window.

To add and remove PGPmenu for various applications

1. Open PGPkeys.
2. Choose **Preferences** from the PGPkeys **Edit** menu, then click the **PGPmenu** tab.

   The **Preferences** menu opens with the **PGPmenu** panel showing, as shown in Figure 6-13.



PGPmenu enables PGP integration with the applications selected below through the PGP icon on the menubar.

SimpleText
Netscape Communicator
America Online
BBEdit
Outlook Express

□ Empty Trash wipes Trash

Command Keys

Encrypt:	⌥⌘E
Sign:	⌥⌘S
Encrypt & Sign:	⌥⌘C
Decrypt & Verify:	⌥⌘D

Figure 6-13. PGP Preferences dialog box (PGPmenu panel)

3. Click **Add** to add the PGP icon to the menu bar of the applications you select. For example, click the **Add** button and add Simpletext to the application list. The PGP icon is added to the **Simpletext** menu bar, so that you can sign, encrypt, decrypt, and verify the selected text in your documents.

Double-click an application name in the PGPmenu preferences to open the **Advanced PGPmenu Preferences** dialog for that particular application, which contains settings that may help if you experience any compatibility problems using PGPmenu with a particular application.

Click **Remove** to remove the PGP icon from the menu bar of applications you have previously selected.

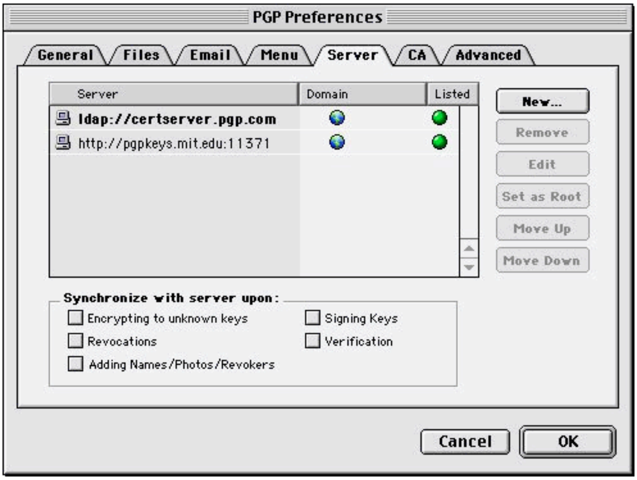
4. Click **OK** to save your changes and return to the PGPkeys menu or choose another tab to continue configuring your PGP preferences.

Setting server preferences

Use the **Server** panel to specify settings for the public key servers that you are using to send and retrieve public keys, and with which you will automatically synchronize keys.

To set key server preferences

1. Open PGPkeys.
2. Choose **Preferences** from the PGPkeys **Edit** menu, then click the **Server** tab.
3. The **Preferences** menu opens with the **Server** panel showing (Figure 6-14).



Server	Domain	Listed
ldap://certserver.pgp.com	🌐	●
http://pgpkeys.mit.edu:11371	🌐	●

Synchronize with server upon:

□ Encrypting to unknown keys
□ Revocations
□ Adding Names/Photos/Revokers
□ Signing Keys
□ Verification

Figure 6-14. PGP Preferences dialog box (Server panel)

The **Domain** column lists the Internet domain (such as “company.com”) of the available key server(s). When sending keys to a server, PGP attempts to find the key’s domain in this list, and thus find the appropriate server entry. If the domain is not found, a server for the first world domain server which serves all keys will be used, and other world domain servers down the list may be searched if the first search is unsuccessful.

4. To set your server options, use these buttons:

   - **New.** Adds a new server to your list.
   - **Remove.** Removes the currently selected server from your list.
   - **Edit.** Allows you to edit server information for the currently selected server.
   - **Set as root.** Identifies the root server that is used for specific corporate operations, such as updating group lists, sending group lists, updating introducers, etc. In corporate settings, your security officer will have already configured this.
   - **Move Up** and **Move Down**. Use these buttons to arrange the servers in order of preference.

5. In the **Synchronize with server upon** area, select the options to use when synchronizing your private keyring with your key server(s). Your options are:

   - **Encrypting to unknown keys.** Select this option to have PGP automatically look up unknown recipients on the server to locate users that are not on your keyring when encrypting email.
   - **Signing keys.** Select this option to allow keys to which you’re adding your signature first to be updated from the server and then your changes sent to the server upon completion of the update.
   - **Adding names/photos/revokers.** Select this option to allow keys to which you’ve added names, photographs, or revokers first to be updated from the server and then your changes sent to the server upon completion of the update. Updating the key beforehand ensures that, for example, the key has not been revoked since you last updated it.
   - **Revocations.** Select this option to allow keys you revoke first to be updated from the server and then your changes sent to the server upon completion of the update.

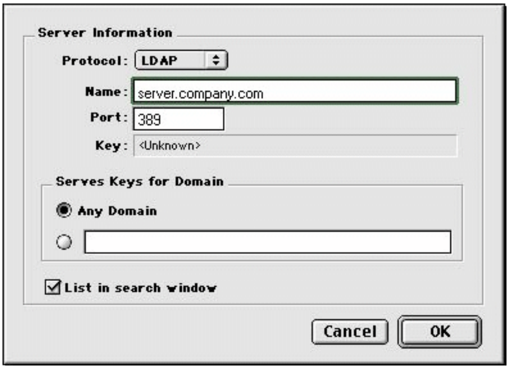
- **Verification.** Select this option to have PGP automatically search and import from the key server when verifying a signed email message or file for which you do not have the sender's public key.

6. Click **OK** to save your changes and return to the PGPkeys main window or choose another tab to continue configuring your PGP **Preferences**.

To add a key server to the server list

1. Open PGP **Preferences**, then click the **Servers** tab.
2. Click the **New** button.

   The **Add New Server** dialog box appears, as shown in Figure 6-15.



Protocol: LDAP
Name: server.company.com
Port: 389
Key: <Unknown>
● Any Domain
○
☑ List in search window

Figure 6-15. Add New Server dialog box

3. In the **Protocol** box, select a protocol to use to access the server. Your options are **LDAP**, **LDAPS**, and **HTTP**.
4. In the **Server Name** box, enter the domain name or IP address of the server. For example, server.company.com or 123.445.67.
5. Type the port number of the server in the **Port** box. For example 11371 is used for old-style HTTP certificate server, 389 is commonly used for LDAP certificate servers.
6. The **Server Key** box is for LDAPS servers. The server key is used by the server to authenticate the connection. (Key information is not displayed until you connect to the server.)
7. Select the **Any Domain** option to allow PGP to send keys from any domain to this key server. This option is enabled by default.

If you want PGP to send only keys from a specific domain to this key server, select the option below **Any Domain**. Then, enter the domain name in the space provided. For example, if you specify the domain company.com, only those keys whose email address ends in company.com will be sent to this server.

8. Select the **List in Search Window** checkbox if you want this key server listed in the **PGPkeys Search** window.

Accessing HTTP servers through your corporate proxy server

If your Macintosh is behind a corporate firewall with an HTTP proxy server, you can access HTTP key servers through the proxy by configuring the proxy server address in the Internet control panel.

NOTE: This feature requires the installation of Internet Config for users not running Mac OS 8.5 or greater.

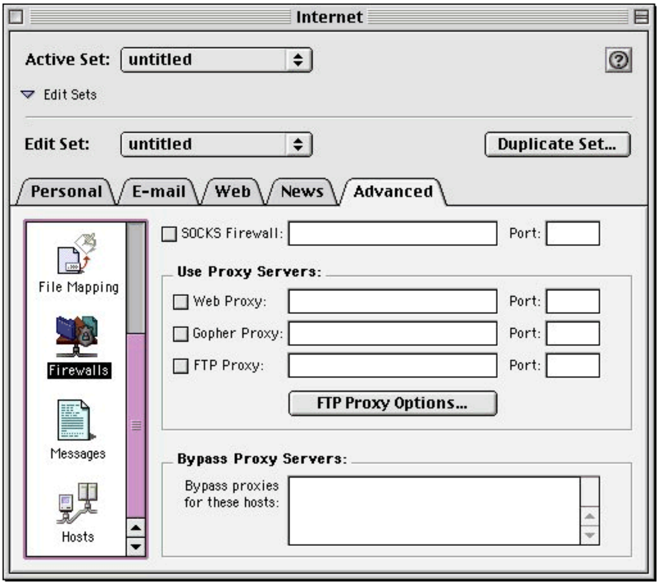
To access HTTP key servers through a proxy server

1. Open the Internet control panel.
2. Choose **User Mode** from the Internet control panel **Edit** menu.

   The **User Mode** dialog box appears.
3. Select **Advanced** or **Administration**, and then click **OK**.
4. On the Internet control panel, click the **Advanced** tab.
5. Click the **Firewalls** icon on the left side of the **Advanced** panel.

   The Firewall options are displayed in the Advanced panel, as shown in Figure 6-16.

.



Active Set: untitled

Edit Set: untitled

□ SOCKS Firewall: Port:

□ Web Proxy: Port:

□ Gopher Proxy: Port:

□ FTP Proxy: Port:

Figure 6-16. Internet control panel (Advanced Firewall options)

6. Select the **Web Proxy** checkbox.

7. Type the name of the proxy server in the **Web Proxy** text box, then enter the port number for the proxy server in the **Port** text box.

   If you do not know the name of the proxy server or the port number, contact your System Administrator.

8. Choose **Save Settings** from the **File** menu.

9. Reboot your system for the changes to take effect.

Setting CA preferences

Use the **CA** panel to add your X.509 certificate to your PGP key. Before you can add your X.509 certificate however, you must first obtain the Root CA certificate from your company's certificate server. For instructions on obtaining the Root CA Server certificate, see "Obtain and add the Root CA certificate to your PGP keyring." on page 53. For detailed instructions about setting CA preferences and adding your X.509 certificate to your key, see "Adding an X.509 certificate to your PGP key" on page 53.

Setting advanced preferences

Use the **Advanced** panel to select key encryption algorithms and key trust options.

PGP gives you the option to select and / or change key encryption algorithms. You can select the encryption algorithm for your PGP keys: CAST (the default), IDEA, or Triple-DES. If you want to use IDEA or Triple-DES you must make the selection before you generate your keys. CAST is a new algorithm in which PGP and other cryptographers have very high confidence, and Triple-DES is a U.S. Government algorithm that has withstood the test of time. IDEA is the algorithm used for all RSA keys generated by PGP. For more information about these algorithms, see "The PGP symmetric algorithms" on page 231.

The **Preferred Algorithm** choice affects the following:

- When using conventional encryption, the preferred cipher is used to encrypt.
- When creating a key, the preferred cipher is recorded as part of the key so that other people will use that algorithm when encrypting to you.

The **Allowed Algorithm** choice affects the following:

- When creating a key, the allowed ciphers are recorded as part of the key so that other people will use one of those algorithms when encrypting to you if the preferred algorithm is not available to them.

NOTE: Encrypting to a public key will fail if neither the Preferred Algorithm nor any of the Allowed Algorithms are available to the person encrypting the message.

WARNING: Use the CAST, IDEA, and Triple-DES checkboxes only if you have suddenly learned that a particular algorithm is insecure. For example, if you become aware that Triple-DES has been broken, you can deselect that checkbox and all new keys you generate will have a record that Triple-DES may not be used when encrypting to you.

PGP gives you the option to select and / or change how key trust is displayed, and whether or not you wish to be warned whenever you encrypt a message to a public key that has an associated Additional Decryption Key. In the Trust Model section, choose from these options:

- **Display marginal validity level.** Use this checkbox to specify whether to display marginally valid keys as such, or simply to show validity as on or off. Marginal validity appears as bar icons having differing shading patterns. On / off validity appears as circle icons; green for valid, gray for invalid (the key has not been validated; it has not been signed by either a trusted introducer or by you).

- **Treat marginally valid keys as invalid.** Use this checkbox to specify whether to treat all marginally valid keys as invalid. Selecting this option causes the **Key Selection** dialog box to appear whenever you encrypt to marginally valid keys.

- **Warn when encrypting to an ADK.** Use this checkbox to specify whether to issue a warning whenever an encrypt-to key has an associated Additional Decryption Key.

- **Export format.**

  - **Compatible:** Exports keys in a format compatible with previous versions of PGP.

  - **Complete:** Exports the new key format, which includes photographic IDs and X.509 certificates.

PGPdisk 7

This chapter describes PGPdisk, its features, and provides instructions on how to use it.

What is PGPdisk?

PGPdisk is an easy-to-use encryption application that enables you to set aside an area of disk space for storing your sensitive data. This reserved space is used to create a file called a PGPdisk *volume.*

Although it is a single file, a PGPdisk volume acts very much like a hard disk in that it provides storage space for your files and applications. You can think of it like a floppy disk or an external hard disk. To use the applications and files stored in the volume, you *mount* it, or make it accessible to you.

When a PGPdisk volume is *mounted*, you can use it as you would any other disk. You can install applications within the volume or move or save your files to the volume. When the volume is *unmounted* it is inaccessible to anyone who does not know your secret *passphrase*, which is a longer version of a password. Even a mounted volume is protected: unless a file or application is in use, it is stored in encrypted format. If your computer should crash while a volume is mounted, the volume's contents remain encrypted.

NOTE: PGP products encourage you to use an entire phrase or long sequence of characters to protect sensitive data. Such passphrases are generally more secure than traditional 6-10 character passwords.

PGPdisk features

The PGPdisk program:

- Allows you to create secure volumes of encrypted data which function just like any other volumes you are accustomed to using for storing your files.
- Provides fast and secure encryption of your data with minimal impact on the amount of time it takes to access your programs and files.
- Uses a strong, military-grade encryption algorithm known as CAST, which has a solid reputation for its ability to withstand unauthorized access.
- Stores the contents of each secure volume in an encrypted file which can easily be backed up and exchanged with colleagues.

Why use PGPdisk?

While other products offer the ability to restrict access to disk files through permission attributes and simple password protection, these safeguards can easily be breached by those truly intent on examining your data. Only by encrypting your data can you rest assured that even with the most sophisticated technologies known today, it is nearly impossible for anyone to decipher the content of your files.

Here are a few reasons to use PGPdisk to secure the contents of your files:

- To protect sensitive financial, medical and personal information that you simply do not want others to have access to. This is particularly important in today's networked environment where information on your personal computer is exposed to the world while you are surfing the net.
- To set up personal work areas on a shared machine where each user is guaranteed exclusive access to his or her own programs and files. Each user can mount his or her own volumes while using the machine and then rest assured that nobody else can access the files once the volumes are unmounted.
- To create volumes of material that are accessible only to designated members of a given workgroup. A volume can be mounted when members of the team want to work on a given project and can then be unmounted and stored in its encrypted format when they are finished.
- To prevent someone from gaining access to proprietary information stored on a notebook computer. In general, if you lose your notebook (or someone steals it), all of your personal information (including access and passwords to on-line services, business and personal contacts, financial records, and so on), are subject to misuse by those with criminal intent and could end up costing more than the price of the missing notebook.
- To secure the contents of external media such as floppy disks and storage cartridges. The ability to encrypt external media provides an added level of security for storing and exchanging sensitive information.

Starting the PGPdisk program

To start PGPdisk

1. Double-click in the PGP program folder

   This opens the PGPdisk toolbar as shown in Figure 7-1.

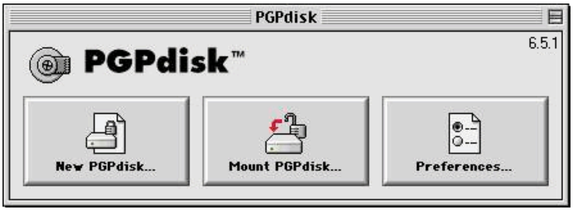


Figure 7-1. The PGPdisk toolbar

The PGPdisk toolbar provides a convenient means of creating and mounting volumes. Here is a brief description of each button:

New PGPdisk Displays the PGPdisk wizard, which guides you through the process of creating a new PGPdisk volume.

Mount PGPdisk Mounts the specified PGPdisk volume provided that the correct passphrase is entered.

Preferences Specifies how you prefer to unmount your volumes.

NOTE: You can also use the PGPdisk **File** menu to access the PGPdisk functions.

Working with PGPdisk Volumes

This section explains how to create, mount and unmount PGPdisk volumes and how to specify preferences which protect volume contents by unmounting them under certain circumstances.

Creating a new PGPdisk volume

To create a new PGPdisk volume

1. Start PGPdisk. The PGPdisk toolbar appears.
2. Click **New**. The PGPdisk Wizard appears on your screen. Read the introductory information.
3. Click **Next**.
4. Specify the name and location of the new volume.
5. Click **Save**.
6. Enter the amount of space you want to reserve for the new volume (PGPdisk Size field). Use whole numbers, no decimal places. You can use the arrows to increase or decrease the number displayed in the field.

   The amount of free disk space for the selected drive is shown above the Size field.
7. Click the appropriate radio button to select kilobytes, megabytes, or gigabytes.

   Depending on the amount of available disk space, you can create a volume of any size between 100 kilobytes and 2 gigabytes.
8. Select the letter of the drive where you want to mount your PGPdisk volume (PGPdisk Drive Letter field). You can use the arrow to display and select a different drive letter.
9. Click **Next**.
10. Enter the string of words or characters that will serve as your passphrase to access the new volume (also called the volume’s master passphrase). To confirm your entry, press TAB to advance to the next text box, then enter the same passphrase again. The minimum size for a passphrase is 8 characters.

Normally, as an added level of security, the characters you enter for the passphrase are not visible on the screen. However, if you are sure that no one is watching (either physically or over the network) and you would like to see the characters of your passphrase as you type, click in the **Hide Typing** check box.

NOTE: Your security is only as good as your passphrase. Your passphrase should contain more than one word, along with spaces, numbers, and other printable characters. The passphrase is case sensitive. The minimum allowable passphrase is 8 characters. Choose something with which you are very familiar and that you have already stored in your long-term memory. Picking a phrase on the spur of the moment is likely to result in forgetting it entirely. It is vital that you *do not forget your passphrase or you will lose your data*! For more information, see "Passphrase quality" on page 149.

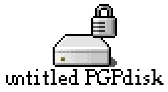
11. Click **Next**.
12. Move your mouse around in a random manner within the Wizard window and / or type characters on the keyboard until the progress bar shown in the dialog box is completely filled in.

    Your mouse movements and typing are used to generate random information used by the PGPdisk program as part of the encryption (data scrambling) process.

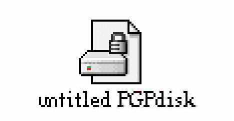
13. Click **Next**. A progress bar indicates how much of the PGPdisk volume has been initialized.

    Click **Done** to begin working with your new PGPdisk. A PGPdisk mounted volume icon representing your volume appears on your desktop, as shown below.

    A PGPdisk encrypted volume icon representing your secure volume appears in the location you specified, as shown below.



Mounted PGPdisk volume



Encrypted PGPdisk volume

14. Double-click the icon to open the volume.

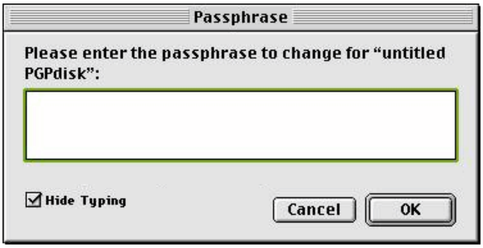
Changing a passphrase

You can change the master or alternate passphrase for a PGPdisk File.

To change your passphrase

1. Make sure that the PGPdisk volume is not mounted. You cannot change a passphrase if the PGPdisk volume is mounted.
2. Choose **Change Passphrase** from the **File** menu.

   The **Open** dialog box appears.
3. Navigate to the disk file of interest.
4. The **Passphrase** dialog box appears, as shown in Figure 7-2.



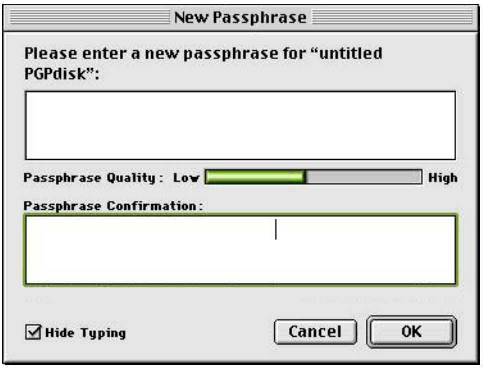
Please enter the passphrase to change for “untitled PGPdisk”:

☑ Hide Typing

Figure 7-2. The Change Passphrase dialog box

5. Enter your passphrase, then click **OK**.

   The **New Passphrase** window appears, as shown in Figure 7-3.



Please enter a new passphrase for “untitled PGPdisk”:

☑ Hide Typing

Figure 7-3. New Passphrase dialog box

6. Enter the string of words or characters that will serve as your new passphrase to access the new volume (also called the volume's master passphrase). To confirm your entry, press TAB to advance to the next text box, then enter the same passphrase again. The minimum size for a passphrase is 8 characters.

7. Click **OK**.

   The **New Passphrase** dialog box closes.

Adding alternate passphrases

Once you have entered the master passphrase (the one used to initially create the disk), you can add up to seven other alternate passphrases which can be used to mount the volume. You might want to do this if you use the same master passphrase on a regular basis and you want to make the volume available to someone else with their own unique passphrase. Only a person who knows the master passphrase can add alternate passphrases.

Any user who knows a passphrase can change that passphrase, but you will always be able to access the contents of the volume if it becomes necessary. You also have the option of assigning a "read-only" status to the volume which allows the individual to read the files but prevents them from altering the files in any way.

To add alternate passphrases

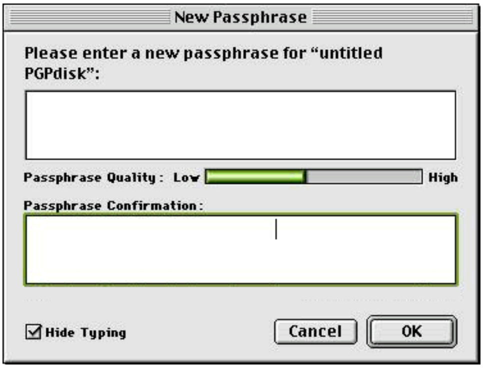
1. Ensure that the PGPdisk volume is not currently mounted. You cannot add or change a passphrase while the PGPdisk volume is mounted.

2. Select **Add Passphrase** from the **File** menu.

   The **Passphrase** dialog box appears, asking you to enter the volume's master passphrase. If you have multiple PGPdisk volumes on your machine, you must select a disk volume.

3. Enter the master passphrase and click **OK**.

   The **New Passphrase** dialog box appears, as shown in Figure 7-4.



Please enter a new passphrase for “untitled PGPdisk”:

Figure 7-4. The New Passphrase dialog box

4. Enter an alternate passphrase for the named volume and then press TAB. Enter the passphrase again to confirm it.

   At this point, you also have the option of checking the **Read-only Passphrase** checkbox to indicate that you want the entire volume contents to be designated as “read-only.”

5. Click **OK**.

Once you have created an alternate passphrase, you (or anyone who knows it) can remove the passphrase by choosing the **Remove Passphrase** command from the **File** menu. Master passphrases cannot be removed. (For more information, see “Removing a passphrase”, below.

Removing a passphrase

Removing a passphrase is similar to adding or changing a passphrase. You cannot remove a master passphrase.

To remove a passphrase

1. Make sure that the PGPdisk volume is not mounted. You cannot remove a passphrase if the PGPdisk volume is mounted.

2. Choose **Remove Passphrase** from the **File** menu.

   A dialog box appears, prompting you to enter the passphrase to be removed.

3. Enter the passphrase and then click **OK**.

Removing all alternate passphrases

You can also remove all alternate passphrases at once. This could be useful if other users have alternate passphrases to a PGPdisk volume, and you no longer want them to have access to the volume.

To remove all alternate passphrases

1. Make sure that the PGPdisk volume is not mounted. You cannot remove a passphrase if the PGPdisk volume is mounted.
2. Hold down the Option key, open the File menu, and select Remove Alternate Passphrases.
3. Follow the rest of the procedure as described in "Removing a passphrase" above.

Add/Remove Public Keys

You can add and remove public keys for a PGPdisk file. This feature allows you and others who know the passphrases for those keys to use the keys to mount the volume.

To add a public key to your PGPdisk volume

1. Make sure that the PGPdisk volume is not mounted. You cannot add a public key if the volume is mounted.
2. Choose **Add/Remove Public Keys** from the **File** menu.
3. Select the PGPdisk from the **Open** dialog box.

   You are prompted to enter the master passphrase.

   The **Recipient Selection** window appears.
4. Drag the key or keys from the top pane in the window to the bottom pane.
5. Click **OK**.

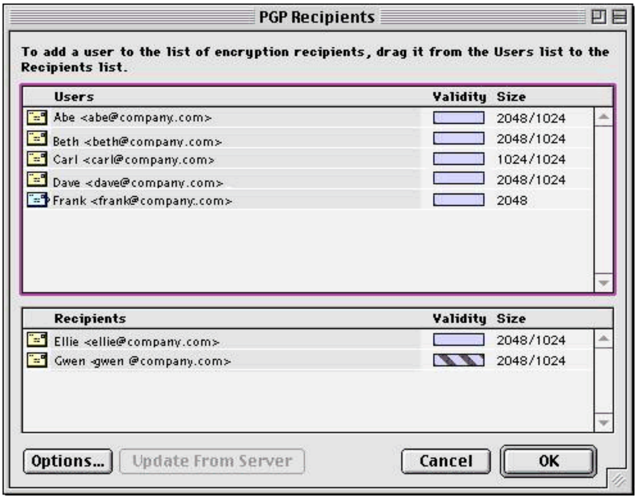
To remove a public key from your PGPdisk volume

1. Make sure that the PGPdisk volume is not mounted. You cannot remove a public key if the volume is mounted.

2. Choose **Add/Remove Public Keys** from the **File** menu.
3. Select the PGPdisk from the Open dialog box.

   You are prompted to enter the master passphrase.

   The **PGP Key Selection** window appears, as shown in Figure 7-5.



PGP Recipients

To add a user to the list of encryption recipients, drag it from the Users list to the Recipients list.

Users	Validity	Size
Abe <abe@company.com>	▭	2048/1024
Beth <beth@company.com>	▭	2048/1024
Carl <carl@company.com>	▭	1024/1024
Dave <dave@company.com>	▭	2048/1024
Frank <frank@company.com>	▭	2048

Recipients	Validity	Size
Ellie <ellie@company.com>	▭	2048/1024
Gwen <gwen @company.com>	▨	2048/1024

Figure 7-5. PGP Key Selection Dialog

4. Drag the key or keys from the bottom pane in the window to the top pane.
5. Click **OK**.

Mounting a PGPdisk volume

When you create a new volume, the PGPdisk program automatically mounts it so you can begin using it to store your files. When you are ready to secure the contents of the volume, you must unmount it (for details, see “Unmounting a PGPdisk volume” on page 144). Once a volume is unmounted, its contents remain secured in an encrypted file where it is inaccessible until the volume is once again mounted.

There are several ways to mount a volume.

- Drag the volume’s icon onto the PGPdisk icon in the PGP 6.5 folder.
- Drag the volume’s icon onto the **Mount** button on the PGPdisk toolbar.

- Use the **Mount** button on the PGPdisk toolbar.

To mount a volume using the Mount button

1. Start PGPdisk.

   The **PGPdisk toolbar** appears.

2. Click **Mount** or use the **Mount PGPdisk** option from the **File** menu.

   The **Mount PGPdisk** dialog box appears.

3. Locate and select the encrypted volume you want to mount, then click **Open**.

   You are prompted to enter the passphrase for the selected volume.

4. Enter the passphrase and click **OK**. If you do not want to modify the files in the volume, click the **Read-only** check box. If you entered the correct passphrase, the volume is mounted and the data in the encrypted file is made accessible.

   Alternatively, you can also mount a volume without running the PGPdisk program. Instead, you can simply double-click on the name of the encrypted file (or its icon) from the Finder, or you can drag the file onto the PGPdisk program icon.

Using a mounted PGPdisk volume

You can create, copy, move, and delete files and folders on a PGPdisk volume just as you normally do with any other volume. Similarly, anyone else who has access to the volume (either on the same machine or perhaps over the network) can also access the data stored in the volume. It is not until you unmount the volume that the data in the encrypted file associated with the volume is made inaccessible.

WARNING: Although the encrypted file associated with each volume is safe from snooping, it can still be deleted. If an unauthorized person is able to access your data, he or she could potentially delete the encrypted file upon which the volume is based. It is a good idea to keep a backup copy of the encrypted file.

Unmounting a PGPdisk volume

After you are through accessing a given volume and you want to lock its contents, you need to unmount the volume. You cannot unmount a volume that has any open files.

To unmount a PGPdisk volume

1. Close all files in the PGPdisk volume that you want to unmount.
2. Select the PGPdisk volume icon, then choose **Put Away** from the **File** menu.

   As an alternative, you can also drag the icon representing the volume to the trash.

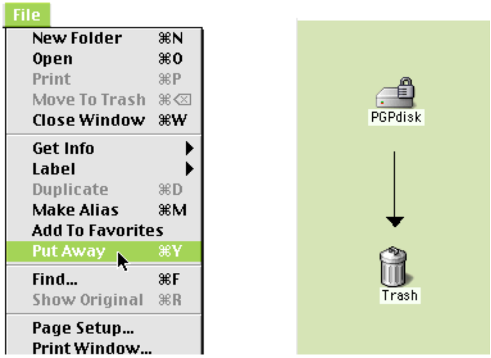


Figure 7-6. Unmounting a PGPdisk volume

Once a volume is unmounted, its contents are locked in the encrypted file associated with the volume. The contents of the volume are stored in the encrypted file and its contents remain inaccessible until the volume is once again mounted. It may help to view PGPdisk volumes as simply a window which provides a view to the data in the encrypted file. The contents of a PGPdisk volume file only become available when it is mounted as a volume by someone who knows a valid passphrase.

Specifying Preferences

The **Preferences** button on the PGPdisk toolbar allows you to specify how you prefer to unmount and create your volumes.

To specify Preferences

1. Click **Preferences** on the PGPdisk toolbar or select **Preferences** from the **File** menu.

   The **Preferences** dialog box appears.

2. Select the desired options by clicking the appropriate checkboxes.

   - **Auto unmount after [15] minutes of inactivity.** When checked, this option causes PGPdisk to automatically unmount any mounted PGPdisk volumes when your computer is inactive for the number of minutes in the box. You can set this value from 1 to 999 minutes.

     ❐ **NOTE:** PGPdisk cannot automatically unmount a PGPdisk volume if any of the files in that volume are open.

   - **Auto unmount on computer sleep.** When checked, this option causes PGPdisk to automatically unmount any mounted PGPdisk volumes when your computer goes into Sleep mode. (Not all computer models have a sleep mode.)

3. Click **OK** when you are through specifying your preferences.

The automatic unmount settings are useful if you need to leave your computer unattended for a period of time. You should adjust the timing for these settings according to how secure your system is from unauthorized physical access. You can set both of these preferences at the same time.

Maintaining PGPdisk Volumes

This section describes how to automatically mount PGPdisk volumes when you start your system, and how to back-up and exchange the data in these volumes with others.

Automatically mounting PGPdisk volumes

If you like, you can automatically mount PGPdisk volumes when you first start your system.

To automatically mount PGPdisk volumes

1. Create an alias for each of the PGPdisk files which you want mounted when you start your computer.
2. Place the alias (or aliases) in the **Startup Items** folder which is located in the System Folder.

   Once you have placed the aliases in your Startup Items folder, the PGPdisk volumes are mounted whenever you start your computer. You are prompted to enter the passphrase for each PGPdisk volume that is being mounted.

Backing up PGPdisk volumes

You may want to back up the contents of your PGPdisk volumes to safeguard your information from system corruption or disk failures. While it is possible to back up the contents of a mounted PGPdisk volume just as you would any other volume, it is probably not a good idea because the contents are not encrypted and will thus be accessible to anyone who can restore the back up. Rather than back up the contents of the mounted PGPdisk volume, you should instead make a back up of the encrypted PGPdisk volume.

To back up PGPdisk volumes

1. Click on the PGPdisk volume icon. Open the **File** menu and select the **Put Away** option. You can also simply drag the icon representing the volume to the Trash.
2. Make a back up of the encrypted file to a floppy disk or removable cartridge just as you would any other file. Even if some unauthorized person has access to the back up, they will not be able to decipher its contents.

When making backups of the encrypted files, keep these issues in mind:

- PGPdisk is a product for security-minded people and organizations. Backing up the encrypted files to a network drive gives others plenty of opportunity to guess at a weak passphrase. We recommend that you back up only to devices over which you have physical control. A lengthy, complicated passphrase helps further reduce the risk in this situation. See “Passphrase quality” on page 23.

- If you are on a network, make sure that any network back up system does not back up your mounted volumes. You may need to discuss this with your System Administrator. Under some circumstances, you may not mind if backups are made of your encrypted files because this information is secure. Under no circumstances is it a good idea to allow the contents of your mounted volumes to be backed up, as this defeats the whole purpose of keeping this information encrypted.

Exchanging PGPdisk volumes

You can exchange PGPdisk volumes with colleagues who have their own PGPdisk program by sending them a copy of the encrypted file which contains the data associated with the volume. Here are some of the ways you might exchange PGPdisk volumes:

- As mail attachments
- On floppy disks or cartridges
- Over a network

TIP: You should carefully consider the method you use to provide someone the passphrase used to gain access to a PGPdisk volume. In general, unless you use PGP to encrypt your message, email is not a good way to exchange passphrases. Telephone lines are also vulnerable to monitoring and your conversation could be overheard. The more security precautions you take, the greater assurance you have that your sensitive information remains confidential. If you do not have secure e-mail, then it is probably safer to tell the other person the passphrase in a face-to-face meeting or even by regular postal mail.

Once the intended party has a copy of the encrypted file, all they need in order to gain access to the contents of the volume is to mount it using the correct passphrase, or, if the volume was encrypted to their public key, their private key. They also need a copy of the PGPdisk program. For more information on how to mount a PGPdisk volume, see: “Mounting a PGPdisk volume” on page 142.

Changing the size of a PGPdisk volume

While you cannot change the size of a PGPdisk volume once it has been created, you can create a larger or smaller volume and then copy the contents from the old volume to the new one.

To change the size of a PGPdisk volume

1. Create a new PGPdisk volume and specify the desired size.
2. Copy the contents of the existing mounted PGPdisk volume into the newly created volume.
3. Unmount the old PGPdisk volume and then delete the encrypted file associated with the volume to free up the disk space.

Technical Details and Security Considerations

This section discusses encryption and security issues and provides user tips and other technical information about PGPdisk.

About PGPdisk volumes

You can use PGPdisk volumes to organize your work, keep similarly named files separate, or keep multiple versions of the same documents or programs separate.

Although the volumes you create with PGPdisk function just as any other volume you are accustomed to working with, the data is actually stored in one large encrypted file. It is only when you mount the file that its contents are presented in the form of a volume. It is important to realize that all of your data remains secure in the encrypted file and is only deciphered when you access one of the files. Having the data for a volume stored in this manner makes it easy to manipulate and exchange PGPdisk volumes with others but it also makes it easier to lose data if the file is somehow deleted. It is wise to keep a back up copy of these encrypted files so that the data can be recovered in case something happens to the original. It is also important to note that you cannot compress an encrypted file in an attempt to reduce its size, but you can compress the individual files contained in the mounted volume and thereby store more encrypted data in the volume. You can also store one secure PGPdisk volume within another and thus nest several volumes for an added level of security.

The PGPdisk encryption algorithm

Encryption employs a mathematical formula to scramble your data so that no one else can use it. When you apply the correct mathematical key, you unscramble your data. The PGPdisk encryption formula uses random data for part of the encryption process. Some of this random data comes from the movement of your mouse during encryption and some random data also comes directly from your passphrase.

The PGPdisk program uses a sophisticated encryption algorithm referred to as CAST, which is considered an excellent block cipher because it is fast and very difficult to break. Its name is derived from the initials of its designers, Carlisle Adams and Stafford Tavares of Northern Telecom (Nortel). Nortel has applied for a patent for CAST, but they have made a commitment to make CAST available to anyone on a royalty-free basis. CAST appears to be exceptionally well-designed by people with good reputations in the field. The design is based on a very formal approach, with a number of formally provable assertions that give good reasons to believe that it probably requires key exhaustion to break its 128-bit key. CAST has no weak keys. There are strong arguments that CAST is immune to both linear and differential cryptanalysis, the two most powerful forms of cryptanalysis in the published literature, both of which have been effective in cracking the Data Encryption Standard (DES).

Passphrase quality

Your security is only as good as your passphrase. However, encrypting a file and then finding yourself unable to decrypt it is a painful lesson in learning how to choose a passphrase you will remember.

Most applications require a password between three and eight letters. A single word password is vulnerable to a “dictionary attack,” which consists of having a computer try all the words in the dictionary until it finds your password. To protect against this manner of attack, it is widely recommended that you create a word that includes a combination of upper and lowercase alphabetic letters, numbers, punctuation marks, and spaces. This results in a stronger password, but an obscure one that you are unlikely to remember easily. We do not recommend that you use a single-word passphrase.

A passphrase is less vulnerable to a dictionary attack. This is accomplished easily by using multiple words in your passphrase, rather than trying to thwart a dictionary attack by arbitrarily inserting a lot of funny non-alphabetic characters, which has the effect of making your passphrase too easy to forget and could lead to a disastrous loss of information because you can’t decrypt your own files. However, unless the passphrase you choose is something that is easily committed to long-term memory, you are unlikely to remember it verbatim. Picking a phrase on the spur of the moment is likely to result in forgetting it entirely. Choose something that is already residing in your long-term memory. It should not be something that you have repeated to others recently, nor a famous quotation, because you want it to be hard for a sophisticated attacker to guess. If it’s already deeply embedded in your long-term memory, you probably won’t forget it. *Do not write it down!*

Your passphrase is part of the random data used to encrypt your PGPdisk files. The Passphrase Quality bar should fill at least half way when you enter your passphrase. Unless you fill the entire bar, you are not achieving maximum security.

You can create a separate or alternate passphrase for every PGPdisk volume you create. This enables you to allow some users access to selected PGPdisk files on a volume-by-volume basis. You can use a passphrase for PGPdisk files that you send to a colleague, and still prevent that colleague from accessing any of your other PGPdisk files.

Special security precautions taken by PGPdisk

PGPdisk takes special care to avoid security problems that other programs may not. These include the following:

Passphrase erasure

When you enter a passphrase, PGPdisk uses it only for a brief time, then erases it from memory. PGPdisk also avoids making copies of the passphrase. The result is that your passphrase typically remains in memory for only a fraction of a second. This feature is crucially important — if the passphrase remained in memory, someone could search for it in your computer memory while you were away from the machine. You would not know it, but they would then have full access to any PGPdisk volumes protected by this passphrase.

Virtual memory protection

Your passphrase or other keys could be written to disk as part of the virtual memory system swapping memory to disk. PGPdisk takes care that the passphrases and keys are never written to disk. This feature is important because someone could scan the virtual memory file looking for passphrases.

Memory Static Ion Migration Protection

When you mount a PGPdisk, your passphrase is turned into a key. This key is used to encrypt and decrypt the data on your PGPdisk volume. While the passphrase is erased from memory immediately, the key (from which your passphrase cannot be derived) remains in memory while the disk is mounted. This key is protected from virtual memory; however, if a certain section of memory stores the exact same data for extremely long periods of time without being turned off or reset, that memory tends to retain a static charge, which could be read by attackers. If your PGPdisk is mounted for long periods, over time, detectable traces of your key could be retained in memory. You won’t find such devices at your neighborhood electronics shop, but major governments are likely to have a few.

PGPdisk protects against this by keeping two copies of the key in RAM, one normal copy and one bit-inverted copy, and inverting both copies every few seconds.

Other security considerations

In general, the ability to protect your data depends on the precautions you take, and no encryption program can protect you from sloppy security practices. For instance, if you leave your computer on with sensitive files open when you leave your desk, anyone can access that information or even obtain the key used to access the data. Here are some tips for maintaining optimal security:

- Make sure that you unmount PGPdisk volumes when you leave your computer. This way, the contents will be safely stored in the encrypted file associated with the volume until you are ready to access it again.
- Use a screen saver with a password option so that it is more difficult for someone to access your machine or see your screen when you are away from your desk.
- Make sure that your PGPdisk volumes cannot be seen by other computers on the network. You may need to talk to your network management people to guarantee this. The files in a mounted PGPdisk volume can be accessed by anyone who can see them on the network.
- Never write down your passphrases. Pick something you can remember. If you have trouble remembering your passphrase, use something to jog your memory, such as a poster, a song, a poem, a joke, but *do not write down your passphrases*.
- If you use PGPdisk at home and share your computer with other people they will probably be able to see your PGPdisk files. As long as you unmount the PGPdisk volumes when you finish using them, no one else will be able to read their contents.
- If another user has physical access to your machine that person can delete your PGPdisk files as well as any other files or volumes. If physical access is an issue, try either backing up your PGPdisk files or keeping them on an external device over which only you have physical control.
- Be aware that copies of your PGPdisk volume use the same secret key as the original. If you exchange a copy of your volume with another and both change your master passwords, both of you are still using the same key to encrypt the data. While it is not a trivial operation to recover the key, it is not impossible.

PGPnet Virtual Private Networking 8

This chapter describes PGPnet, its features, and provides instructions on how to use it. This chapter also introduces you to the concept of Virtual Private Networks.

The technology of today has brought many changes to the workplace. The bulk of interoffice memos and reports traditionally placed in a mailbox and received in a few days is now sent electronically and received in a matter of seconds. Employees who work at home or travel can now make a phone call to transfer data to and from their local or home office.

Two by-products of these advances are an increased security threat to data transmitted over phone lines, and a significant rise in the cost of phone services. Companies saw the Internet as an answer to rising costs, but security remained an issue.

Fortunately, even newer technology provides a solution to both of these problems. *Virtual Private Networks (VPNs)* allow corporations to transmit data securely over the Internet, reducing the security threat to transmitted data and sharply reducing the cost of phone services.

What is a VPN?

A VPN allows a corporation to make their applications and data securely available to all corporate users and branches, no matter where they are in the world, as long as they have access to the Internet. VPNs allow secure connections between two machines, a machine and a subnet, or between two subnets.

Let’s look at an example. Company A, located in Boston, has branch offices in California, Texas, and Florida. Each of the branch offices send weekly sales reports to the home office. Before Company A installed a VPN, each of the branch offices dialed a corporate phone number to transmit the sales report to the home office. After Company A installed their VPN, the branches could connect to the Internet via their local *Internet Service Provider (ISP)*, connect to the home office’s intranet via the Internet, and use the VPN to transmit the data. What was previously a costly long-distance call is now a local call. And there is a big bonus — an increased level of security and privacy. Data is protected as it travels from sender to receiver — through the ISP, Internet, and any routers and gateways on its path. A VPN gives users data privacy, data integrity, and data origin authentication.

Companies that install VPNs can also use them to make their internal data available to trusted companies and individuals (for example, suppliers and consultants). This arrangement can save all parties time, money, and other resources. In addition to letting legitimate users send and receive data securely, a VPN used in conjunction with a firewall keeps unwanted users off your intranet. (A *firewall* controls the machines that an external host can see on a company's intranet, and the services that the host can access. A firewall also controls the machines that a host on a company's intranet can see on the internet, and the services that the host can access.)

In addition to the advantages of increased security and reduced costs, VPNs also prevent Internet Service Providers (ISPs) from reading any cleartext messages (that is, unencrypted messages), and give you an additional level of security against internal attacks.

How does a VPN work?

A VPN extends a company's *intranet* (that is, its internal network) across the Internet, creating a secure private *tunnel*. How does this work? A VPN uses a tunneling protocol (for example, Internet Protocol Security (IPSec)) and encryption to protect data from the time it leaves the sender to the time it reaches the designated recipient.

What do you need to protect?

It is critical that you protect a wide variety of information stored on your machines or transmitted to other entities (for example, banks, clients, business partners, and state and federal tax agencies):

- Employee records
- Payroll records
- User passwords and accounts
- Customer sales records
- Product research and development files
- Source code files

Other security concerns include attackers gaining access to your intranet and performing a variety of attacks:

- Deleting or downloading important files
- Reading email
- Crashing machines

- Prevent authorized users from accessing machines (denial of service attack)
- Sniffing packets off the wire to obtain user passwords and other information

The security of your data, machines, and networks is very important, and PGPnet is designed to eliminate many of the threats that continue to plague networks.

PGPnet features

The PGPnet program includes the following features:

- Secure peer-to-peer communication — no intermediary gateway is required.
- Simple user interface.
- A list of all active PGPnet Security Associations at a glance. (A *Security Association (SA)* contains information that identifies how two machines communicate with each other.)
- Automatic re-key (that is, initialization and negotiation) of expiring Security Associations.
- Log information, used for diagnostics, is displayed in easy-to-read format — no need to search through log files.

What is PGPnet?

PGPnet, a *Virtual Private Network (VPN)*, is an easy-to-use encryption application that allows you to communicate securely and economically with other PGPnet users. PGPnet, a standards-based product based upon the IETF IPSec and IETF IKE (Internet Key Exchange) protocols, extends the IKE protocol to add support for PGP key authentication.

PGPnet maintains the privacy, integrity, and authenticity of information sent from a PGPnet host to a secure host, subnet, or gateway.

- A *secure host* is a machine running PGPnet or another IPSec-compatible peer-to-peer capable client software (that is, software that allows hosts to communicate directly with each other).

- A *secure gateway* is a firewall or other gateway machine that tunnels packets through it for authorized parties. In this case, authorized means the certificate or shared passphrase of the client software is configured as acceptable on the gateway. (When you use PGPnet, you can elect to communicate with a host using your PGP key, an X.509 certificate, or a shared passphrase.)
- A *secure subnet* is one that has up to 254 machines behind it that are generally running PGPnet or a compatible client software. The secure subnet designation allows you or your administrator to identify a number of machines in the same IP address range that are known to be IPSec compatible. Note that secure subnets do not have to be behind gateways.

TIP: If a subnet has many secure hosts but a small number of insecure hosts, setup the subnet as a secure subnet and then add insecure hosts for each exception.

You can communicate securely with PGPnet users on your own corporate intranet and with other PGPnet users throughout the world. You can communicate with gateways, subnets, and hosts that you (or your PGPnet administrator, if applicable) have identified as secure. PGPnet gives you the ability to send data securely across the Internet and other untrusted networks.

What is a Security Association?

The first time a local machine communicates with a remote machine, PGPnet performs an Internet Key Exchange (IKE) negotiation and creates a Security Association.

- During the *IKE negotiation*, the two machines establish how they will communicate with each other (for example, type of encryption, duration of Security Association, and authentication method).
- The resulting *Security Association (SA)* contains information that identifies how the two machines are communicating.

PGPnet records and monitors all SAs that your machine initiates and that other machines initiate with your machine. When an SA that your machine initiated is close to expiration, PGPnet initiates another SA with the remote host. You can view all active SAs on PGPnet's **Status** panel. For more information on the **Status** panel, see “Viewing the Status Panel” on page 163.

PGPnet's two modes: tunnel and transport

PGPnet uses tunnel mode to communicate with hosts or subnets behind a secure gateway, and transport mode for peer-to-peer communications between two secure hosts that do not have a gateway between them.

What is tunnel mode?

Tunneling occurs when the machine running PGPnet sends packets through a secure gateway to a host or subnet behind the gateway. (In the PGPnet Hosts window, the destination host or subnet is indented beneath the gateway.) Packets sent to such hosts are *tunneled*. That is, the entire packet sent to the destination is physically placed inside another packet, encrypted, and then sent to the gateway.

What is transport mode?

PGPnet is fully capable of peer-to-peer secure communications. Two machines running PGPnet can communicate securely—no matter where they are on the internet. A secure gateway is not necessary. This type of communication is called *transport mode*. There is no secure gateway or firewall, and packets are transmitted securely from the source machine to the destination machine. In this mode, packets are encrypted and authenticated.

How does PGPnet communicate with secure and insecure hosts?

The following paragraphs describe how PGPnet communicates with hosts:

<u>Secure host with no secure gateway between hosts</u> — PGPnet packets are encrypted once and authenticated to their destination (transport mode).

<u>Insecure host behind secure gateway</u> — PGPnet tunnels packets to the gateway, and the gateway forwards the packets to the final destination (tunnel mode).

How do you use PGPnet?

If you have a PGPnet administrator, PGPnet may be configured when you install the software.

If you do not have a PGPnet administrator or if PGPnet is not pre-configured, you must install PGPnet, select your authentication key or certificate (or both), and configure hosts, gateways, and subnets to PGPnet via the **Host/Gateway** dialog box.

When PGPnet is configured, the software runs in the background. Any time you attempt to communicate with another machine (for example, via email or web browser), PGPnet checks to see if there is an active SA for the machine.

- If there is an SA for the target machine, PGPnet transmits your communication according to the terms of the existing SA.
- If there is no SA for the target machine and the machine is secure, PGPnet initiates an IKE negotiation which establishes an SA, and transmits your communication.
- If there is no SA for the target machine and the machine is not secure, PGPnet handles the communication according to the Security settings on the **General** panel (**Edit—>Preferences—>General**). That is, if both Require secure communications with all hosts and Allow communications with unconfigured hosts are checked, PGPnet only allows the machines to communicate securely.

NOTE: This is potentially dangerous as you will not be able to talk to DNS or DHCP servers unless they are running PGPnet or are explicitly designated as insecure hosts.

Please note the following:

- All SAs are terminated when you reboot your machine or put it in sleep mode. As a result, any machine that you have not communicated with since the last time you rebooted requires a new IKE negotiation.
- PGPnet is always listening for SA requests from other machines.

Securing your TCP/IP configurations

The first time you start PGPnet, a dialog asks you to select the TCP/IP configurations that you want to secure. Secure the configurations that you will use to communicate with other PGPnet hosts. For example, if the TCP/IP connection that you will use to communicate with other PGPnet hosts is set to “Connect via Ethernet,” PGPnet must secure that TCP/IP configuration. If the connection you want to secure is the remote access dialup connection, PGPnet must secure the TCP/IP configuration that corresponds to the dialup connection (usually “Connect via PPP”). If the network connection is via AppleTalk, then PGPnet must secure the TCP/IP configurations that use AppleTalk.

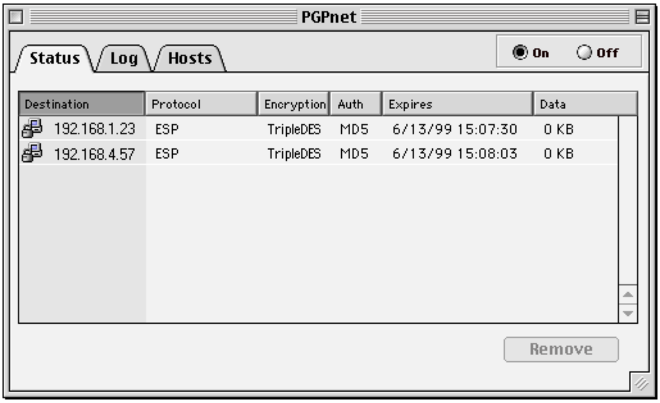
PGPnet can secure one or more TCP/IP configurations; as a result, you can elect to secure all TCP/IP configurations if desired.

You can change your selection at any time by selecting **Configurations** from the **File** menu.

Starting the PGPnet program

To start PGPnet

1. Click **PGPmenu** and select **PGPnet** (the **PGPmenu** appears as an icon in the menu bar of the Finder).
2. The PGPnet window displays on your screen (see Figure 8-1).



● On ○ Off

Destination	Protocol	Encryption	Auth	Expires	Data
192.168.1.23	ESP	TripleDES	MD5	6/13/99 15:07:30	0 KB
192.168.4.57	ESP	TripleDES	MD5	6/13/99 15:08:03	0 KB

Figure 8-1. The PGPnet window

The default setting for PGPnet is on. Use the buttons in the upper right corner of the window to turn PGPnet on and off. If however, PGPnet is turned off and the machine is rebooted, PGPnet will be off at reboot. For more information, see “Turning PGPnet off” on page 162 and “Turning PGPnet on” on page 162.

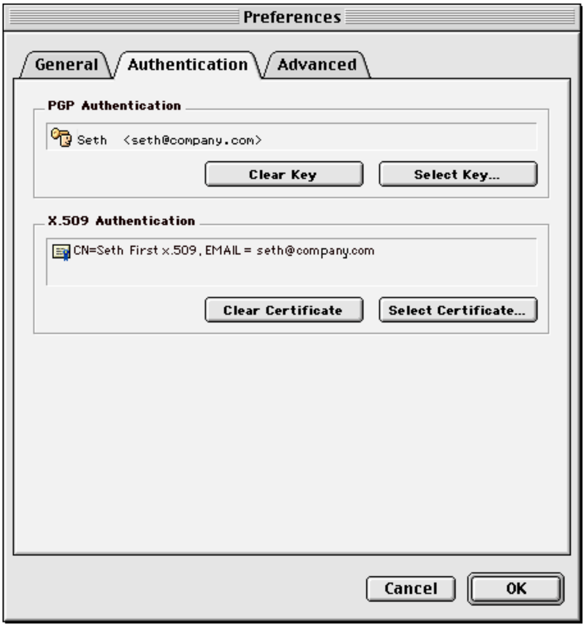
Selecting your authentication key or certificate

The first step that you must take before you use PGPnet is to select the key and / or X.509 certificate that you will use for authentication purposes. If you do not have an existing key pair or X.509 certificate, see Chapter 3, “Making and Exchanging Keys.”.

To select your authenticating key and/or certificate:

1. Click the **Edit** menu and select **Preferences**.
2. Click the **Authentication** tab (see Figure 8-2 on page 160).
3. Select the key and / or the certificate that you will use to authenticate (click **Select Key** or **Select Certificate**). Note that the key or certificate must be part of a key pair; you must have the private key. PGPnet displays the selected key or certificate in the **PGP Authentication** or **X.509 Authentication** box.
4. Click **OK**. You are prompted to enter the passphrase for the key or certificate you selected.
5. Enter the passphrase and click **OK**.

IMPORTANT: If you are creating a VPN connection with another PGPnet host, and using PGPkeys for authentication, you must both use the same type of PGP key. You cannot negotiate an SA if one side of the connection uses an RSA key and the other side uses a Diffie-Hellman key.



Seth <seth@company.com>

CN=Seth First x.509, EMAIL = seth@company.com

Figure 8-2. The Authentication Panel

The PGPnet window at a glance

When PGPnet is active, there are four menus in the menu bar:

- **File** (**Configurations** and **Quit**)
- **Edit** (Preferences)
- **View** (**Status**, **Log**, and **Hosts**)
- **Help** (**Help Center**, **Show Balloons**, and **PGP Help**)

There are three panels on the PGPnet window:

- **Status** Panel — Use to review the status of existing SAs (see "Viewing the Status Panel" on page 163).
- **Log** Panel — Use to review log entries for diagnostic purposes (see "Viewing the Log Panel" on page 164).
- **Hosts** Panel — Use to add, edit, or remove entries to PGPnet's host list and to establish and terminate SAs (see "Using the Hosts Panel" on page 165).

The default setting for PGPnet is on. Use the radio buttons in the top right corner of the window to turn PGPnet on and off.

Using PGPnet's menus

Use PGPnet's menus to perform the following tasks:

To...	Do this...
Display the **Log** panel	Click the **View** menu and select **Log**.
Display the **Status** panel	Click the **View** menu and select **Status**.
Display the **Hosts** panel	Click the **View** menu and select **Hosts**.
Display **Preferences** (includes **General**, **Authentication**, and **Advanced** tabs)	Click on the **Edit** menu and select **Preferences**.
Display the **Configurations** window	Click the **File** menu and select **Configurations**.
Quit PGPnet	Click the **File** menu and select **Quit**.

Turning PGPnet off

There may be times when you want to turn PGPnet off. For example, for diagnostic purposes. Turning PGPnet off allows all communication with all machines to pass through unmodified and unsecured.

To turn PGPnet off, click the **Off** radio button on the PGPnet window.

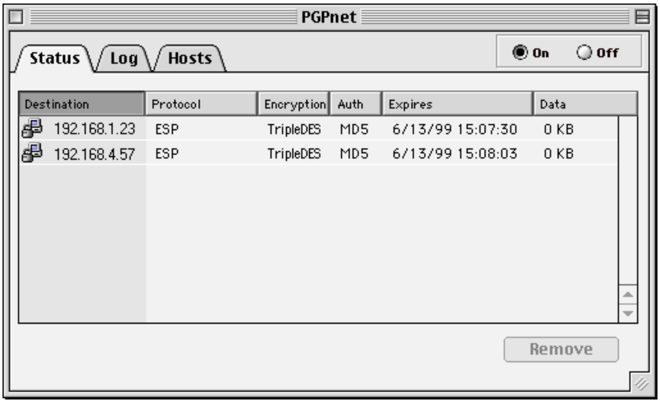
Turning PGPnet on

To turn PGPnet on, click the **On** radio button on the PGPnet window.

Quitting PGPnet

Select Quit from PGPnet’s **File** menu.

Note that quitting PGPnet does not disable the PGPnet service or terminate SAs.



Destination	Protocol	Encryption	Auth	Expires	Data
192.168.1.23	ESP	TripleDES	MD5	6/13/99 15:07:30	0 KB
192.168.4.57	ESP	TripleDES	MD5	6/13/99 15:08:03	0 KB

Figure 8-3. PGPnet window

Using PGPnet

When PGPnet is on, it is running in the background. To communicate with a machine, use your software (for example, email or web browser) as you normally would. PGPnet evaluates each communication and encrypts and tunnels as required.

Viewing the Status Panel

The **Status** panel in the PGPnet window lists active PGPnet SAs and, if applicable, tells you when they expire. An SA may be terminated when it reaches a certain byte limit (for example, 4 MB of data is transmitted over the SA), or after a certain amount of time. The length of an SA is negotiated when it is initiated. When PGPnet negotiates the SA, it sets an expiration value and automatically creates a new SA when the SA reaches that expiration value and expires. (The SA expiration value is user-configurable; for more information, see “Setting key expiration values” on page 182.)

- If your machine initiated an SA and the SA is about to expire, PGPnet automatically initiates the negotiation of a new SA to replace the expiring SA. As a result, there may be times when the **Status** panel displays two SAs for the same machine.
- When you establish an SA with another host, PGPnet uses the most restrictive expiration values set by either of the two hosts. As a result, you may see an SA expire before your maximum expiration value is met.

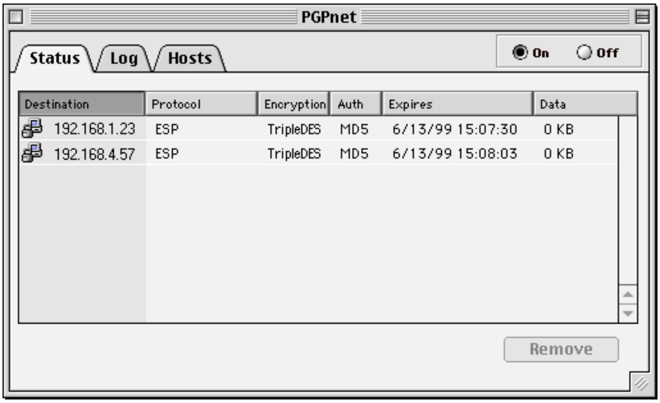
The following table describes the information that PGPnet’s **Status** panel displays for each SA:

Column	Description
Destination	IP address of target host or gateway.
Protocol	Type of protocol negotiated, for example, AH, ESP, or IPCOMP.
Encryption	Type of encryption algorithm negotiated. If it is an authentication-only SA, this column can be empty. Types of encryption include TripleDES or CAST.
Auth (Authentication)	Type of authentication algorithm negotiated. This column can be empty or contain one of the following: HMAC MD5 or HMAC SHA. If both ESP and AH protocols are used, this column can contain two entries.
Expires	Date and time that the SA expires (mm/dd/yy hh:mm:ss AM or PM), or displays “Never” if the SA’s expiration is based only on MB rather than time.
Data	Bytes sent and maximum number of MB that the SA will transport before expiring. Format is N XX/M YY where N represents bytes transferred, XX is either KB, MB or GB, M is expiration byte count, and YY is MB or GB.

Use the **Remove** feature to terminate an SA. Terminate an SA when you think that it has been compromised, if you know that the target host is down, or for any reason that you think the connection should be terminated.

Use the **On** and **Off** buttons to turn PGPnet on or off.

You can also click the **Log** tab to view recent log entries.



Destination	Protocol	Encryption	Auth	Expires	Data
192.168.1.23	ESP	TripleDES	MD5	6/13/99 15:07:30	0 KB
192.168.4.57	ESP	TripleDES	MD5	6/13/99 15:08:03	0 KB

Figure 8-4. The Status Panel

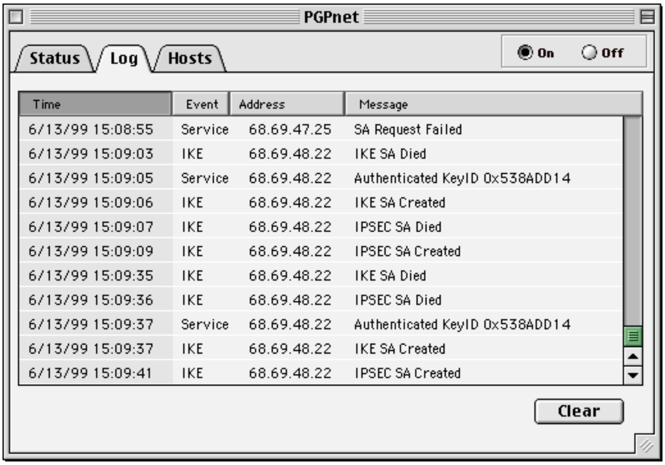
Viewing the Log Panel

The **Log** panel shows system and service errors, when they occurred (date and time), and a description of the error. Use this information to help resolve problems that occur.

Use the **Clear** button to clear current log information from the log file and screen.

The following table describes the information that PGPnet displays for each log entry:

Column	Description
Time	Date and time error occurred in format mm/dd/yy hh:mm:ss AM or PM
Event	Type of event, Service, IKE, IPSec, PGP, or System error.
Address	IP address of the remote host.
Message	Text that describes the type of error (for example, Unable to establish Security Association with peer).



Time	Event	Address	Message
6/13/99 15:08:55	Service	68.69.47.25	SA Request Failed
6/13/99 15:09:03	IKE	68.69.48.22	IKE SA Died
6/13/99 15:09:05	Service	68.69.48.22	Authenticated KeyID 0x538ADD14
6/13/99 15:09:06	IKE	68.69.48.22	IKE SA Created
6/13/99 15:09:07	IKE	68.69.48.22	IPSEC SA Died
6/13/99 15:09:09	IKE	68.69.48.22	IPSEC SA Created
6/13/99 15:09:35	IKE	68.69.48.22	IKE SA Died
6/13/99 15:09:36	IKE	68.69.48.22	IPSEC SA Died
6/13/99 15:09:37	Service	68.69.48.22	Authenticated KeyID 0x538ADD14
6/13/99 15:09:37	IKE	68.69.48.22	IKE SA Created
6/13/99 15:09:41	IKE	68.69.48.22	IPSEC SA Created

Figure 8-5. The Log Panel

Using the Hosts Panel

The **Hosts** panel displays secure gateways, subnets, and hosts. If an arrow appears to the left of an item, click the arrow to expand the display and view other entries associated with that item.

The following table describes the information displayed for each entity.

Column	Description
Name	Descriptive name of host, subnet, or gateway.
Address	IP address of host, subnet, or gateway.
Subnet	If the host entry is a subnet, this field displays the subnet mask. Otherwise, this field is blank.
Authentication	An icon appears, indicating the type of authentication used for the host entry. • A key icon indicates public-key cryptography authentication. • A certificate icon indicates X.509 certificate authentication. • An ear icon indicates shared secret authentication. • No icon indicates that the configured host entry is insecure.
SA	Displays a green dot when there is an SA with the host. If there is no SA with the host, the column is blank.

The following table describes the buttons on the **Hosts** panel.

Button	Description
Edit	Displays the values for the selected item in the **Host/Gateway** dialog box.
Remove	Removes selected host entry.
Add	Activates the **Host/Gateway** dialog.
Connect / Disconnect	Connect establishes an SA; Disconnect terminates an SA.

TIP: Hold the command key down to display keystroke commands for the buttons on the **Hosts** panel.

The Connect and Disconnect buttons

Use the **Connect** button to establish an SA with a configured host. Select the host, then click the **Connect** button. The **Connect** button beeps when an inappropriate host entry is selected (for example, when you select a secure subnet or insecure host that is not behind a gateway).

Use the **Disconnect** button to terminate an SA with a configured host. Select the host, then click the **Disconnect** button.

For more information about establishing an SA, please see “Establishing an SA” on page 166.

Establishing an SA

Establish an SA using PGP keys authentication

Follow the steps below to establish an SA with another host using PGP keys for authentication.

To establish an SA with another host using PGP keys for authentication:

1. Verify that each system has a network connection.
2. Install PGPnet on both systems.

   During installation you must select the appropriate TCP/IP configurations for PGPnet. For example, if the network connection is via ethernet, PGPnet must secure configurations that use ethernet; if the network connection is via AppleTalk, then PGPnet must secure the configurations that use AppleTalk. You can secure more than one TCP/IP configuration.

3. After installing PGPnet, reboot both systems.
4. Verify that each system has an authentication key set in the **PGP Authentication** section of the **Authentication** panel (**Edit—>Preferences—>Authentication**).
5. Exchange, sign, and validate the public keys that each system is using for authentication. For more information, see Chapter 3, “Making and Exchanging Keys.”

TIP: For scalability, use a trusted third-party or CA for this.

6. At least one user must create an entry in PGPnet’s host list for the other system. You must know the other system’s domain name or IP address. Verify that the entry identifies the host as a secure host (if the host’s entry in the **Hosts** panel displays a lock icon, then it is a secure host).
7. Select the host’s entry on the **Hosts** panel and click **Connect**. If the connection is successful, a green dot appears in the SA column.

Establish an SA using X.509 certificate authentication

Follow the steps below to establish an SA with another host using X.509 certificates for authentication.

To establish an SA with another host using X.509 certificates for authentication:

1. Verify that each system has a network connection.
2. Install PGPnet on both systems.

   During installation you must select the appropriate TCP/IP configurations for PGPnet. For example, if the network connection is via ethernet, PGPnet must secure configurations that use ethernet; if the network connection is via AppleTalk, then PGPnet must secure the configurations that use AppleTalk. You can secure more than one TCP/IP configuration.

3. After installing PGPnet, reboot both systems.
4. Verify that each system has an authentication certificate in the **X.509 Authentication** section of the **Authentication** panel (**Edit—>Preferences—>Authentication**).

5. Ensure that the Root CA for the X.509 certificate exists and is signed and fully trusted on both systems. Both systems must have the same Root CA.

6. At least one user must create an entry in PGPnet's host list for the other system. You must know the other system's domain name or IP address. Verify that the entry identifies the host as a secure host (if the host's entry in the **Hosts** panel displays a lock icon, then it is a secure host).

7. Click on the host's entry on the **Hosts** panel and click **Connect**. If the connection is successful, a green dot appears in the SA column.

Establish an SA using shared secret passphrase authentication

Follow the steps below to establish an SA with another host using a shared secret passphrase for authentication.

To establish an SA with another host using shared secret for authentication:

> **WARNING:** Unlike traditional PGP passphrases, Shared Secret passphrases are stored on your computer unencrypted. This presents a potential security risk. To avoid this risk, use keys or certificates.

1. Verify that each system has a network connection.

2. Install PGPnet on both systems.

   During installation you must select the appropriate TCP/IP configurations for PGPnet. For example, if the network connection is via ethernet, PGPnet must secure configurations that use ethernet; if the network connection is via AppleTalk, then PGPnet must secure the configurations that use AppleTalk. You can secure more than one TCP/IP configuration.

3. After installing PGPnet, reboot both systems.

4. Both users must create an entry in PGPnet's host list for the other system. You must know the other system's domain name or IP address, and agree on a shared secret passphrase.

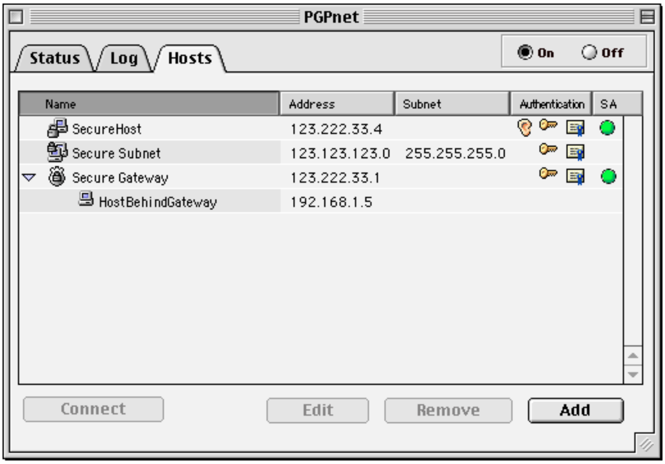
   For more information on configuring a secure host, see "Adding a host, subnet, or gateway" on page 169.

5. Click on the host's entry on the **Hosts** panel and click **Connect**. If the connection is successful, a green dot appears in the SA column.

Adding a host, subnet, or gateway

If you are in a corporate environment with a PGPnet administrator, many of the hosts, subnets, and gateways that you communicate with may have been preconfigured by your administrator. Each preconfigured host, subnet, and gateway is an entry in PGPnet's host list. You can use PGPnet's **Host/Gateway** dialog to add additional entries to the host list.

If you do not have a PGPnet administrator or hosts, subnets, or gateways are not configured when you install PGPnet, use the **Host/Gateway** dialog to add the necessary hosts, subnets, and gateways.



Name	Address	Subnet	Authentication	SA
SecureHost	123.222.33.4			●
Secure Subnet	123.123.123.0	255.255.255.0		
Secure Gateway	123.222.33.1			●
HostBehindGateway	192.168.1.5			

Figure 8-6. The Hosts Panel

What you need to know

The following paragraphs identify information that you need to add a host, subnet, or gateway.

Table 8-1. What you must know to add hosts, gateways, and subnets

To:	You must know:
Add a secure host	Host domain name or IP address
Add a subnet	IP address and subnet mask
Add a gateway	Host domain name or IP address
Add a host or subnet behind a configured gateway	Host domain name or IP address

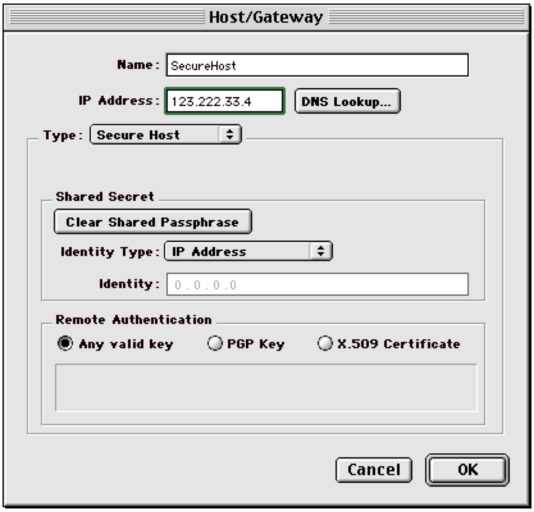
To...	See page...
Add a host	page 170
Add a subnet	page 172
Add a gateway	page 174
Add a host or subnet behind a configured gateway	page 176

Adding a host

NOTE: To add a host behind an existing configured gateway, see "Adding a host or subnet behind a configured gateway" on page 176.

Use PGPnet's **Host/Gateway** dialog to add a host entry to the host list.

1. Click the **Hosts** tab.
2. Click **Add**. PGPnet displays the **Host/Gateway** dialog.



Host/Gateway

Name: SecureHost

IP Address: 123.222.33.4

Type: Secure Host

Shared Secret

Identity Type: IP Address

Identity: 0.0.0.0

Remote Authentication

● Any valid key ○ PGP Key ○ X.509 Certificate

Figure 8-7. The Host/Gateway Dialog

3. Enter a descriptive name and IP address for the host. If you do not know the host’s IP address, click **DNS Lookup**. PGPnet displays a dialog box. Enter the domain name for the host and click **OK**. PGPnet searches for the IP address.

   - If PGPnet finds the IP address, it displays the IP address; click Use to use the IP address in the **Host/Gateway** dialog.
   - If PGPnet does not find an IP address for the host, it advises you.

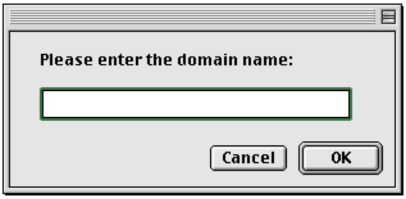


Figure 8-8. DNS Lookup Dialog

4. Select the type of host (**Insecure Host**) from the **Type** drop down menu.

   - If you select **Insecure Host**, click **OK**. PGPnet adds the host’s entry to the **Hosts** panel.
   - If you select **Secure Host**, proceed to the next step.

5. If you are not using a shared secret for authentication, go to Step 9. If you are using a shared secret for authentication with this machine, click **Set Shared Passphrase**. PGPnet displays a dialog box.

6. Enter the passphrase that you intend to use for authentication; enter the passphrase a second time in the **Confirmation** box. Click **OK**. Note that both hosts must configure the same shared secret passphrase.

   WARNING: Unlike traditional PGP passphrases, Shared Secret passphrases are stored on your computer unencrypted. This presents a potential security risk. Use keys or certificates to avoid this risk.

7. Select how you want to identify yourself to the remote computer from the **Identity Type** drop down menu (applies only if shared secret authentication is used): IP Address, Host Domain Name, User Domain Name, or Distinguished Name.

   IP Address — by the IP address of this computer [nnn.nnn.nnn.nnn]

   Domain Name — by the domain name of this computer

User Domain Name — by a user and host domain name which you specify [for example, username@computerName.nameOfNetwork]

Distinguished Name — by a text string which you specify, such as “CN=”Bob Jones”,_C=US,_O=”Acme,_Inc.””

8. Enter the IP address or name (Domain Name, User Domain Name, or Distinguished Name) in the **Identity** box.

9. The controls in the **Remote Authentication** section of the **Host/Gateway** dialog allow you to require the remote host to present a specific PGP key or X.509 certificate each time the host attempts to establish an SA with your machine. The default setting is **Any valid key**.

   For information about setting a remote authentication key, refer to “Requiring a host to present a specific key or certificate” on page 178.

10. Click **OK** to add the host entry to PGPnet’s host list.

Adding a subnet

❐ **NOTE:** To add a subnet behind an existing configured gateway, see “Adding a host or subnet behind a configured gateway” on page 176.

Use PGPnet’s **Host/Gateway** dialog to add subnet entries to the host list.

1. Click the **Hosts** tab.

2. Click **Add**. PGPnet displays the **Host/Gateway** dialog.

3. Enter a descriptive name and IP address for the subnet.

4. Select the type of host entry from the **Type** drop down menu: Insecure Subnet or Secure Subnet.

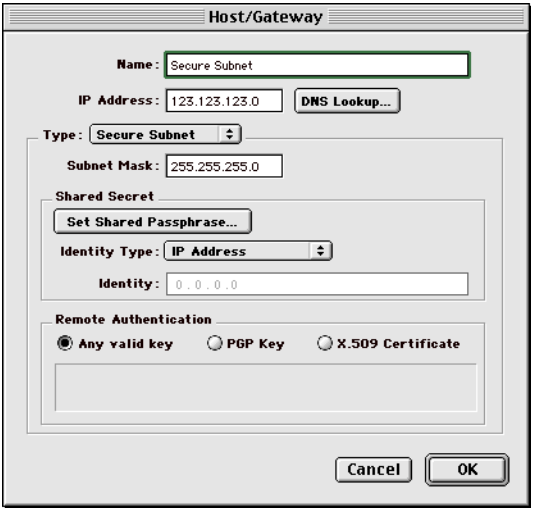
5. Enter the **Subnet Mask** for the subnet.

   - If you select **Insecure Subnet**, click **OK**. PGPnet adds the subnet’s entry to the **Hosts** panel.

   - If you select **Secure Subnet**, proceed to the next step.

6. If you are not using shared secret, go to Step 10. If you are using shared secret for authentication with this machine, click **Set Shared Passphrase**. PGPnet displays a dialog box.

NOTE: If you configure a subnet with shared secret passphrase, all machines in that subnet must be configured with the same shared secret passphrase.



Name: Secure Subnet

IP Address: 123.123.123.0

Type: Secure Subnet

Subnet Mask: 255.255.255.0

Identity Type: IP Address

Identity: 0.0.0.0

● Any valid key ○ PGP Key ○ X.509 Certificate

Figure 8-9. The Host/Gateway Dialog

7. Enter the passphrase that you intend to use for authentication; enter the passphrase a second time in the **Confirmation** box. Click **OK**. Note that both hosts must configure the same shared secret passphrase.

WARNING: Unlike traditional PGP passphrases, Shared Secret passphrases are stored on your computer unencrypted. This presents a potential security risk. Use keys or certificates to avoid this risk.

8. Select how you want to identify yourself to the subnet from the **Identity Type** drop down menu (applies only if shared secret authentication is used): IP Address, Domain Name, User Domain Name, or Distinguished Name. (Note: If you are using PGPnet with GVPN, PGPnet always uses your IP address.)

   IP Address — by the IP address of this computer [nnn.nnn.nnn.nnn]

   Domain Name — by the domain name of this computer

   User Domain Name — by a user and host domain name which you specify [for example, username@computerName.nameOfNetwork]

Distinguished Name — by a text string which you specify, such as "CN="Bob Jones",_C=US,_O="Acme,_Inc.""

9. Enter the IP address or name (Domain Name, User Domain Name, or Distinguished Name) in the **Identity** box.

10. The controls in the **Remote Authentication** section of the **Host/Gateway** dialog allow you to require the remote host to present a specific PGP key or X.509 certificate each time the host attempts to establish an SA with your host.

    For information about setting a remote authentication key, refer to "Requiring a host to present a specific key or certificate" on page 178.

🛎 **IMPORTANT:** If you select a specific PGP key or X.509 certificate within a secure subnet configuration, all users must use the same key to authenticate themselves.

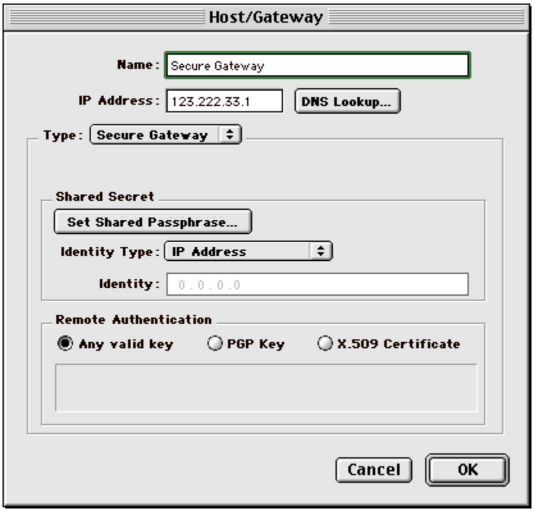
11. Click **OK** to add the host entry to PGPnet's host list.

Adding a gateway

Use PGPnet's **Host/Gateway** dialog to add gateway entries to the host list.

1. Click the **Hosts** tab.
2. Click **Add**. PGPnet displays the **Host/Gateway** dialog.
3. Enter a descriptive name and IP address for the gateway. If you do not know the gateway's IP address, click **DNS Lookup**. PGPnet displays a dialog box. Enter the domain name for the gateway and click **OK**. PGPnet searches for the IP address.
    - If PGPnet finds the IP address, it displays the IP address; click **Use** to use the IP address in the **Host/Gateway** dialog.
    - If PGPnet does not find an IP address for the host, it advises you.
4. Select the type of host from the **Type** drop down menu, **Secure Gateway**.
5. If you are not using shared secret, go to Step 9. If you are using shared secret for authentication with this machine, click **Set Shared Passphrase**. PGPnet displays a dialog box.
6. Enter the passphrase that you intend to use for authentication; enter the passphrase a second time in the **Confirmation** box. Click **OK**. Note that both hosts must configure the same shared secret passphrase.

WARNING: Unlike traditional PGP passphrases, Shared Secret passphrases are stored on your computer unencrypted. This presents a potential security risk. To avoid this risk, use keys or certificates.



Host/Gateway

Name: Secure Gateway

IP Address: 123.222.33.1

Type: Secure Gateway

Shared Secret

Identity Type: IP Address

Identity: 0.0.0.0

Remote Authentication

● Any valid key ○ PGP Key ○ X.509 Certificate

Figure 8-10. The Host/Gateway Dialog

7. Select how you want to identify yourself to the remote computer from the **Identity Type** drop down menu (applies only if shared secret authentication is used): IP Address, Domain Name, User Domain Name, or Distinguished Name.

   IP Address — by the IP address of this computer [nnn.nnn.nnn.nnn]

   Domain Name — by the domain name of this computer

   User Domain Name — by a user and host domain name which you specify [for example, username@computerName.nameOfNetwork]

   Distinguished Name — by a text string which you specify, such as “CN=”Bob Jones”,_C=US,_O=”Acme,_Inc.””

8. Enter the IP address or name (Domain Name, User Domain Name, or Distinguished Name) in the **Identity** box.

9. The controls in the **Remote Authentication** section of the **Host/Gateway** dialog allow you to require the remote host to present a specific PGP key or X.509 certificate each time the host attempts to establish an SA with your host.

   For information about setting a remote authentication key, refer to "Requiring a host to present a specific key or certificate" on page 178.

10. Click **OK** to add the host entry to PGPnet's host list.

Adding a host or subnet behind a configured gateway

Use PGPnet's **Host/Gateway** dialog to add a host or subnet behind a configured gateway.

1. Click the **Hosts** tab.

2. Select the configured gateway and click **Add**. PGPnet displays the **Host/Gateway** dialog.

3. Enter a descriptive name and IP address for the host. If you do not know the host's IP address, click **DNS Lookup**. PGPnet displays a dialog box. Enter the domain name for the host and click **OK**. PGPnet searches for the IP address.

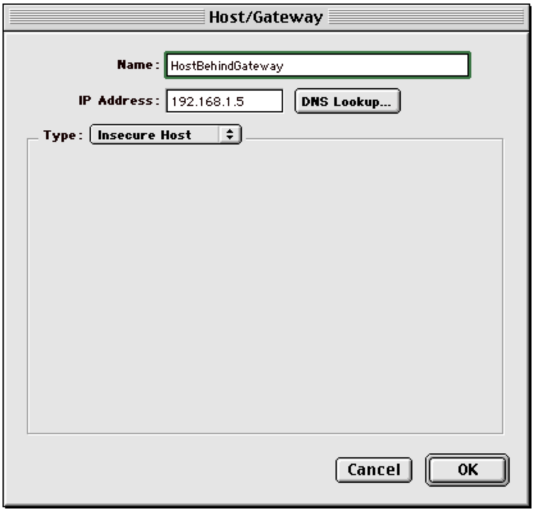
   - If PGPnet finds the IP address, it displays the IP address; click **Use** to use the IP address in the **Host/Gateway** dialog.

   - If PGPnet does not find an IP address for the host, it advises you.

4. Select the host type from the **Type** drop down menu: **Insecure Host** or **Insecure Subnet**.

5. If you are adding an insecure subnet, enter the subnet mask.

6. Click **OK** to add the host entry to PGPnet's host list.



Name: HostBehindGateway

IP Address: 192.168.1.5

Type: Insecure Host

Figure 8-11. The Host/Gateway Dialog

Modifying a host, subnet, or gateway entry

There may be times when you need to modify the configuration of a host, subnet, or gateway. For example, when an IP address, subnet mask, or host domain name changes. To modify a configuration, follow these instructions:

1. Click the **Hosts** tab.
2. Select the host, subnet, or gateway entry that you want to modify.
3. Click **Edit**.

   Shortcut: Instead of selecting the host and clicking **Edit**, double-click the host’s entry in the host list.
4. Make the required edits.
5. Click **OK**.

The PGPnet database is updated immediately. However, if the PGPnet service or driver are not operating normally, the PGPnet database is not updated until they are working properly. This may require a computer reboot.

Removing a host, subnet, or gateway entry

There may be times when you want to remove a configured host, subnet, or gateway. For example, when you feel that any entity is no longer secure. To remove a host, subnet, or gateway entry, follow these instructions:

1. Click the **Hosts** tab.

2. Select the host, subnet, or gateway entry that you want to remove.

3. Click **Remove**.

Requiring a host to present a specific key or certificate

You may want to require a host to present a specific key or certificate when the host attempts to establish an SA. If the host does not present the appropriate key or certificate, your system will refuse to communicate with the host.

To require a host to present a specific key or certificate:

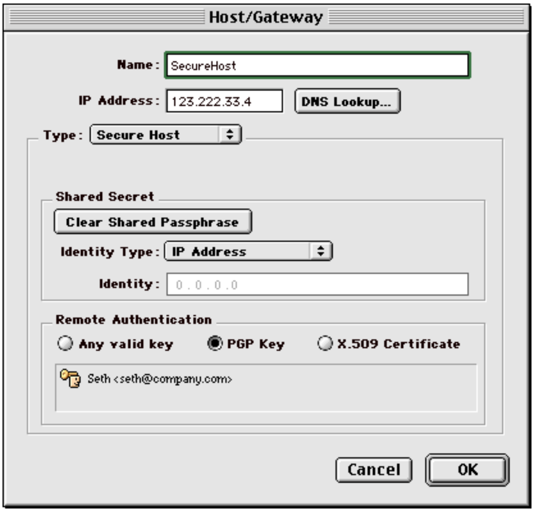
1. If you have not already done so, add the host, subnet, or gateway to PGPnet (for instructions, see "Adding a host, subnet, or gateway" on page 169). PGPnet adds an entry to the host list on the **Hosts** panel.

2. Select the entry on the **Hosts** panel and click **Edit**. PGPnet displays the **Host/Gateway** dialog. The **Authentication** section is at the bottom of the dialog.

3. You can require the host, subnet, or gateway to present a specific PGP key or X.509 certificate to authenticate itself.

   - To require a specific PGP key, click **PGP Key**. PGPnet displays a dialog box. Select the appropriate key and click **OK**. PGPnet displays the key in the **Remote Authentication** box. Click **OK** to close the **Host/Gateway** dialog.

   - To require a specific X.509 certificate, click **X.509 Certificate**. PGPnet displays a dialog box. Select the appropriate certificate and click **OK**. PGPnet displays the certificate in the Remote Authentication box. Click **OK** to close the **Host/Gateway** dialog.



Name: SecureHost

IP Address: 123.222.33.4

Type: Secure Host

Identity Type: IP Address

Identity: 0.0.0.0

○ Any valid key ● PGP Key ○ X.509 Certificate

Seth <seth@company.com>

Figure 8-12. Host/Gateway dialog

Viewing the General Preferences Panel

To view the **General** panel, select **Preferences** from the **Edit** menu.

Use the **General** panel to perform the following tasks:

- Control the security level of communications with hosts
- Require valid authentication keys from all hosts
- Set expiration values for Setup Keys (IKE) and Primary Keys (IPSec) which create Security Associations with other configured hosts

Controlling the security level of communications with hosts

Communicating securely with other hosts is one of the primary reasons to use PGPnet. PGPnet’s security features (encryption, authentication, and tunneling) allow you to transmit your data over the Internet or other public or private networks securely. Your data is protected as it travels over networks and machines that are not under corporate control. Any attempts by attackers to intercept, decipher, or alter the data are eliminated. Your data reaches its final destination intact.

PGPnet includes features that allow you to communicate with unconfigured hosts (that is, hosts that have not been added to the PGPnet host list), and also to require secure communications with all hosts.

Allow communications with unconfigured hosts and Require secure communications with all hosts

Use these two settings to control who you communicate with and to minimize the number of systems that you are required to add to the hosts list.

If most of the systems that you communicate with are not running PGPnet, use the wizard to add the few secure hosts to the hosts list and check the **Allow communications with unconfigured hosts** setting. This will allow you to communicate with both the secure hosts that you have identified in the hosts list and all other hosts.

If most of the systems that you communicate with are running PGPnet, use the wizard to add the few insecure hosts to the hosts list as insecure hosts and check the **Require secure communications with all hosts** setting. This will allow you to communicate with both the insecure hosts that you have identified in the hosts list and all other IPSec-compliant enabled hosts.

Allow communications with unconfigured hosts

Use this feature (**Edit—>Preferences**), to send and receive data that is not confidential or sensitive to and from hosts that are not configured in PGPnet. For example, you might want to use this feature if you routinely browse the web. This setting is enabled by default.

- To allow communications with unconfigured hosts, check this box.
- To disallow communications with unconfigured hosts, leave this box blank.

Require secure communications with all hosts

Use this feature (**Edit—>Preferences**), to require secure communications with all hosts. For example, if all of your company's systems are configured with PGPnet. This eliminates the need to identify each host.

When this box is checked, PGPnet negotiates an SA with each target machine before it allows communication. The default for this setting is off (unchecked).

- To require PGPnet to negotiate secure communications with all hosts, check this box.
- To allow insecure communications with all hosts, uncheck this box.

NOTE: If this feature is on, two machines configured as insecure hosts can still communicate with each other.

WARNING: This security feature is designed for environments where all machines are configured with PGPnet. When this feature is active (checked), it blocks communication from any machine that is not configured with PGPnet. As a result, if you are not in a PGPnet configured environment and you activate this feature, you may lose the bulk of your network traffic.

Require valid authentication key

Use this feature (**Edit—>Preferences**), to control whether PGPnet verifies that the keys presented by remote hosts are valid on the local keyring.

- To require PGPnet to verify that the keys presented by remote hosts are valid on the local keyring, make this setting active (checked). Use this setting if you only communicate with hosts who will use keys and certificates that are valid on your keyring.
- To instruct PGPnet to accept any key regardless of validity, make this setting inactive (unchecked). Use this setting when you are running PGPnet on servers (for example, mail or web servers) that allow connectivity with any client host. The server uses the appropriate key to authenticate itself to the client host, but the server accepts any key the client host presents. (In this case this setting is inactive (unchecked) for the server, and active (checked) for the client host.) The client host must have the server's trusted authentication key for this scenario to work.

IMPORTANT: When this box is inactive (unchecked), it overrides the **Any valid key** setting in the **Remote Authentication** section of the **Host/Gateway** dialog. When this occurs, the server accepts any key rather than any valid key. However, you can still use the **Host/Gateway** dialog to require a specific key or certificate for each host. For more information, see "Requiring a host to present a specific key or certificate" on page 178.

NOTE: All key authentications appear on the **Log** panel, and each entry displays the key ID.

NOTE: When this box is active (checked), and a PGP Key is selected as the **Remote Authentication** method (**Host/Gateway** dialog), both requirements apply (the machine must present the correct key, and the key must also be valid).

Setting key expiration values

You can set expiration values for Setup Keys (IKE) and Primary Keys (IPSec). These keys are responsible for creating your Security Associations. Values can be set in time (**Duration**) or data size (**Megabytes**).

Duration is displayed in the following manner:

08:10:33 (key expires in 8 hours, 10 minutes, and 33 seconds)

Megabytes is displayed in the following manner:

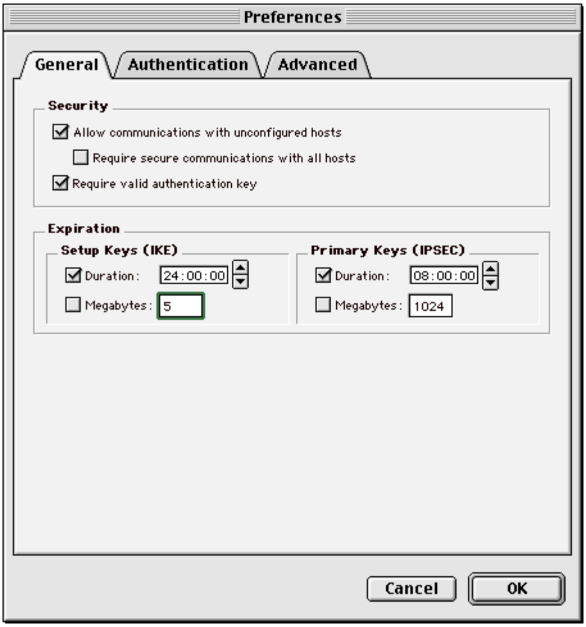
99 (key expires after 99 megabytes of data are transferred)

Note that when you establish an SA with another host, PGPnet uses the most restrictive expiration values set by either of the two hosts. As a result, you may see an SA expire before your maximum expiration value is met.

WARNING: Lowering the default value for Megabytes may result in multiple rekeyings when transmitting large files, which may, in turn, cause temporary interruption of normal network function.

To set expiration values for Setup Keys (IKE):

1. Display the **General** panel (**Edit—>Preferences**). The **Expiration** information appears in the bottom section of the **General** tab.
2. To set a duration for Setup Keys, click **Duration**. Use the up and down arrows next to the **Duration** field to set the appropriate time limit or enter a numeric value in each field: hh:mm:ss.
3. To set a data value in **Megabytes** for Setup Keys, click **Megabytes** and enter a numeric value.
4. Click **OK**.



Security

☑ Allow communications with unconfigured hosts

□ Require secure communications with all hosts

☑ Require valid authentication key

Expiration

Setup Keys (IKE)

☑ Duration: 24:00:00

□ Megabytes: 5

Primary Keys (IPSEC)

☑ Duration: 08:00:00

□ Megabytes: 1024

Figure 8-13. The General Panel

To set expiration values for Primary Keys (IPSec):

1. Display the **General** panel (**Edit—>Preferences**). The **Expiration** information appears in the bottom section of the **General** tab.

2. To set a duration for Primary Keys, click **Duration**. Use the up and down arrows next to the **Duration** field to set the appropriate time limit or enter a numeric value in each field: hh:mm:ss.

3. To set a data value in **Megabytes** for Primary Keys, click **Megabytes** and enter a numeric value.

4. Click **OK**.

Authenticating a connection

The controls on the **Authentication** panel allow you to perform the following tasks:

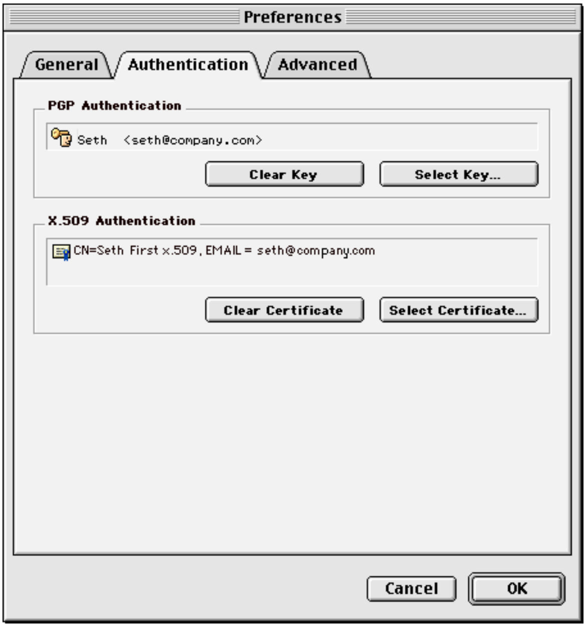
- Select a PGP key to authenticate your local machine (**PGP Authentication**).

- Select an X.509 certificate to authenticate your local machine (**X.509 Authentication**).

When you click **OK**, you are asked to enter the passphrase for the selected authentication key or certificate. Enter the passphrase and click **OK**. You are asked to enter this passphrase each time you login to PGPnet.

The following table describes the buttons on the **Authentication** panel.

Button	Description
Select Key	Displays a dialog box. Use this dialog box to select a key pair with which to authenticate your machine. You must then enter the passphrase for the selected key.
Clear Key	Clears the selected PGP key.
Select Certificate	Displays a dialog box. Use this dialog box to select an X.509 certificate with which to authenticate your machine. You must then enter the passphrase for the key to which the certificate is attached.
Clear Certificate	Clears the selected X.509 certificate.



Seth <seth@company.com>

CN=Seth First x.509, EMAIL = seth@company.com

Figure 8-14. The Authentication Panel

Advanced Panel

WARNING: The default settings on this panel allow you to communicate with PGPnet or strong-crypto GVPN users. Do not change the settings unless you are an experienced IPSec user.

The **Advanced** panel (**Edit—>Preferences**) displays the **Allowed Remote Proposals** and IKE and IPSec **Proposals**.

- The **Allowed Remote Proposals** section *tells PGPnet to accept any proposal from other users that includes any item checked (allowed) in these boxes*. The exceptions to this are the None items for Cipher and Hashes. Use the None items with extreme caution or not at all. If you check None for Ciphers (encryption), PGPnet accepts proposals that do not include encryption. If you check None for Hashes (authentication), PGPnet accepts proposals that do not include authentication.
- The IKE and IPSec **Proposals** sections *identify the proposals that you make to others*. Other users must accept exactly what is specified in at least one of your proposals for IKE and for IPSec.

Allowed Remote Proposals

The **Allowed Remote Proposals** portion of this panel identifies the types of ciphers, hashes, compression, and Diffie-Hellman keys that PGPnet allows. Only experienced IPSec users should make any changes to the settings on this panel:

Ciphers are algorithms used to encrypt and decrypt. To allow a specific type of cipher (CAST or TripleDES), place a check in the box to the left of the cipher. Check None with extreme caution or not at all, as it tells PGPnet to accept proposals that do not include encryption from other users.

A *hash function* takes a variable-sized input string and converts it to a fixed-sized output string. To allow a specific type of hash (SHA-1 or MD5), place a check in the box to the left of the hash function. Check None with extreme caution or not at all, as it tells PGPnet to accept proposals that do not include authentication from other users.

A *compression function* takes a fixed-sized input and returns a shorter, fixed sized output. There are two types of compression: LZS and Deflate. To allow a specific type of compression, place a check in the box to the left of the compression type.

NOTE: LZS and Deflate increase performance for low-speed communications such as modems and ISDN. LZS and Deflate decrease performance for fast-speed communications (for example, cable modem, DSL, T-1, and T-3). This is due to the overhead of the compression routines.

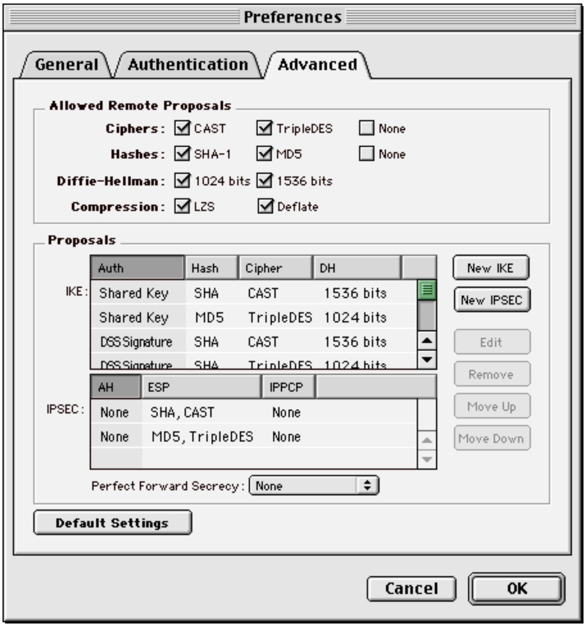
Diffie-Hellman is a key agreement protocol. To allow a specific key size (1024 or 1536), place a check in the box to the left of the key size.

To add an item to the Allowed Remote Proposals:

1. Display the Preferences window (**Edit—>Preferences**).
2. Click the **Advanced** tab.
3. Click the box to the left of the item; a check appears.
4. Click **OK**.

To remove an item from the Allowed Remote Proposals:

1. Display the Preferences window (**Edit—>Preferences**).
2. Click the **Advanced** tab.
3. Click the box to the left of the item; the check is removed.
4. Click **OK**.



Allowed Remote Proposals

Ciphers:	☑ CAST	☑ TripleDES	☐ None
Hashes:	☑ SHA-1	☑ MD5	☐ None
Diffie-Hellman:	☑ 1024 bits	☑ 1536 bits	
Compression:	☑ LZS	☑ Deflate	

Proposals

IKE:

Auth	Hash	Cipher	DH
Shared Key	SHA	CAST	1536 bits
Shared Key	MD5	TripleDES	1024 bits
DSS Signature	SHA	CAST	1536 bits
DSS Signature	SHA	TripleDES	1024 bits

IPSEC:

AH	ESP	IPPCP
None	SHA, CAST	None
None	MD5, TripleDES	None

Perfect Forward Secrecy: None

Figure 8-15. The Advanced Panel

Term	Description
Ciphers	An algorithm used to encrypt and decrypt. Types: CAST TripleDES When None is checked, PGPnet accepts proposals that do not include authentication from other users.
Hashes	A hash function takes a variable-sized input string and converts it to a fixed-sized output string. Types: SHA-1 (Secure Hash Algorithm) MD5 (Message Digest Algorithm). When None is checked, PGPnet accepts proposals that do not include authentication from other users.

Term	Description
Diffie-Hellman	Key agreement protocol. Sizes: 1024 bits 1536 bits
Compression	Takes a fixed-sized input and creates a smaller fixed-sized output. Types: LZS Deflate NOTE: LZS and Deflate increase performance for low-speed communications such as modems and ISDN. LZS and Deflate decrease performance for fast-speed communications (for example, cable modem, DSL, T-1, and T-3). This is due to the overhead of the compression routines.

Proposals

Use the **Proposals** portion of the **Advanced** panel to add, edit, remove, or reorder your existing proposals. Again, only experienced IPSec users should make any edits to this panel. The IKE and IPSec proposals tell PGPnet what proposals to make to other users; proposals must be accepted exactly as specified. Note that PGPnet allows a minimum of one and maximum of 16 proposals for both IKE and IPSec proposals.

NOTE: LZS and Deflate increase performance for low-speed communications such as modems and ISDN. LZS and Deflate decrease performance for fast-speed communications (for example, cable modem, DSL, T-1, and T-3). This is due to the overhead of the compression routines.

The following table identifies the types of Authentication, Hash, Ciphers, and Diffie-Hellman used in IKE proposals.

Term	Description
Authentication	Means of verifying information such as identity. Types: Shared Key (a secret key is shared by two or more users) DSS Signature (a Digital Signature Standard signature) RSA Signature
Hash	A hash function takes a variable size input string and converts it to a fixed size output string. Types: SHA (Secure Hash Algorithm) MD5 (Message-Digest Algorithm).
Cipher	An algorithm used to encrypt and decrypt. Types: CAST TripleDES
DH (Diffie-Hellman)	A key agreement protocol. Sizes: 1024 bits 1536 bits.

The following table identifies the types of AH, ESP, and IPPCP used in IPSec Proposals.

Term	Description
AH	Authentication Header, a sub-protocol of IPSec that handles authentication only. In addition, authenticates various pieces of the IP header. Useful when encryption is unnecessary, for example, when an ESP communication is tunneled through a gateway with AH. Types: SHA and MD5.
ESP	Encapsulating Security Payload, a sub-protocol of IPSec that handles both encryption and authentication. Hash types: None, SHA, and MD5. Cipher types: None, CAST, and TripleDES.
IPPCP	IP Payload Compression Protocol. Types: Deflate and LZS. NOTE: LZS and Deflate increase performance for low-speed communications such as modems and ISDN. LZS and Deflate decrease performance for fast-speed communications (for example, cable modem, DSL, T-1, and T-3). This is due to the overhead of the compression routines.

Perfect Forward Secrecy

All IPSec proposals use the same Diffie-Hellman setting: None, 1024, or 1536 bits.

Adding an IKE or IPSec proposal

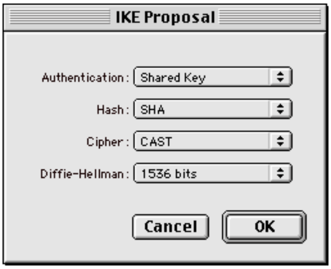
To add an IKE or IPSec proposal:

1. Display **Preferences** (**Edit—>Preferences**).
2. Click the **Advanced** tab.
3. Click **New**, and select IKE or IPSec.
4. Make the appropriate selections in the IKE or IPSec Proposal popup window.

5. Click **OK**.

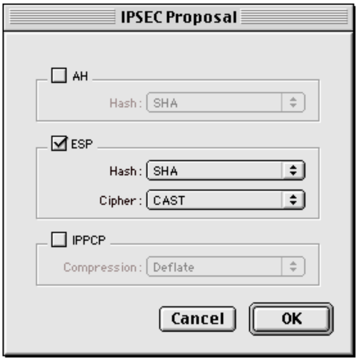
6. If you are adding an IPSec proposal, select the appropriate Diffie-Hellman setting (None, 1024, and 1536) in the **Perfect Forward Secrecy** setting. All IPSec proposals use the same Diffie-Hellman setting.

7. Click **OK**.



Authentication: Shared Key
Hash: SHA
Cipher: CAST
Diffie-Hellman: 1536 bits

Figure 8-16. IKE Proposal Dialog



☐ AH
Hash: SHA

☑ ESP
Hash: SHA
Cipher: CAST

☐ IPPCP
Compression: Deflate

Figure 8-17. IPSec Proposal Dialog

Editing an IKE or IPSec proposal

To edit an IKE or IPSec proposal:

1. Display **Preferences** (**Edit—>Preferences**).

2. Click the **Advanced** tab.

3. Select the **Proposal**.

4. Click **Edit**.
5. Make the appropriate changes in the IKE or IPSec Proposal popup window.
6. Click **OK** on the popup window.
7. Review the setting displayed in the **Perfect Forward Secrecy** box. Note that all IPSec proposals use the same Diffie-Hellman setting. Change the setting if required.
8. Click **OK** on the **Advanced** panel.

Removing an IKE or IPSec proposal

To remove an IKE or IPSec proposal:

1. Display the Preferences window (**Edit—>Preferences**).
2. Click the **Advanced** tab.
3. Click the proposal.
4. Click **Remove**.
5. Click **OK**.

Reordering IKE or IPSec proposals

To reorder IKE or IPSec proposals:

1. Display the Preferences window (**Edit—>Preferences**).
2. Click the **Advanced** tab.
3. Select the proposal.
4. To move the proposal up, click **Move Up**. To move the proposal down, click **Move Down.**
5. Click **OK**.

Default Settings button

Use this button to restore the default settings for all fields on this screen. In most cases, the default settings will be sufficient to establish SAs and use PGPnet.

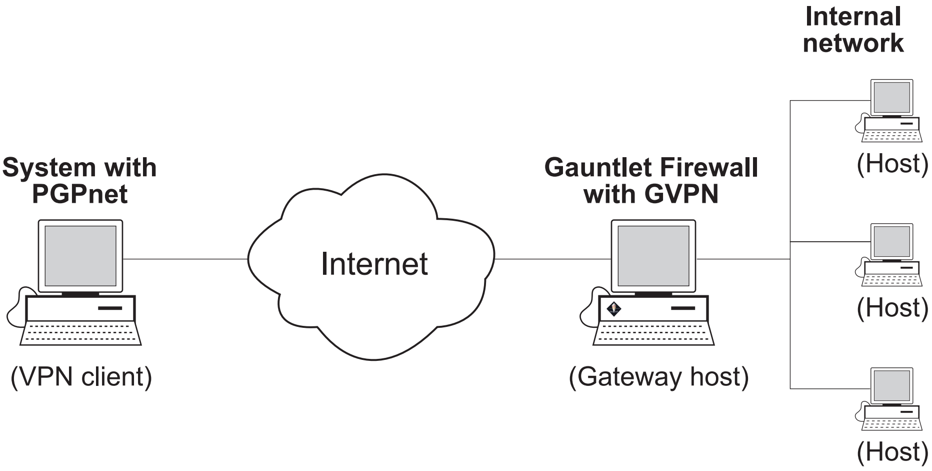
Creating a VPN with PGPnet 9

This chapter describes one way to use PGPnet to establish a VPN with a Gauntlet Firewall using its GVPN feature.

For the example in this chapter, we will be creating a trusted link between the two devices using IKE Client mode and certificate-based authentication. This type of VPN configuration is suitable for situations where a company employee is accessing the corporate network through a firewall over the Internet using an Internet Service Provider or if they get their IP address dynamically; via DHCP, for example.

Topology

The topology of such a VPN looks like this:



Some Firewall Terms

The following firewall terms are important to know when establishing a VPN with a Gauntlet Firewall:

- **Pre-shared secret and certificate-based authentication** — Gauntlet Firewalls support two methods of authentication: pre-shared secret, where the person or persons configuring the link use an agreed-upon passphrase to authenticate, and certificate-based authentication, where both devices in the link exchange certificates to authenticate.
- **Trusted and private links** — With a trusted link, data coming from the VPN client bypasses proxies on the firewall and goes directly to the intended destination. You are bypassing the security features of the firewall, so only do this if the VPN client is completely trusted; that is, a member of your organization. A private link does not bypass the proxies, which means that the VPN client must authenticate to the firewall in order to gain access to the intended destination.
- **Internal and external interfaces** — Firewalls have two physical interfaces: one goes to the Internet (the outside world), the other goes to the internal network. Each have their own IP addresses. The interface that connects to the Internet is called the outside or external interface; the interface that goes to the internal network is called the inside or internal interface. In most cases, the firewall protects the internal network from what's coming in on the external interface.
- **IKE Client and IPSec with IKE modes** — Gauntlet Firewalls support two connection modes: IKE Client, which works with certificate-based authentication only but supports VPN clients who acquire their IP addresses using DHCP (that is, they don't have fixed IP addresses but receive a different IP address each time they log on); and IPSec with IKE, which supports certificate-based or pre-shared secret authentication but requires that all hosts or subnets have fixed IP addresses (that is, it doesn't support DHCP).

  IKE Client mode is generally better suited for VPN client to VPN gateway configurations (PGPnet to firewall, for example), while IPSec with IKE mode is generally better suited for VPN gateway to VPN gateway configurations (firewall to firewall).

Establishing the VPN

To establish a VPN using IKE Client mode and certificate-based authentication between the system with PGPnet and the Gauntlet Firewall, you must:

- Set up certificate-based authentication
- Configure the Gauntlet Firewall
- Configure PGPnet
- Establish the VPN using PGPnet

All of these items are described in the following sections.

Setting up certificate-based authentication

The first step in establishing the VPN is to configure both devices to use certificate-based authentication. Valid certificates are needed to establish trust between the two devices in the VPN.

To obtain valid X.509 certificates for the Gauntlet Firewall, refer to the Gauntlet Firewall Global Virtual Private Network User’s Guide for Windows NT or UNIX (depending on which version of the Gauntlet Firewall you are using) for instructions. The documents came in hardcopy format with the firewall; they are also on the installation CD as PDF files.

To obtain a valid X.509 certificate for PGPnet (the VPN client), you will need to retrieve the Root CA certificate from the Certification Authority (CA) that both devices in the VPN trust (in this case, your company’s CA) and add it to your keyring, request a certificate for PGPnet from the CA, and then retrieve the certificate for PGPnet once it has been issued. All of these functions are done using PGPkeys.

To obtain a valid X.509 certificate for PGPnet (the VPN client):

1. Open your Web browser and connect to the CA enrollment site.

   For example, if your company uses the Net Tools PKI Server as the Certificate Authority, the URL will be similar to this format: **https://10.0.1.54**

   If you don’t know the URL for this site, contact your PGP or PKI Administrator.

2. Locate and examine the Root CA certificate.

For example, if your company were using the Net Tools PKI Server, you would click the “Download a CA Certificate” link and then examine the Root CA certificate.

3. Copy the key block (including the “-- Designated Cert --” and certificate extensions) for the Root CA certificate and paste it into your PGPkeys window.

   The Import Key dialog box appears and imports the Root CA certificate into your keyring.

4. Sign the Root CA certificate with your key to make it valid.

   You may also want to make the Root CA certificate a meta-introducer so you automatically trust certificates signed by it.

5. View its Properties and set its trust to **Trusted**.

6. Select **Options** from PGPkeys Edit menu, then click on the **CA** tab.

   The CA tab displays.

7. Enter the URL for the Root CA in the **Certificate Authority URL** text box.

   This is the same URL you used in Step 1.

   If there is a separate URL for the Revocation CA, enter it in the corresponding text box. If you do not know the URL for the Revocation CA, leave this field blank or consult your company’s PGP or PKI administrator.

8. In the **Type** box, select the type of PKI Server your company is using: Net Tools PKI Server, VeriSign OnSite, or Entrust.

9. Click the **Select Certificate** button, then select the Root CA certificate.

10. Click **OK**.

11. On the PGPkeys screen, select your key pair (or private key), pull down the Keys menu, select **Add**, then slide over and select **Certificate**.

    The Certificate Attributes dialog box appears.

12. Verify the certificate attributes; use the Add, Edit, and Remove buttons to make any required changes.

13. Click **OK**.

    The PGP Enter Passphrase dialog box displays.

14. Enter the passphrase for your keypair, then click **OK**.

    The certificate request is sent to the CA server. The server authenticates itself to your computer and then accepts your request.

At this point, your company's PGP or PKI administrator verifies your information in the request. The identifying information and public key are assembled and then digitally signed with the CA's own certificate. The whole, signed package that results is your new certificate.

The administrator sends you an email message (using the email address supplied on your keypair) stating that your certificate is ready for retrieval.

15. To retrieve your certificate and add it to your keypair, open PGPkeys (if it's not already open) and select the PGP key for which you made the certificate request.

16. Pull down the Server menu and select **Retrieve Certificate**.

    PGP contacts the CA server and automatically retrieves your new X.509 certificate and adds it to your PGP key.

Configuring the Gauntlet Firewall

The next step in establishing a VPN between a system with PGPnet and a Gauntlet Firewall is to appropriately configure the firewall.

NOTE: This procedure assumes a working Gauntlet Firewall. Please refer to the documentation that came with the firewall for complete information.

This procedure applies to both the Gauntlet Firewall for Windows NT and the Gauntlet Firewall for UNIX. Significant differences between the two are called out in the text.

To configure a Gauntlet Firewall for a VPN:

1. Using the Gauntlet Firewall Manager, click on the VPN tab.

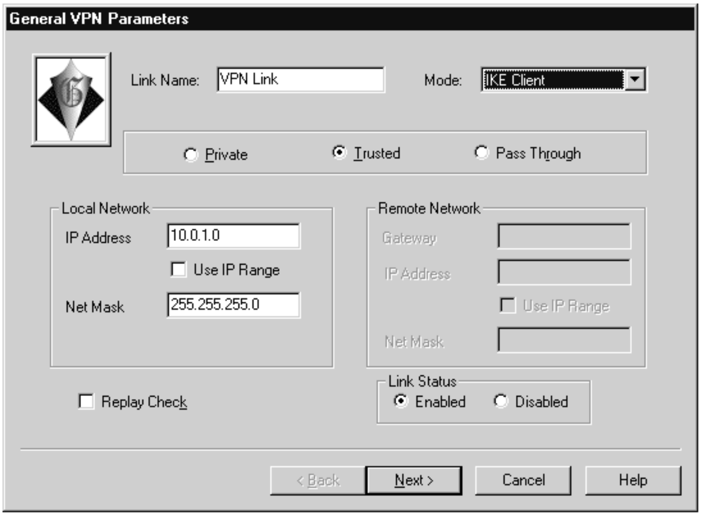
   The VPN screen displays.

   (On the Gauntlet Firewall for UNIX, select the **VPNs** folder and then click on **Links**.)

2. Click the **Add** button.

   The General VPN Parameters screen displays.

(On the Gauntlet Firewall for UNIX, this screen is called the Add GVPN Link Configuration screen.)



General VPN Parameters

Link Name: VPN Link

Mode: IKE Client

○ Private ● Trusted ○ Pass Through

Local Network

IP Address: 10.0.1.0

□ Use IP Range

Net Mask: 255.255.255.0

Remote Network

Gateway:

IP Address:

□ Use IP Range

Net Mask:

□ Replay Check

Link Status

● Enabled ○ Disabled

3. Add a VPN link with the following settings:

   Link Name: Enter a descriptive name

   Mode: IKE Client

   Link Type: Trusted

   IP Address: Enter the IP address of the host or subnet behind the firewall that will be participating in the VPN (generally you would be configuring a subnet here so that you aren't restricted to accessing just one computer)

   Use IP Range: Unchecked

   Net Mask: Enter the subnet mask of the subnet entered in the IP Address field or 255.255.255.255 if the IP address in the IP Address field is a host and not a subnet

> **NOTE:** The IP address and netmask information you enter here for the host or subnet you are configuring must also be entered in PGPnet.

   Replay Check: Unchecked

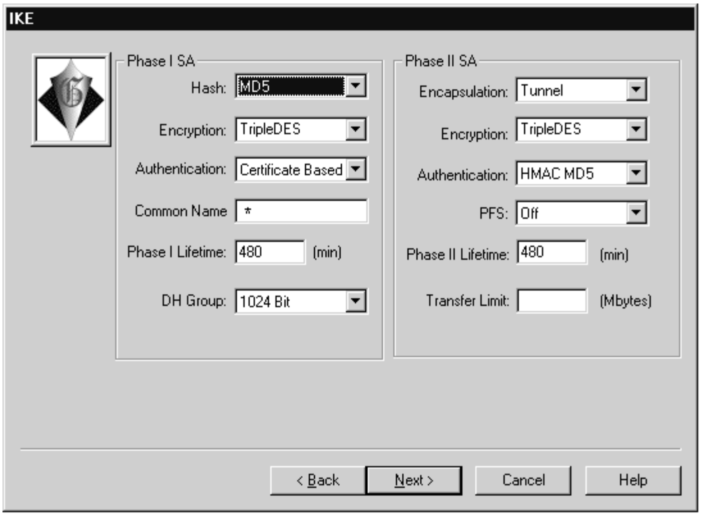
   Link Status: Enabled

4. Click **Next** to move to next screen.

   (On the Gauntlet Firewall for UNIX, click **Link Details**.)

The IKE screen displays.

(This screen is called Edit IKE Configuration on the Gauntlet Firewall for UNIX.)



IKE

Phase I SA

Hash: MD5

Encryption: TripleDES

Authentication: Certificate Based

Common Name *

Phase I Lifetime: 480 (min)

DH Group: 1024 Bit

Phase II SA

Encapsulation: Tunnel

Encryption: TripleDES

Authentication: HMAC MD5

PFS: Off

Phase II Lifetime: 480 (min)

Transfer Limit: (Mbytes)

Use the following settings:

Phase I SA

Hash: MD5

Encryption: TripleDES

Authentication: Certificate Based

Common Name: * (On the Gauntlet Firewall for UNIX, leave blank.)

Phase I Lifetime: 480

DH Group: 1024 Bit

Phase II SA

Encapsulation: Tunnel

Encryption: TripleDES

Authentication: HMAC MD5

PFS: Off

Phase II Lifetime: 480

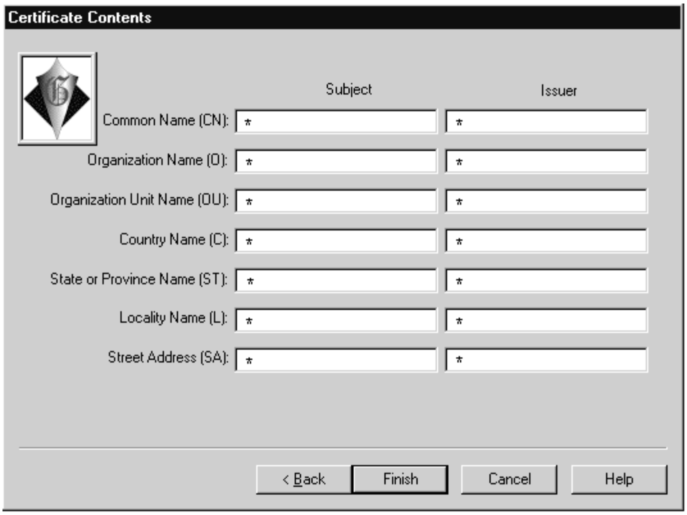
Transfer Limit: Leave blank

5. Click **Next** to move to next screen.

   (On the Gauntlet Firewall for UNIX, click the **Certificate Contents** button.)

   The Certificate Contents screen displays. Each field should have an asterisk (*).

   (This screen is called Client Certificate Configuration on the Gauntlet Firewall for UNIX. All fields should be blank; do not enter asterisks.)



	Subject	Issuer
Common Name (CN):	*	*
Organization Name (O):	*	*
Organization Unit Name (OU):	*	*
Country Name (C):	*	*
State or Province Name (ST):	*	*
Locality Name (L):	*	*
Street Address (SA):	*	*

6. Click **Finish**, and apply the changes to the firewall.

   (On the Gauntlet Firewall for UNIX, click **OK** and then apply the changes to the firewall.)

Configuring PGPnet

The next step in establishing a VPN between a system with PGPnet and a Gauntlet Firewall is to appropriately configure PGPnet.

NOTE: This procedure assumes a working installation of PGP 6.5 or greater with the PGPnet component installed.

In this example we will be configuring communications to an insecure host or subnet behind a secure gateway.

NOTE: Communication with a secure host behind a secure gateway requires Version 5.0 or greater of the Gauntlet Firewall for UNIX or Version 5.5 or greater of the Gauntlet Firewall for Windows NT.

To configure PGPnet for the VPN using the Host/Gateway screen:

1. Open PGPnet and select the **Hosts** tab.
2. On the Hosts tab, click the **Add** button.
3. On the Host/Gateway screen, enter a descriptive name for the gateway in the **Name** text box.

   Because you want to communicate with a host behind a firewall, you must configure the gateway host (the firewall) first, then the internal host behind the firewall.
4. Enter the IP address of the gateway (this is the IP address of the firewall's *external* interface).
5. Select **Secure Gateway** from the drop-down list.
6. Click **OK** to add the gateway entry to PGPnet's host list.

   The Hosts tab displays.
7. On the Hosts tab, select the secure gateway configuration you just created and click the **Add** button again.
8. On the Host/Gateway screen, enter a descriptive name for the host or subnet behind the firewall that you want to communicate with.
9. Enter the host or subnet's IP address.

   This IP address must be the same as the IP address you entered in Step 3 of the procedure to configure the Gauntlet Firewall.

10. Select **Insecure Host** or **Insecure Subnet** as appropriate from the drop-down list.

    If you select **Insecure Subnet**, you must specify a subnet mask.

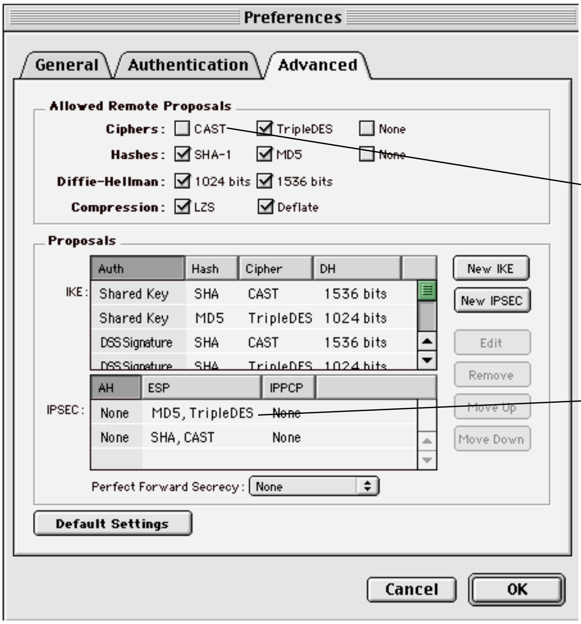
11. Click **OK** to add the host or subnet to PGPnet's host list.

    The Hosts tab displays.

12. Use the default PGPnet settings unless you want to set SA expiration values (found on the General tab of the Preferences screen).

IMPORTANT: If you are establishing a VPN with a host or subnet behind a Gauntlet Firewall for Windows NT version 5.0, you must deselect CAST from the list of Allowed Remote Proposals. To do this, on the PGPnet menu bar, pull down the Edit menu and select Preferences. Click the Advanced tab, and then uncheck the CAST option.

If you are establishing a VPN with a host or subnet behind a Gauntlet Firewall for UNIX version 5.0, you must move the IPSec proposal being used (in this example, MD5, Triple DES) to the top of the list of IPSec proposals. To do this: in PGPnet, pull down the Edit menu and select Preferences; click the Advanced tab and find the IPSec Proposals; in the ESP column, click on the MD5, Triple DES listing, then click the Move Up button until MD5, Triple DES is at the top of the list; click OK.



Allowed Remote Proposals

Ciphers: □ CAST ☑ TripleDES □ None

Hashes: ☑ SHA-1 ☑ MD5 □ None

Diffie-Hellman: ☑ 1024 bits ☑ 1536 bits

Compression: ☑ LZS ☑ Deflate

Proposals

IKE:

Auth	Hash	Cipher	DH
Shared Key	SHA	CAST	1536 bits
Shared Key	MD5	TripleDES	1024 bits
DSS Signature	SHA	CAST	1536 bits
DSS Signature	SHA	TripleDES	1024 bits

IPSEC:

AH	ESP	IPPCP
None	MD5, TripleDES	None
None	SHA, CAST	None

Perfect Forward Secrecy: None

For a VPN with a Gauntlet Firewall for Windows NT 5.0, CAST must be de-selected.

For a VPN with a Gauntlet Firewall for UNIX 5.0, the IPSEC proposal being used (in this example, MD5, Triple DES) must be at the top of the list of IPSEC proposals.

Establishing the VPN using PGPnet

The final step in establishing a VPN between a system with PGPnet and a Gauntlet Firewall is to actually establish the VPN (called a Security Association in PGPnet terminology) using PGPnet.

To establish the VPN using PGPnet:

1. Open PGPnet and click the **Hosts** tab.
2. Click on the name of the gateway host (the firewall) you configured.
3. If your X.509 certificate has already been set as your authenticating key, skip to Step 10. If you have not set your X.509 certificate as your authenticating key or you are not sure, continue with Step 4.
4. Pull down the View menu and select **Options**.

   The Options screen displays.
5. Click the **Authentication** tab.
6. On the Authentication tab, click **Select Certificate**.

   A list of X.509 certificates that are on your keyring displays.
7. Click on the name of the certificate you would like to use to authenticate yourself and click **OK**.
8. Click **OK** again to close the Options screen.

   A dialog box prompts you for the passphrase for the selected key.
9. Enter the passphrase for the key and click **OK**.

   The Hosts screen displays.
10. Click the plus sign next to the gateway host (the firewall) you configured.

    A list of host entries (hosts or subnets behind the gateway) displays.
11. To start communications with an insecure host or subnet, click on the host entry you want to connect to, then click **Connect**.

    If everything is configured correctly, the IPSec protocols establish a Security Association between the VPN client (PGPnet) and the Gauntlet Firewall.

    When the Security Association is created, a green dot displays to the right of the gateway host in the SA column.

12. Click the **Status** tab.

    The Security Association is listed.

13. If the Security Association is not listed, click the **Log** tab to see what the problem was.

NOTE: Refer to the PGPnet chapter for more information about establishing a Security Association, log entry error descriptions, and detailed PGPnet configuration information.

Troubleshooting PGP A

This appendix presents information about problems you may encounter while using PGP and suggests solutions.

Error	Cause	Solution
Administrative preferences file not found	The preference file containing the configuration set up by your PGP administrator, usually IS/IT personnel, is missing.	Re-install PGP onto your machine. If the message continues to appear after re-installing, contact your PGP administrator and report this message. They will need to generate a new PGP installer for you.
Authentication rejected by remote SKEP connection	The user on the remote side of the network share file connection rejected the key that you provided for authentication.	Use a different key to authenticate the network share file connection, or contact the remote user to assure them that the key you're using is valid.
Cannot perform the requested operation because the output buffer is too small.	The output is larger than the internal buffers can handle.	If you are encrypting or signing, you may have to break up the message and encrypt/sign smaller pieces at a time. If you are decrypting or verifying, ask the sender to encrypt/sign smaller pieces and re-send them to you.
Could not encrypt to specified key because it is a sign-only key.	The selected key can only be used for signing.	Choose a different key, or generate a new key that can encrypt data.
Could not sign with specified key because it is an encrypt-only key.	The selected key can only be used for encrypting.	Choose a different key, or generate a new key that can sign data.
Error in domain name systemic	The destination address you provided is incorrect, or your network connection is misconfigured.	Check to make sure that the destination address you provided is the correct one. If you are sure of this, check your connection to the network.
Identical shares cannot be combined	You attempted to combine the same share twice.	If you received the shares from a share file, try choosing a different share file. If you received the shares from the network, you may need to contact the user at the remote location and tell them to send a different set of shares

Error	Cause	Solution
No secret keys could be found on your keyring.	There are no private keys on your keyring.	Generate your own pair of keys in PGPkeys.
Socket is not connected	The network connection to the PGP cert server or to the network share file connection has been broken.	Try re-establishing the connection by repeating the procedure you used to start the connection. If that fails, check your connection to the network.
The action could not be completed due to an invalid file operation.	The program failed to read or write data in a certain file.	The file is probably corrupt. Try altering your PGP Preferences to use a different file, if possible.
The evaluation time for PGP encrypting and signing has passed. Operation aborted.	The product evaluation time has expired.	Download the freeware version or buy the commercial version of the product.
The keyring contains a bad (corrupted) PGP packet.	The PGP message that you are working with has been corrupted, or your keyring has been corrupted.	Ask the sender to re-send the message if it's a message that you're working with. If it's your keyring, try restoring from your backup keyring.
The keyring file is corrupt.	The program failed to read or write data in a certain file.	There is a file that is probably corrupt or missing. It may or may not be the keyring file. Try using a different file name or path, if possible.
The message/data contains a detached signature.	The signature for the message/file is located in a separate file.	Double-click on the detached signature file first.
The passphrase you entered does not match the passphrase on the key.	The passphrase you entered is incorrect.	You may have the CAPS LOCK on, or you simply may have mis-typed the passphrase. Try again.
The PGP library has run out of memory.	The operating system has run out of memory.	Close other running programs. If that doesn't work, you may need more memory in your machine.
The specified user ID was not added because it already exists on the selected key.	You can't add a User ID to a key if there is one just like it already on the key.	Try adding a different user ID, or delete the matching one first.

Error	Cause	Solution
The specified key could not be found on your keyring.	The key needed to decrypt the current message is not on your keyring.	Ask the sender of the message to re-send the message and make sure they encrypt the message to your public key.
The specified input file does not exist.	The file name typed in does not exist.	Browse to find the exact name and path of the file you want.
There is not enough random data currently available.	The random number generator needs more input in order to generate good random numbers.	When prompted, move the mouse around, or press random keys, in order to generate input.
There was an error during the writing of the keyring or the exported file.	The program failed to write data to a certain file.	Your hard drive may be full, or if the file is on a floppy, the floppy is not present in the floppy drive.
There was an error opening or writing the keyring or the output file.	A file that was needed couldn’t be opened.	Make sure the settings in your PGP Preferences is correct. If you’ve recently deleted files in the directory that you installed PGP, you may need to re-install the product.
This key is already signed by the specified signing key.	You can’t sign a key that you have already signed.	You may have accidentally picked the wrong key. If so, choose a different key to sign.
Unable to perform operation because this file is read-only or otherwise protected. If you store your keyring files on removable media the media may not be inserted.	A file that was needed is set to read-only or is being used by another program.	Close other programs that may be accessing the same files as the program you are running. If you keep your keyring files on a floppy disk, make sure that the floppy disk is in the floppy drive.

Transferring Files Between the Mac OS and Windows

B

Transferring files to and from Mac OS is a classic problem in using almost any kind of data exchange software, such as email applications, FTP, compression utilities, and PGP. This appendix is intended to document how this problem is finally solved by PGP version 5.5.x, and to discuss how to communicate with previous versions of PGP.

The Mac OS stores files differently from other operating systems. Even the text file format of the Mac OS is different. Mac OS files are really two files consisting of a Data segment and a Resource segment. In order to send a file from Mac OS to Windows without losing data, the two segments must be merged into one. The standard method by which a Mac OS file is converted into a single file so that it can be transferred to another Macintosh or PC without losing either of its halves is called MacBinary.

The problem is that, without special software, Windows and other platforms cannot inherently understand the MacBinary format. If a situation occurs where the receiving software fails to convert a MacBinary format file into a Windows file, the resulting file is unusable. Third-party utilities exist on Windows to convert it after the fact into a usable file, but that can be rather inconvenient.

Previous versions of PGP and most utilities available on the market today generally try to ignore this problem as much as possible and leave all decisions up to the user as to whether or not to encode a file with MacBinary when sending from Mac OS. This places the burden of deciding to send with MacBinary, and not risk losing any data, or send without MacBinary, with hope that no important data will be lost on the user, who often has no idea what the correct decision is. The decision should generally be based on whether the file is being sent to Windows or Mac OS. But what about if you're sending to both at the same time? There is no good solution to that problem with older versions of PGP and many other utilities. This has resulted in great confusion and inconvenience for users.

The reverse, sending a file from Windows to the Mac OS, has also been a major problem. Windows uses filename extensions, such as .doc, to identify the type of a file. This is meaningless to the Mac OS. These files are sent to a Macintosh computer without any file type or creator information. The process of making them readable after receipt generally involves various arcane motions in the Open dialog of the creator application, or in many cases requires the user to understand Mac OS lore of creator and type codes by setting them manually in a third-party utility.

Fortunately, the latest version of PGP (versions 5.5 through 6.5) leads the way out of this confusion. If all PGP users were to use the latest versions, no one would have to think about how to send files from Mac OS to Windows and vice versa.

Sending from the Mac OS to Windows

On the Mac OS, there are three options when encrypting or signing a file:

- **MacBinary: Yes.** This is the recommended option for all encryptions when sending to another user of PGP Version 5.5 or above on any platform. This means that Mac OS users will receive the exact file that was intended, and the Windows version will automatically decode the MacBinary and even append the appropriate file extension, such as .doc for Microsoft Word or .ppt for Microsoft PowerPoint. PGP includes information on most popular application filename extensions and Macintosh-creator codes. In cases where the type is unknown or known to be a Mac OS-only file such as a Mac OS application, the file remains in MacBinary format so that it can later be forwarded to a Macintosh fully intact.

- **MacBinary: No.** If you are communicating with users who have an older version of PGP, the decision of whether to send with MacBinary generally ends up in the sender's hands as in most other programs and in previous versions of PGP for Mac OS. When sending to a PC using an older version, if you know that the file you are sending can be read by Windows applications when no MacBinary is used, select this option. This includes most files that are generally cross-platform such as those created by the Microsoft Office applications, graphics files, compressed files, and many others. The sender or the recipient will have to manually rename the file to have the correct filename extension on Windows. This is required because the Windows recipient does not have the creator information normally encoded with MacBinary.

- **MacBinary: Smart.** There are some very limited cases where this option can be useful when communicating with users who are not using later versions of PGP. This option makes a decision as to whether to encode with MacBinary based on an analysis of the actual data in the file. If the file is one of the following types, it will not be encoded with MacBinary, thereby making it readable on a PC with any version of PGP:

  – PKzip compressed file

  – Lempel-Ziv compressed file

  – MIDI music format file

  – PackIt compressed file

- GIF graphics file
- StuffIt compressed file
- Compactor compressed file
- Arc compressed file
- JPEG graphics file

As shown, only a limited selection of files will result in a readable file by old versions of PGP on other platforms using the Smart option. Any other file received on a PC with an older version of PGP will be unreadable without stripping the MacBinary encoding with a third-party utility. Also, the file will not have the correct filename extension on the PC unless that extension was manually added by the user on the sending side. Using Smart mode, the resulting file may not be the same as the original when sent to a Macintosh, because it may lose its creator and type codes. This mode remains in the product mostly due to the fact that it was in PGP Version 5.0 and some users may only have a need to send the above file types. This option is not recommended in most cases.

In summary, if you are sending only to versions 6.x, always select MacBinary: Yes (the default). Thus, no thought is required if your environment is using PGP version 6.x exclusively. When sending to users with older versions, you should select MacBinary: No for cross-platform file types and MacBinary: Yes for files which simply wouldn't be readable to PC users anyway (such as a Mac OS application).

NOTE: PGP Version 5.0 did not have a MacBinary: No option. In order to send file types without MacBinary, which are not included in the MacBinary: Smart list to a PC using 5.0, the file must be manually set to one of the creator and type codes on the Smart list before sending.

Receiving Windows files on the Mac OS

When decrypting, PGP version 5.5.x and later automatically attempts to translate filename extensions for non-MacBinary files into Mac OS creator and type information. For example, if you receive a file from Windows with an extension of .doc, the file will be saved as a Microsoft Word document. The same list of applications used when adding filename extensions upon receipt of a MacBinary file on Windows is used to translate filename extensions back into the Mac OS equivalent when received on a Macintosh computer. In almost all cases, this results in files which are immediately readable and double-clickable on Mac OS.

Previous versions of PGP for Mac OS do not have this feature. The user will have to manually determine that a file named “report.doc” is a Microsoft Word file. After determining the creator application, in the case of Microsoft Word, one can simply use the Open dialog to open the file by selecting Show All Files from the popup menu. Many other applications also have this feature, but some don't. If the document cannot be opened from within the application, the user must find out what the appropriate Macintosh creator and type codes are for the file and manually set them with a third-party utility. There are many free utilities to do this. Upgrading to version 6.x is probably the easiest option in this case, as it eliminates this problem.

Supported Applications

The following list of major applications produce documents which are automatically translated by PGP when sent from Windows to Mac OS and vice versa. You can add items to this list by editing the PGPMacBinaryMappings.txt file in the \WINDOWS directory. On the Mac side, remove the .txt suffix on the filename—PGPMacBinaryMappings is located in System Folder / Preferences / Pretty Good Preferences.

- PhotoShop (GIF, native Photoshop documents, TGA, JPEG)
- PageMaker (Versions 3.X, 4.X, 5.X, 6.X)
- Microsoft Project (project and template files)
- FileMaker Pro
- Adobe Acrobat
- Lotus 123
- Microsoft Word (text, RTF, templates)
- PGP
- Microsoft PowerPoint
- StuffIt
- QuickTime
- Corel WordPerfect
- Microsoft Excel (many different types of files)
- Quark XPress

The following general filename extensions are also converted:

.cvs
.dl
.msp
.plt
.spc
.gz
.mpg
.mod
.pit
.arj
.fli
.pac
.pm
.sr
.vga
.dvi
.svx
.txt
.ima
.ico
.pbm
.ppm
.sun
.hal
.tex
.wav
.mdi
.eps
.iff
.pcs
.rif
.sup
.lzh
.aif
.tar
.pak
.mac
.img
.pcx
.rle
.wmf
.Z
.zip
.pct
.tif
.cgm
.lbm
.pgm
.shp
.flc
.exe
.au
.pic
.eps

Phil Zimmermann on PGP C

This chapter contains introductory and background information about cryptography and PGP as written by Phil Zimmermann.

Why I wrote PGP

"Whatever you do will be insignificant, but it is very important that you do it."
—Mahatma Gandhi.

It's personal. It's private. And it's no one's business but yours. You may be planning a political campaign, discussing your taxes, or having a secret romance. Or you may be communicating with a political dissident in a repressive country. Whatever it is, you don't want your private electronic mail (email) or confidential documents read by anyone else. There's nothing wrong with asserting your privacy. Privacy is as apple-pie as the Constitution.

The right to privacy is spread implicitly throughout the Bill of Rights. But when the United States Constitution was framed, the Founding Fathers saw no need to explicitly spell out the right to a private conversation. That would have been silly. Two hundred years ago, all conversations were private. If someone else was within earshot, you could just go out behind the barn and have your conversation there. No one could listen in without your knowledge. The right to a private conversation was a natural right, not just in a philosophical sense, but in a law-of-physics sense, given the technology of the time.

But with the coming of the information age, starting with the invention of the telephone, all that has changed. Now most of our conversations are conducted electronically. This allows our most intimate conversations to be exposed without our knowledge. Cellular phone calls may be monitored by anyone with a radio. Electronic mail, sent across the Internet, is no more secure than cellular phone calls. Email is rapidly replacing postal mail, becoming the norm for everyone, not the novelty it was in the past. And email can be routinely and automatically scanned for interesting keywords, on a large scale, without detection. This is like driftnet fishing.

Perhaps you think your email is legitimate enough that encryption is unwarranted. If you really are a law-abiding citizen with nothing to hide, then why don’t you always send your paper mail on postcards? Why not submit to drug testing on demand? Why require a warrant for police searches of your house? Are you trying to hide something? If you hide your mail inside envelopes, does that mean you must be a subversive or a drug dealer, or maybe a paranoid nut? Do law-abiding citizens have any need to encrypt their email?

What if everyone believed that law-abiding citizens should use postcards for their mail? If a nonconformist tried to assert his privacy by using an envelope for his mail, it would draw suspicion. Perhaps the authorities would open his mail to see what he’s hiding. Fortunately, we don’t live in that kind of world, because everyone protects most of their mail with envelopes. So no one draws suspicion by asserting their privacy with an envelope. There’s safety in numbers. Analogously, it would be nice if everyone routinely used encryption for all their email, innocent or not, so that no one drew suspicion by asserting their email privacy with encryption. Think of it as a form of solidarity.

Until now, if the government wanted to violate the privacy of ordinary citizens, they had to expend a certain amount of expense and labor to intercept and steam open and read paper mail. Or they had to listen to and possibly transcribe spoken telephone conversation, at least before automatic voice recognition technology became available. This kind of labor-intensive monitoring was not practical on a large scale. It was only done in important cases when it seemed worthwhile.

Senate Bill 266, a 1991 omnibus anticrime bill, had an unsettling measure buried in it. If this non-binding resolution had become real law, it would have forced manufacturers of secure communications equipment to insert special “trap doors” in their products, so that the government could read anyone’s encrypted messages. It reads, “It is the sense of Congress that providers of electronic communications services and manufacturers of electronic communications service equipment shall ensure that communications systems permit the government to obtain the plain text contents of voice, data, and other communications when appropriately authorized by law.” It was this bill that led me to publish PGP electronically for free that year, shortly before the measure was defeated after vigorous protest by civil libertarians and industry groups.

The 1994 Digital Telephony bill mandated that phone companies install remote wiretapping ports into their central office digital switches, creating a new technology infrastructure for “point-and-click” wiretapping, so that federal agents no longer have to go out and attach alligator clips to phone lines. Now they will be able to sit in their headquarters in Washington and listen in on your phone calls. Of course, the law still requires a court order for a wiretap. But while technology infrastructures can persist for generations, laws and policies can change overnight. Once a communications infrastructure optimized for surveillance becomes entrenched, a shift in political conditions may lead to abuse of this new-found power. Political conditions may shift with the election of a new government, or perhaps more abruptly from the bombing of a federal building.

A year after the 1994 Digital Telephony bill passed, the FBI disclosed plans to require the phone companies to build into their infrastructure the capacity to simultaneously wiretap 1 percent of all phone calls in all major U.S. cities. This would represent more than a thousandfold increase over previous levels in the number of phones that could be wiretapped. In previous years, there were only about a thousand court-ordered wiretaps in the United States per year, at the federal, state, and local levels combined. It’s hard to see how the government could even employ enough judges to sign enough wiretap orders to wiretap 1 percent of all our phone calls, much less hire enough federal agents to sit and listen to all that traffic in real time. The only plausible way of processing that amount of traffic is a massive Orwellian application of automated voice recognition technology to sift through it all, searching for interesting keywords or searching for a particular speaker’s voice. If the government doesn’t find the target in the first 1 percent sample, the wiretaps can be shifted over to a different 1 percent until the target is found, or until everyone’s phone line has been checked for subversive traffic. The FBI says they need this capacity to plan for the future. This plan sparked such outrage that it was defeated in Congress, at least this time around, in 1995. But the mere fact that the FBI even asked for these broad powers is revealing of their agenda. And the defeat of this plan isn’t so reassuring when you consider that the 1994 Digital Telephony bill was also defeated the first time it was introduced, in 1993.

Advances in technology will not permit the maintenance of the status quo, as far as privacy is concerned. The status quo is unstable. If we do nothing, new technologies will give the government new automatic surveillance capabilities that Stalin could never have dreamed of. The only way to hold the line on privacy in the information age is strong cryptography.

You don’t have to distrust the government to want to use cryptography. Your business can be wiretapped by business rivals, organized crime, or foreign governments. Several foreign governments, for example, admit to using their signals intelligence against companies from other countries to give their own corporations a competitive edge. Ironically, the United States government’s restrictions on cryptography have weakened U.S. corporate defenses against foreign intelligence and organized crime.

The government knows what a pivotal role cryptography is destined to play in the power relationship with its people. In April 1993, the Clinton administration unveiled a bold new encryption policy initiative, which had been under development at the National Security Agency (NSA) since the start of the Bush administration. The centerpiece of this initiative was a government-built encryption device, called the Clipper chip, containing a new classified NSA encryption algorithm. The government tried to encourage private industry to design it into all their secure communication products, such as secure phones, secure faxes, and so on. AT&T put Clipper into its secure voice products. The catch: At the time of manufacture, each Clipper chip is loaded with its own unique key, and the government gets to keep a copy, placed in escrow. Not to worry, though—the government promises that they will use these keys to read your traffic only “when duly authorized by law.” Of course, to make Clipper completely effective, the next logical step would be to outlaw other forms of cryptography.

The government initially claimed that using Clipper would be voluntary, that no one would be forced to use it instead of other types of cryptography. But the public reaction against the Clipper chip has been strong, stronger than the government anticipated. The computer industry has monolithically proclaimed its opposition to using Clipper. FBI director Louis Freeh responded to a question in a press conference in 1994 by saying that if Clipper failed to gain public support, and FBI wiretaps were shut out by non-government-controlled cryptography, his office would have no choice but to seek legislative relief. Later, in the aftermath of the Oklahoma City tragedy, Mr. Freeh testified before the Senate Judiciary Committee that public availability of strong cryptography must be curtailed by the government (although no one had suggested that cryptography was used by the bombers).

The Electronic Privacy Information Center (EPIC) obtained some revealing documents under the Freedom of Information Act. In a briefing document titled “Encryption: The Threat, Applications and Potential Solutions,” and sent to the National Security Council in February 1993, the FBI, NSA, and Department of Justice (DOJ) concluded that “Technical solutions, such as they are, will only work if they are incorporated into all encryption products. To ensure that this occurs, legislation mandating the use of Government-approved encryption products or adherence to Government encryption criteria is required.”

The government has a track record that does not inspire confidence that they will never abuse our civil liberties. The FBI's COINTELPRO program targeted groups that opposed government policies. They spied on the antiwar movement and the civil rights movement. They wiretapped the phone of Martin Luther King Jr. Nixon had his enemies list. And then there was the Watergate mess. Congress now seems intent on passing laws curtailing our civil liberties on the Internet. At no time in the past century has public distrust of the government been so broadly distributed across the political spectrum, as it is today.

If we want to resist this unsettling trend in the government to outlaw cryptography, one measure we can apply is to use cryptography as much as we can now while it's still legal. When use of strong cryptography becomes popular, it's harder for the government to criminalize it. Therefore, using PGP is good for preserving democracy.

If privacy is outlawed, only outlaws will have privacy. Intelligence agencies have access to good cryptographic technology. So do the big arms and drug traffickers. But ordinary people and grassroots political organizations mostly have not had access to affordable "military grade" public-key cryptographic technology. Until now.

PGP empowers people to take their privacy into their own hands. There's a growing social need for it. That's why I created it.

The PGP symmetric algorithms

PGP offers a selection of different secret key algorithms to encrypt the actual message. By secret key algorithm, we mean a conventional, or symmetric, block cipher that uses the same key to both encrypt and decrypt. The three symmetric block ciphers offered by PGP are CAST, Triple-DES, and IDEA. They are not "home-grown" algorithms. They were all developed by teams of cryptographers with distinguished reputations.

For the cryptographically curious, all three ciphers operate on 64-bit blocks of plaintext and ciphertext. CAST and IDEA have key sizes of 128 bits, while Triple-DES uses a 168-bit key. Like Data Encryption Standard (DES), any of these ciphers can be used in cipher feedback (CFB) and cipher block chaining (CBC) modes. PGP uses them in 64-bit CFB mode.

I included the CAST encryption algorithm in PGP because it shows promise as a good block cipher with a 128-bit key size, it's very fast, and it's free. Its name is derived from the initials of its designers, Carlisle Adams and Stafford Tavares of Northern Telecom (Nortel). Nortel has applied for a patent for CAST, but they have made a commitment in writing to make CAST available to anyone on a royalty-free basis. CAST appears to be exceptionally well designed, by people with good reputations in the field. The design is based on a very formal approach, with a number of formally provable assertions that give good reasons to believe that it probably requires key exhaustion to break its 128-bit key. CAST has no weak or semiweak keys. There are strong arguments that CAST is completely immune to both linear and differential cryptanalysis, the two most powerful forms of cryptanalysis in the published literature, both of which have been effective in cracking DES. CAST is too new to have developed a long track record, but its formal design and the good reputations of its designers will undoubtedly attract the attentions and attempted cryptanalytic attacks of the rest of the academic cryptographic community. I'm getting nearly the same preliminary gut feeling of confidence from CAST that I got years ago from IDEA, the cipher I selected for use in earlier versions of PGP. At that time, IDEA was also too new to have a track record, but it has held up well.

The IDEA (International Data Encryption Algorithm) block cipher is based on the design concept of "mixing operations from different algebraic groups." It was developed at ETH in Zurich by James L. Massey and Xuejia Lai, and published in 1990. Early published papers on the algorithm called it IPES (Improved Proposed Encryption Standard), but they later changed the name to IDEA. So far, IDEA has resisted attack much better than other ciphers such as FEAL, REDOC-II, LOKI, Snefru and Khafre. And IDEA is more resistant than DES to Biham and Shamir's highly successful differential cryptanalysis attack, as well as attacks from linear cryptanalysis. As this cipher continues to attract attack efforts from the most formidable quarters of the cryptanalytic world, confidence in IDEA is growing with the passage of time. Sadly, the biggest obstacle to IDEA's acceptance as a standard has been the fact that Ascom Systec holds a patent on its design, and unlike DES and CAST, IDEA has not been made available to everyone on a royalty-free basis.

As a hedge, PGP includes three-key Triple-DES in its repertoire of available block ciphers. The DES was developed by IBM in the mid-1970s. While it has a good design, its 56-bit key size is too small by today's standards. Triple-DES is very strong, and has been well studied for many years, so it might be a safer bet than the newer ciphers such as CAST and IDEA. Triple-DES is the DES applied three times to the same block of data, using three different keys, except that the second DES operation is run backwards, in decrypt mode. While Triple-DES is much slower than either CAST or IDEA, speed is usually not critical for email applications. Although Triple-DES uses a key size of 168 bits, it appears to have an effective key strength of at least 112 bits against an attacker with impossibly immense data storage capacity to use in the attack. According to a paper presented by Michael Weiner at Crypto96, any remotely plausible amount of data storage available to the attacker would enable an attack that would require about as much work as breaking a 129-bit key. Triple-DES is not encumbered by any patents.

PGP public keys that were generated by PGP Version 5.0 or later have information embedded in them that tells a sender what block ciphers are understood by the recipient's software, so that the sender's software knows which ciphers can be used to encrypt. Diffie-Hellman / DSS public keys accept CAST, IDEA, or Triple-DES as the block cipher, with CAST as the default selection. At present, for compatibility reasons, RSA keys do not provide this feature. Only the IDEA cipher is used by PGP to send messages to RSA keys, because older versions of PGP only supported RSA and IDEA.

About PGP data compression routines

PGP normally compresses the plaintext before encrypting it, because it's too late to compress the plaintext after it has been encrypted; encrypted data is not compressible. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Most cryptanalysis techniques exploit redundancies found in the plaintext to crack the cipher. Data compression reduces this redundancy in the plaintext, thereby greatly enhancing resistance to cryptanalysis. It takes extra time to compress the plaintext, but from a security point of view it's worth it.

Files that are too short to compress, or that just don't compress well, are not compressed by PGP. In addition, the program recognizes files produced by most popular compression programs, such as PKZIP, and does not try to compress a file that has already been compressed.

For the technically curious, the program uses the freeware ZIP compression routines written by Jean-Loup Gailly, Mark Adler, and Richard B. Wales. This ZIP software uses compression algorithms that are functionally equivalent to those used by PKWare's PKZIP 2.x. This ZIP compression software was selected for PGP mainly because it has a really good compression ratio and because it's fast.

About the random numbers used as session keys

PGP uses a cryptographically strong pseudo-random-number generator for creating temporary session keys. If this random seed file does not exist, it is automatically created and seeded with truly random numbers derived from your random events gathered by the PGP program from the timing of your keystroke and mouse movements.

This generator reseeds the seed file each time it is used, by mixing in new material partially derived from the time of day and other truly random sources. It uses the conventional encryption algorithm as an engine for the random number generator. The seed file contains both random seed material and random key material used to key the conventional encryption engine for the random generator.

This random seed file should be protected from disclosure, to reduce the risk of an attacker deriving your next or previous session keys. The attacker would have a very hard time getting anything useful from capturing this random seed file, because the file is cryptographically laundered before and after each use. Nonetheless, it seems prudent to try to keep it from falling into the wrong hands. If possible, make the file readable only by you. If this is not possible, don't let other people indiscriminately copy disks from your computer.

About the message digest

The message digest is a compact (160-bit or 128-bit) "distillate" of your message or file checksum. You can also think of it as a "fingerprint" of the message or file. The message digest "represents" your message, in such a way that if the message were altered in any way, a different message digest would be computed from it. This makes it possible to detect any changes made to the message by a forger. A message digest is computed using a cryptographically strong one-way hash function of the message. It should be computationally infeasible for an attacker to devise a substitute message that would produce an identical message digest. In that respect, a message digest is much better than a checksum, because it is easy to devise a different message that would produce the same checksum. But like a checksum, you can't derive the original message from its message digest.

The message digest algorithm now used in PGP (Version 5.0 and later) is called SHA, which stands for Secure Hash Algorithm, designed by the NSA for the National Institute of Standards and Technology (NIST). SHA is a 160-bit hash algorithm. Some people might regard anything from the NSA with suspicion, because the NSA is in charge of intercepting communications and breaking codes. But keep in mind that the NSA has no interest in forging signatures, and the government would benefit from a good unforgeable digital signature standard that would preclude anyone from repudiating their signatures. That has distinct benefits for law enforcement and intelligence gathering. Also, SHA has been published in the open literature and has been extensively peer-reviewed by most of the best cryptographers in the world who specialize in hash functions, and the unanimous opinion is that SHA is extremely well designed. It has some design innovations that overcome all the observed weaknesses in message digest algorithms previously published by academic cryptographers. All new versions of PGP use SHA as the message digest algorithm for creating signatures with the new DSS keys that comply with the NIST Digital Signature Standard. For compatibility reasons, new versions of PGP still use MD5 for RSA signatures, because older versions of PGP used MD5 for RSA signatures.

The message digest algorithm used by older versions of PGP is the MD5 Message Digest Algorithm, placed in the public domain by RSA Data Security, Inc. MD5 is a 128-bit hash algorithm. In 1996, MD5 was all but broken by a German cryptographer, Hans Dobbertin. Although MD5 was not completely broken at that time, it was discovered to have such serious weaknesses that no one should keep using it to generate signatures. Further work in this area might completely break it, allowing signatures to be forged. If you don't want to someday find your PGP digital signature on a forged confession, you might be well advised to migrate to the new PGP DSS keys as your preferred method for making digital signatures, because DSS uses SHA as its secure hash algorithm.

How to protect public keys from tampering

In a public key cryptosystem, you don't have to protect public keys from exposure. In fact, it's better if they are widely disseminated. But it's important to protect public keys from tampering, to make sure that a public key really belongs to the person to whom it appears to belong. This may be the most important vulnerability of a public key cryptosystem. Let's first look at a potential disaster, then describe how to safely avoid it with PGP.

Suppose you want to send a private message to Alice. You download Alice's public key certificate from an electronic bulletin board system (BBS). You encrypt your letter to Alice with this public key and send it to her through the BBS's email facility.

Unfortunately, unbeknownst to you or Alice, another user named Charlie has infiltrated the BBS and generated a public key of his own with Alice's user ID attached to it. He covertly substitutes his bogus key in place of Alice's real public key. You unwittingly use this bogus key belonging to Charlie instead of Alice's public key. All looks normal because this bogus key has Alice's user ID. Now Charlie can decipher the message intended for Alice because he has the matching private key. He may even reencrypt the deciphered message with Alice's real public key and send it on to her so that no one suspects any wrongdoing. Furthermore, he can even make apparently good signatures from Alice with this private key because everyone will use the bogus public key to check Alice's signatures.

The only way to prevent this disaster is to prevent anyone from tampering with public keys. If you got Alice's public key directly from Alice, this is no problem. But that may be difficult if Alice is a thousand miles away or is currently unreachable.

Perhaps you could get Alice's public key from a mutually trusted friend, David, who knows he has a good copy of Alice's public key. David could sign Alice's public key, vouching for the integrity of Alice's public key. David would create this signature with his own private key.

This would create a signed public key certificate, and would show that Alice's key had not been tampered with. This requires that you have a known good copy of David's public key to check his signature. Perhaps David could provide Alice with a signed copy of your public key also. David is thus serving as an "Introducer" between you and Alice.

This signed public key certificate for Alice could be uploaded by David or Alice to the BBS, and you could download it later. You could then check the signature via David's public key and thus be assured that this is really Alice's public key. No impostor can fool you into accepting his own bogus key as Alice's because no one else can forge signatures made by David.

A widely trusted person could even specialize in providing this service of "introducing" users to each other by providing signatures for their public key certificates. This trusted person could be regarded as a "Certificate Authority." Any public key certificates bearing the Certificate Authority's signature could be trusted as truly belonging to the person to whom they appear to belong to. All users who wanted to participate would need a known good copy of just the Certificate Authority's public key, so that the Certificate Authority's signatures could be verified. In some cases, the Certificate Authority may also act as a key server, allowing users on a network to look up public keys by asking the key server, but there is no reason why a key server must also certify keys.

A trusted centralized Certificate Authority is especially appropriate for large impersonal centrally-controlled corporate or government institutions. Some institutional environments use hierarchies of Certificate Authorities.

For more decentralized environments, allowing all users to act as trusted introducers for their friends would probably work better than a centralized key certification authority.

One of the attractive features of PGP is that it can operate equally well in a centralized environment with a Certificate Authority or in a more decentralized environment where individuals exchange personal keys.

This whole business of protecting public keys from tampering is the single most difficult problem in practical public key applications. It is the "Achilles heel" of public key cryptography, and a lot of software complexity is tied up in solving this one problem.

You should use a public key only after you are sure that it is a good public key that has not been tampered with, and that it actually belongs to the person with whom it purports to be associated. You can be sure of this if you got this public key certificate directly from its owner, or if it bears the signature of someone else that you trust, from whom you already have a good public key. Also, the user ID should have the full name of the key's owner, not just her first name.

No matter how tempted you are, you should *never* give in to expediency and trust a public key you downloaded from a bulletin board, unless it is signed by someone you trust. That uncertified public key could have been tampered with by anyone, maybe even by the system administrator of the bulletin board.

If you are asked to sign someone else's public key certificate, make certain that it really belongs to the person named in the user ID of that public key certificate. This is because your signature on her public key certificate is a promise by you that this public key really belongs to her. Other people who trust you will accept her public key because it bears your signature. It can be ill-advised to rely on hearsay—don't sign her public key unless you have independent first-hand knowledge that it really belongs to her. Preferably you should sign it only if you got it directly from her.

In order to sign a public key, you must be far more certain of that key's ownership than if you merely want to use that key to encrypt a message. To be convinced of a key's validity enough to use it, certifying signatures from trusted introducers should suffice. But to sign a key yourself, you should require your own independent first-hand knowledge of who owns that key. Perhaps you could call the key's owner on the phone and read the key fingerprint to her, to confirm that the key you have is really her key—and make sure you really are talking to the right person.

Bear in mind that your signature on a public key certificate does not vouch for the integrity of that person, but only vouches for the integrity (the ownership) of that person's public key. You aren't risking your credibility by signing the public key of a sociopath, if you are completely confident that the key really belongs to him. Other people would accept that key as belonging to him because you signed it (assuming they trust you), but they wouldn't trust that key's owner. Trusting a key is not the same as trusting the key's owner.

It would be a good idea to keep your own public key on hand with a collection of certifying signatures attached from a variety of "introducers," in the hope that most people will trust at least one of the introducers who vouch for the validity of your public key. You could post your key with its attached collection of certifying signatures on various electronic bulletin boards. If you sign someone else's public key, return it to them with your signature so that they can add it to their own collection of credentials for their own public key.

Make sure that no one else can tamper with your own public keyring. Checking a newly signed public key certificate must ultimately depend on the integrity of the trusted public keys that are already on your own public keyring. Maintain physical control of your public keyring, preferably on your own personal computer rather than on a remote time-sharing system, just as you would do for your private key. This is to protect it from tampering, not from disclosure. Keep a trusted backup copy of your public keyring and your private key on write-protected media.

Since your own trusted public key is used as a final authority to directly or indirectly certify all the other keys on your keyring, it is the most important key to protect from tampering. You may want to keep a backup copy on a write-protected floppy disk.

PGP generally assumes that you will maintain physical security over your system and your keyrings, as well as your copy of PGP itself. If an intruder can tamper with your disk, then in theory he can tamper with the program itself, rendering moot the safeguards the program may have to detect tampering with keys.

One somewhat complicated way to protect your own whole public keyring from tampering is to sign the whole ring with your own private key. You could do this by making a detached signature certificate of the public keyring.

How does PGP keep track of which keys are valid?

Before you read this section, you should read the previous section, "How to protect public keys from tampering"

PGP keeps track of which keys on your public keyring are properly certified with signatures from introducers that you trust. All you have to do is tell PGP which people you trust as introducers, and certify their keys yourself with your own ultimately trusted key. PGP can take it from there, automatically validating any other keys that have been signed by your designated introducers. And of course you can directly sign more keys yourself.

There are two entirely separate criteria that PGP uses to judge a public key's usefulness—don't get them confused:

1. Does the key actually belong to the person to whom it appears to belong? In other words, has it been certified with a trusted signature?

2. Does it belong to someone you can trust to certify other keys?

PGP can calculate the answer to the first question. To answer the second question, you must tell PGP explicitly. When you supply the answer to question 2, PGP can then calculate the answer to question 1 for other keys signed by the introducer you designated as trusted.

Keys that have been certified by a trusted introducer are deemed valid by PGP. The keys belonging to trusted introducers must themselves be certified either by you or by other trusted introducers.

PGP also allows for the possibility of your having several shades of trust for people to act as introducers. Your trust for a key's owner to act as an introducer does not just reflect your estimation of their personal integrity—it should also reflect how competent you think they are at understanding key management and using good judgment in signing keys. You can designate a person as untrusted, marginally trusted, or completely trusted to certify other public keys. This trust information is stored on your keyring with their key, but when you tell PGP to copy a key off your keyring, PGP does not copy the trust information along with the key, because your private opinions on trust are regarded as confidential.

When PGP is calculating the validity of a public key, it examines the trust level of all the attached certifying signatures. It computes a weighted score of validity—for example, two marginally trusted signatures are deemed to be as credible as one fully trusted signature. The program's skepticism is adjustable—for example, you can tune PGP to require two fully trusted signatures or three marginally trusted signatures to judge a key as valid.

Your own key is "axiomatically" valid to PGP, needing no introducer's signature to prove its validity. PGP knows which public keys are yours by looking for the corresponding private keys on the private key. PGP also assumes that you completely trust yourself to certify other keys.

As time goes on, you will accumulate keys from other people whom you may want to designate as trusted introducers. Everyone else will choose their own trusted introducers. And everyone will gradually accumulate and distribute with their key a collection of certifying signatures from other people, with the expectation that anyone receiving it will trust at least one or two of the signatures. This will cause the emergence of a decentralized fault-tolerant web of confidence for all public keys.

This unique grass-roots approach contrasts sharply with standard public key management schemes developed by government and other monolithic institutions, such as Internet Privacy Enhanced Mail (PEM), which are based on centralized control and mandatory centralized trust. The standard schemes rely on a hierarchy of Certifying Authorities who dictate who you must trust. The program's decentralized probabilistic method for determining public key legitimacy is the centerpiece of its key management architecture. PGP lets you alone choose who you trust, putting you at the top of your own private certification pyramid. PGP is for people who prefer to pack their own parachutes.

Note that while this decentralized, grass-roots approach is emphasized here, it does not mean that PGP does not perform equally well in the more hierarchical, centralized public key management schemes. Large corporate users, for example, will probably want a central figure or person who signs all the employees' keys. PGP handles that centralized scenario as a special degenerate case of PGP's more generalized trust model.

How to protect private keys from disclosure

Protect your own private key and your passphrase very carefully. If your private key is ever compromised, you'd better get the word out quickly to all interested parties before someone else uses it to make signatures in your name. For example, someone could use it to sign bogus public key certificates, which could create problems for many people, especially if your signature is widely trusted. And of course, a compromise of your own private key could expose all messages sent to you.

To protect your private key, you can start by always keeping physical control of it. Keeping it on your personal computer at home is OK, or keep it in your notebook computer that you can carry with you. If you must use an office computer that you don't always have physical control of, then keep your public and private keyrings on a write-protected removable floppy disk, and don't leave it behind when you leave the office. It wouldn't be a good idea to allow your private key to reside on a remote timesharing computer, such as a remote dial-in UNIX system. Someone could eavesdrop on your modem line and capture your passphrase and then obtain your actual private key from the remote system. You should only use your private key on a machine that is under your physical control.

Don't store your passphrase anywhere on the computer that has your private key file. Storing both the private key and the passphrase on the same computer is as dangerous as keeping your PIN in the same wallet as your Automatic Teller Machine bank card. You don't want somebody to get their hands on your disk containing both the passphrase and the private key file. It would be most secure if you just memorize your passphrase and don't store it anywhere but your brain. If you feel you must write down your passphrase, keep it well protected, perhaps even better protected than the private key file.

And keep backup copies of your private key—remember, you have the only copy of your private key, and losing it will render useless all the copies of your public key that you have spread throughout the world.

The decentralized noninstitutional approach that PGP supports for management of public keys has its benefits, but unfortunately it also means that you can't rely on a single centralized list of which keys have been compromised. This makes it a bit harder to contain the damage of a private key compromise. You just have to spread the word and hope that everyone hears about it.

If the worst case happens—your private key and passphrase are both compromised (hopefully you will find this out somehow)—you will have to issue a "key revocation" certificate. This kind of certificate is used to warn other people to stop using your public key. You can use PGP to create such a certificate by using the Revoke command from the PGPkeys menu or by having your Designated Revoker do it for you. Then you must send this to a certificate server so others can find it. Their own PGP software installs this key revocation certificate on their public keyrings and automatically prevents them from accidentally using your public key ever again. You can then generate a new private / public key pair and publish the new public key. You could send out one package containing both your new public key and the key revocation certificate for your old key.

What if you lose your private key?

Normally, if you want to revoke your own private key, you can use the Revoke command from the PGPkeys menu to issue a revocation certificate, signed with your own private key.

But what can you do if you lose your private key, or if your private key is destroyed? You can't revoke it yourself, because you must use your own private key to revoke it, and you don't have it anymore. If you do not have a designated revoker for your key, someone specified in PGP who can revoke the key on your behalf, you must ask each person who signed your key to retire his or her certification. Then anyone attempting to use your key based on the trust of one of your introducers will know not to trust your public key.

For more information on designated revokers, see the section *"Appointing a designated revoker" on page 116* in *Chapter 6*.

Beware of snake oil

When examining a cryptographic software package, the question always remains, why should you trust this product? Even if you examined the source code yourself, not everyone has the cryptographic experience to judge the security. Even if you are an experienced cryptographer, subtle weaknesses in the algorithms could still elude you.

When I was in college in the early seventies, I devised what I believed was a brilliant encryption scheme. A simple pseudorandom number stream was added to the plaintext stream to create ciphertext. This would seemingly thwart any frequency analysis of the ciphertext, and would be uncrackable even to the most resourceful government intelligence agencies. I felt so smug about my achievement.

Years later, I discovered this same scheme in several introductory cryptography texts and tutorial papers. How nice. Other cryptographers had thought of the same scheme. Unfortunately, the scheme was presented as a simple homework assignment on how to use elementary cryptanalytic techniques to trivially crack it. So much for my brilliant scheme.

From this humbling experience I learned how easy it is to fall into a false sense of security when devising an encryption algorithm. Most people don't realize how fiendishly difficult it is to devise an encryption algorithm that can withstand a prolonged and determined attack by a resourceful opponent. Many mainstream software engineers have developed equally naive encryption schemes (often even the very same encryption scheme), and some of them have been incorporated into commercial encryption software packages and sold for good money to thousands of unsuspecting users.

This is like selling automotive seat belts that look good and feel good, but snap open in the slowest crash test. Depending on them may be worse than not wearing seat belts at all. No one suspects they are bad until a real crash. Depending on weak cryptographic software may cause you to unknowingly place sensitive information at risk when you might not otherwise have done so if you had no cryptographic software at all. Perhaps you may never even discover that your data has been compromised.

Sometimes commercial packages use the Federal Data Encryption Standard (DES), a fairly good conventional algorithm recommended by the government for commercial use (but not for classified information, oddly enough—Hmmm). There are several “modes of operation” that DES can use, some of them better than others. The government specifically recommends not using the weakest simplest mode for messages, the Electronic Codebook (ECB) mode. But they do recommend the stronger and more complex Cipher Feedback (CFB) and Cipher Block Chaining (CBC) modes.

Unfortunately, most of the commercial encryption packages I've looked at use ECB mode. When I've talked to the authors of a number of these implementations, they say they've never heard of CBC or CFB modes, and don't know anything about the weaknesses of ECB mode. The very fact that they haven't even learned enough cryptography to know these elementary concepts is not reassuring. And they sometimes manage their DES keys in inappropriate or insecure ways. Also, these same software packages often include a second faster encryption algorithm that can be used instead of the slower DES. The author of the package often thinks his proprietary faster algorithm is as secure as DES, but after questioning him I usually discover that it's just a variation of my own brilliant scheme from college days. Or maybe he won't even reveal how his proprietary encryption scheme works, but assures me it's a brilliant scheme and I should trust it. I'm sure he believes that his algorithm is brilliant, but how can I know that without seeing it?

In fairness I must point out that in most cases these terribly weak products do not come from companies that specialize in cryptographic technology.

Even the really good software packages, that use DES in the correct modes of operation, still have problems. Standard DES uses a 56-bit key, which is too small by today's standards, and can now be easily broken by exhaustive key searches on special high-speed machines. The DES has reached the end of its useful life, and so has any software package that relies on it.

There is a company called AccessData (http://www.accessdata.com) that sells a very low-cost package that cracks the built-in encryption schemes used by WordPerfect, Lotus 1-2-3, MS Excel, Symphony, Quattro Pro, Paradox, MS Word, and PKZIP. It doesn't simply guess passwords—it does real cryptanalysis. Some people buy it when they forget their password for their own files. Law enforcement agencies buy it too, so they can read files they seize. I talked to Eric Thompson, the author, and he said his program only takes a split second to crack them, but he put in some delay loops to slow it down so it doesn't look so easy to the customer.

In the secure telephone arena, your choices look bleak. The leading contender is the STU-III (Secure Telephone Unit), made by Motorola and AT&T for $2,000 to $3,000, and used by the government for classified applications. It has strong cryptography, but requires some sort of special license from the government to buy this strong version. A commercial version of the STU-III is available that is watered down for NSA's convenience, and an export version is available that is even more severely weakened. Then there is the $1,200 AT&T Surity 3600, which uses the government's famous Clipper chip for encryption, with keys escrowed with the government for the convenience of wiretappers. Then, of course, there are the analog (nondigital) voice scramblers that you can buy from the spy-wannabe catalogs, that are really useless toys as far as cryptography is concerned, but are sold as "secure" communications products to customers who just don't know any better.

In some ways, cryptography is like pharmaceuticals. Its integrity may be absolutely crucial. Bad penicillin looks the same as good penicillin. You can tell if your spreadsheet software is wrong, but how do you tell if your cryptography package is weak? The ciphertext produced by a weak encryption algorithm looks as good as ciphertext produced by a strong encryption algorithm. There's a lot of snake oil out there. A lot of quack cures. Unlike the patent medicine hucksters of old, these software implementors usually don't even know their stuff is snake oil. They may be good software engineers, but they usually haven't even read any of the academic literature in cryptography. But they think they can write good cryptographic software. And why not? After all, it seems intuitively easy to do so. And their software seems to work OK.

Anyone who thinks they have devised an unbreakable encryption scheme either is an incredibly rare genius or is naive and inexperienced. Unfortunately, I sometimes have to deal with would-be cryptographers who want to make “improvements” to PGP by adding encryption algorithms of their own design.

I remember a conversation with Brian Snow, a highly placed senior cryptographer with the NSA. He said he would never trust an encryption algorithm designed by someone who had not “earned their bones” by first spending a lot of time cracking codes. That made a lot of sense. I observed that practically no one in the commercial world of cryptography qualifies under this criterion. “Yes,” he said with a self-assured smile, “And that makes our job at NSA so much easier.” A chilling thought. I didn’t qualify either.

The government has peddled snake oil too. After World War II, the United States sold German Enigma ciphering machines to third-world governments. But they didn’t tell them that the Allies cracked the Enigma code during the war, a fact that remained classified for many years. Even today many UNIX systems worldwide use the Enigma cipher for file encryption, in part because the government has created legal obstacles against using better algorithms. They even tried to prevent the initial publication of the RSA algorithm in 1977. And they have for many years squashed essentially all commercial efforts to develop effective secure telephones for the general public.

The principal job of the United States government’s National Security Agency is to gather intelligence, principally by covertly tapping into people’s private communications (see James Bamford’s book, *The Puzzle Palace*). The NSA has amassed considerable skill and resources for cracking codes. When people can’t get good cryptography to protect themselves, it makes NSA’s job much easier. NSA also has the responsibility of approving and recommending encryption algorithms. Some critics charge that this is a conflict of interest, like putting the fox in charge of guarding the hen house. In the 1980s, NSA had been pushing a conventional encryption algorithm that they designed (the COMSEC Endorsement Program), and they won’t tell anybody how it works because that’s classified. They wanted others to trust it and use it. But any cryptographer can tell you that a well-designed encryption algorithm does not have to be classified to remain secure. Only the keys should need protection. How does anyone else really know if NSA’s classified algorithm is secure? It’s not that hard for NSA to design an encryption algorithm that only they can crack, if no one else can review the algorithm.

There are three main factors that have undermined the quality of commercial cryptographic software in the United States.

- The first is the virtually universal lack of competence of implementors of commercial encryption software (although this is starting to change since the publication of PGP). Every software engineer fancies himself a cryptographer, which has led to the proliferation of really bad crypto software.
- The second is the NSA deliberately and systematically suppressing all the good commercial encryption technology, by legal intimidation and economic pressure. Part of this pressure is brought to bear by stringent export controls on encryption software which, by the economics of software marketing, has the net effect of suppressing domestic encryption software.
- The third principle method of suppression comes from the granting of all the software patents for all the public key encryption algorithms to a single company, affording a single choke point to suppress the spread of this technology (although this crypto patent cartel broke up in the fall of 1995).

The net effect of all this is that before PGP was published, there was almost no highly secure general purpose encryption software available in the United States.

I'm not as certain about the security of PGP as I once was about my brilliant encryption software from college. If I were, that would be a bad sign. But I don't think PGP contains any glaring weaknesses (although I'm pretty sure it contains bugs). I have selected the best algorithms from the published literature of civilian cryptologic academia. For the most part, these algorithms have been individually subject to extensive peer review. I know many of the world's leading cryptographers, and have discussed with some of them many of the cryptographic algorithms and protocols used in PGP. It's well researched, and has been years in the making. And I don't work for the NSA. But you don't have to trust my word on the cryptographic integrity of PGP, because source code is available to facilitate peer review.

One more point about my commitment to cryptographic quality in PGP: Since I first developed and released PGP for free in 1991, I spent three years under criminal investigation by U.S. Customs for PGP's spread overseas, with risk of criminal prosecution and years of imprisonment. By the way, you didn't see the government getting upset about other cryptographic software—it's PGP that really set them off. What does that tell you about the strength of PGP? I have earned my reputation on the cryptographic integrity of my products. I will not betray my commitment to our right to privacy, for which I have risked my freedom. I'm not about to allow a product with my name on it to have any secret back doors.

Vulnerabilities

"If all the personal computers in the world—260 million—were put to work on a single PGP-encrypted message, it would still take an estimated 12 million times the age of the universe, on average, to break a single message."

--William Crowell, Deputy Director, National Security Agency, March 20, 1997.

No data security system is impenetrable. PGP can be circumvented in a variety of ways. In any data security system, you have to ask yourself if the information you are trying to protect is more valuable to your attacker than the cost of the attack. This should lead you to protect yourself from the cheapest attacks, while not worrying about the more expensive attacks.

Some of the discussion that follows may seem unduly paranoid, but such an attitude is appropriate for a reasonable discussion of vulnerability issues.

Compromised passphrase and private key

Probably the simplest attack comes if you leave the passphrase for your private key written down somewhere. If someone gets it and also gets your private key file, they can read your messages and make signatures in your name.

Here are some recommendations for protecting your passphrase:

1. Don't use obvious passphrases that can be easily guessed, such as the names of your kids or spouse.

2. Use spaces and a combination of numbers and letters in your passphrase. If you make your passphrase a single word, it can be easily guessed by having a computer try all the words in the dictionary until it finds your password. That's why a passphrase is so much better than a password. A more sophisticated attacker may have his computer scan a book of famous quotations to find your passphrase.

3. Be creative. Use an easy to remember but hard to guess passphrase; you can easily construct one by using some creatively nonsensical sayings or obscure literary quotes.

Public key tampering

A major vulnerability exists if public keys are tampered with. This may be the most crucially important vulnerability of a public key cryptosystem, in part because most novices don't immediately recognize it.

To summarize: When you use someone's public key, make certain it has not been tampered with. A new public key from someone else should be trusted only if you got it directly from its owner, or if it has been signed by someone you trust. Make sure no one else can tamper with your own public keyring. Maintain physical control of both your public keyring and your private key, preferably on your own personal computer rather than on a remote timesharing system. Keep a backup copy of both keyrings.

Not quite deleted files

Another potential security problem is caused by how most operating systems delete files. When you encrypt a file and then delete the original plaintext file, the operating system doesn't actually physically erase the data. It merely marks those disk blocks as deleted, allowing the space to be reused later. It's sort of like discarding sensitive paper documents in the paper recycling bin instead of the paper shredder. The disk blocks still contain the original sensitive data you wanted to erase, and will probably be overwritten by new data at some point in the future. If an attacker reads these deleted disk blocks soon after they have been deallocated, he could recover your plaintext.

In fact, this could even happen accidentally, if something went wrong with the disk and some files were accidentally deleted or corrupted. A disk recovery program may be run to recover the damaged files, but this often means that some previously deleted files are resurrected along with everything else. Your confidential files that you thought were gone forever could then reappear and be inspected by whoever is attempting to recover your damaged disk. Even while you are creating the original message with a word processor or text editor, the editor may be creating multiple temporary copies of your text on the disk, just because of its internal workings. These temporary copies of your text are deleted by the word processor when it's done, but these sensitive fragments are still on your disk somewhere.

The only way to prevent the plaintext from reappearing is to somehow cause the deleted plaintext files to be overwritten. Unless you know for sure that all the deleted disk blocks will soon be reused, you must take positive steps to overwrite the plaintext file, and also any fragments of it on the disk left by your word processor. You can take care of any fragments of the plaintext left on the disk by using PGP's Secure Wipe and Freespace Wipe features.

Viruses and Trojan horses

Another attack could involve a specially tailored hostile computer virus or worm that might infect PGP or your operating system. This hypothetical virus could be designed to capture your passphrase or private key or deciphered messages and to covertly write the captured information to a file or send it through a network to the virus's owner. Or it might alter PGP's behavior so that signatures are not properly checked. This attack is cheaper than cryptanalytic attacks.

Defending against this kind of attack falls into the category of defending against viral infection generally. There are some moderately capable antiviral products commercially available, and there are hygienic procedures to follow that can greatly reduce the chances of viral infection. A complete treatment of antiviral and antiworm countermeasures is beyond the scope of this document. PGP has no defenses against viruses, and assumes that your own personal computer is a trustworthy execution environment. If such a virus or worm actually appeared, hopefully word would soon get around warning everyone.

A similar attack involves someone creating a clever imitation of PGP that behaves like PGP in most respects, but that doesn't work the way it's supposed to. For example, it might be deliberately crippled to not check signatures properly, allowing bogus key certificates to be accepted. This *Trojan horse* version of PGP is not hard for an attacker to create, because PGP source code is widely available, so anyone could modify the source code and produce a lobotomized zombie imitation PGP that looks real but does the bidding of its diabolical master. This Trojan horse version of PGP could then be widely circulated, claiming to be from a legitimate source. How insidious.

You should make an effort to get your copy of PGP directly from Network Associates, Inc.

There are other ways to check PGP for tampering, using digital signatures. You could use another trusted version of PGP to check the signature on a suspect version of PGP. But this won't help at all if your operating system is infected, nor will it detect if your original copy of pgp.exe has been maliciously altered in such a way as to compromise its own ability to check signatures. This test also assumes that you have a good trusted copy of the public key that you use to check the signature on the PGP executable.

Swap files or virtual memory

PGP was originally developed for MS-DOS, a primitive operating system by today's standards. But as it was ported to other more complex operating systems, such as Microsoft Windows and the Macintosh OS, a new vulnerability emerged. This vulnerability stems from the fact that these fancier operating systems use a technique called *virtual memory*.

Virtual memory allows you to run huge programs on your computer that are bigger than the space available in your computer's semiconductor memory chips. This is handy because software has become more and more bloated since graphical user interfaces became the norm and users started running several large applications at the same time. The operating system uses the hard disk to store portions of your software that aren't being used at the moment. This means that the operating system might, without your knowledge, write out to disk some things that you thought were kept only in main memory—-things like keys, passphrases, and decrypted plaintext. PGP does not keep that kind of sensitive data lying around in memory for longer than necessary, but there is some chance that the operating system could write it out to disk anyway.

The data is written out to some scratchpad area of the disk, known as a *swap file*. Data is read back in from the swap file as needed, so that only part of your program or data is in physical memory at any one time. All this activity is invisible to the user, who just sees the disk chattering away. Microsoft Windows swaps chunks of memory, called *pages*, using a Least Recently Used (LRU) page-replacement algorithm. This means pages that have not been accessed for the longest period of time are the first ones to be swapped to the disk. This approach suggests that in most cases the risk is fairly low that sensitive data will be swapped out to disk, because PGP doesn't leave it in memory for very long. But we don't make any guarantees.

This swap file can be accessed by anyone who can get physical access to your computer. If you are concerned about this problem, you may be able to solve it by obtaining special software that overwrites your swap file. Another possible cure is to turn off your operating system's virtual memory feature. Microsoft Windows allows this, and so does the Mac OS. Turning off virtual memory may mean that you need to have more physical RAM chips installed in order to fit everything in RAM.

Physical security breach

A physical security breach may allow someone to physically acquire your plaintext files or printed messages. A determined opponent might accomplish this through burglary, trash-picking, unreasonable search and seizure, or bribery, blackmail, or infiltration of your staff. Some of these attacks may be especially feasible against grass-roots political organizations that depend on a largely volunteer staff.

Don’t be lulled into a false sense of security just because you have a cryptographic tool. Cryptographic techniques protect data only while it’s encrypted—direct physical security violations can still compromise plaintext data or written or spoken information.

This kind of attack is cheaper than cryptanalytic attacks on PGP.

Tempest attacks

Another kind of attack that has been used by well-equipped opponents involves the remote detection of the electromagnetic signals from your computer. This expensive and somewhat labor-intensive attack is probably still cheaper than direct cryptanalytic attacks. An appropriately instrumented van can park near your office and remotely pick up all of your keystrokes and messages displayed on your computer video screen. This would compromise all of your passwords, messages, and so on. This attack can be thwarted by properly shielding all of your computer equipment and network cabling so that it does not emit these signals. This shielding technology, known as "Tempest," is used by some government agencies and defense contractors. There are hardware vendors who supply Tempest shielding commercially.

Some newer versions of PGP (after version 6.0) can display decrypted plaintext using a specially designed font that may have reduced levels of radio frequency emissions from your computer's video screen. This may make it harder for the signals to be remotely detected. This special font is available in some versions of PGP that support the "Secure Viewer" feature.

Protecting against bogus timestamps

A somewhat obscure vulnerability of PGP involves dishonest users creating bogus timestamps on their own public key certificates and signatures. You can skip over this section if you are a casual user and aren’t deeply into obscure public-key protocols.

There's nothing to stop a dishonest user from altering the date and time setting of his own system's clock, and generating his own public key certificates and signatures that appear to have been created at a different time. He can make it appear that he signed something earlier or later than he actually did, or that his public/private key pair was created earlier or later. This may have some legal or financial benefit to him, for example by creating some kind of loophole that might allow him to repudiate a signature.

I think this problem of falsified timestamps in digital signatures is no worse than it is already in handwritten signatures. Anyone can write any date next to their handwritten signature on a contract, but no one seems to be alarmed about this state of affairs. In some cases, an "incorrect" date on a handwritten signature might not be associated with actual fraud. The timestamp might be when the signator asserts that he signed a document, or maybe when he wants the signature to go into effect.

In situations where it is critical that a signature be trusted to have the actual correct date, people can simply use notaries to witness and date a handwritten signature. The analog to this in digital signatures is to get a trusted third party to sign a signature certificate, applying a trusted timestamp. No exotic or overly formal protocols are needed for this. Witnessed signatures have long been recognized as a legitimate way of determining when a document was signed.

A trustworthy Certifying Authority or notary could create notarized signatures with a trustworthy timestamp. This would not necessarily require a centralized authority. Perhaps any trusted introducer or disinterested party could serve this function, the same way real notary publics do now. When a notary signs other people's signatures, it creates a signature certificate of a signature certificate. This would serve as a witness to the signature in the same way that real notaries now witness handwritten signatures. The notary could enter the detached signature certificate (without the actual whole document that was signed) into a special log controlled by the notary. Anyone could read this log. The notary's signature would have a trusted timestamp, which might have greater credibility or more legal significance than the timestamp in the original signature.

There is a good treatment of this topic in Denning's 1983 article in IEEE Computer. Future enhancements to PGP might have features to easily manage notarized signatures of signatures, with trusted timestamps.

Exposure on multi-user systems

PGP was originally designed for a single-user PC under your direct physical control. If you run PGP at home on your own PC, your encrypted files are generally safe, unless someone breaks into your house, steals your PC and persuades you to give them your passphrase (or your passphrase is simple enough to guess).

PGP is not designed to protect your data while it is in plaintext form on a compromised system. Nor can it prevent an intruder from using sophisticated measures to read your private key while it is being used. You will just have to recognize these risks on multiuser systems, and adjust your expectations and behavior accordingly. Perhaps your situation is such that you should consider only running PGP on an isolated single-user system under your direct physical control.

Traffic analysis

Even if the attacker cannot read the contents of your encrypted messages, he may be able to infer at least some useful information by observing where the messages come from and where they are going, the size of the messages, and the time of day the messages are sent. This is analogous to the attacker looking at your long-distance phone bill to see who you called and when and for how long, even though the actual content of your calls is unknown to the attacker. This is called traffic analysis. PGP alone does not protect against traffic analysis. Solving this problem would require specialized communication protocols designed to reduce exposure to traffic analysis in your communication environment, possibly with some cryptographic assistance.

Cryptanalysis

An expensive and formidable cryptanalytic attack could possibly be mounted by someone with vast supercomputer resources, such as a government intelligence agency. They might crack your public key by using some new secret mathematical breakthrough. But civilian academia has been intensively attacking public key cryptography without success since 1978.

Perhaps the government has some classified methods of cracking the conventional encryption algorithms used in PGP. This is every cryptographer’s worst nightmare. There can be no absolute security guarantees in practical cryptographic implementations.

Still, some optimism seems justified. The public key algorithms, message digest algorithms, and block ciphers used in PGP were designed by some of the best cryptographers in the world. PGP's algorithms has had extensive security analysis and peer review from some of the best cryptanalysts in the unclassified world.

Besides, even if the block ciphers used in PGP have some subtle unknown weaknesses, PGP compresses the plaintext before encryption, which should greatly reduce those weaknesses. The computational workload to crack it is likely to be much more expensive than the value of the message.

If your situation justifies worrying about very formidable attacks of this caliber, then perhaps you should contact a data security consultant for some customized data security approaches tailored to your special needs.

In summary, without good cryptographic protection of your data communications, it may be practically effortless and perhaps even routine for an opponent to intercept your messages, especially those sent through a modem or email system. If you use PGP and follow reasonable precautions, the attacker will have to expend far more effort and expense to violate your privacy.

If you protect yourself against the simplest attacks, and you feel confident that your privacy is not going to be violated by a determined and highly resourceful attacker, then you'll probably be safe using PGP. PGP gives you Pretty Good Privacy.

Biometric Word Lists D

Biometric Word Lists

By Philip Zimmermann and Patrick Juola

PGP uses a special list of words to convey binary information in an authenticated manner over a voice channel, such as a telephone, via biometric signatures. The human voice that speaks the words, if recognized by the listener, serves as a means of biometric authentication of the data carried by the words. The word list serves the same purpose as the military alphabet, which is used to transmit letters over a noisy radio voice channel. But the military alphabet has 26 words, each word representing one letter. For our purposes, our list has 256 carefully selected phonetically distinct words to represent the 256 possible byte values of 0 to 255.

We created a word list for reading binary information over the phone, with each word representing a different byte value. We tried to design the word list to be useful for a variety of applications. The first application we had envisioned was to read PGP public key fingerprints over the phone to authenticate the public key. In that case, the fingerprint is 20 bytes long, requiring 20 words to be read aloud. Experience has shown it to be fairly tedious and error prone to read that many bytes in hexadecimal, so it seems worth using a word list to represent each byte by a word.

Some applications may require transmitting even lengthier byte sequences over the phone, for example, entire keys or signatures. This may entail reading more than a hundred bytes. Using words instead of hex bytes seems even more justified in that case.

When reading long sequences of bytes aloud, errors may creep in. The kinds of error syndromes you get on human-spoken data are different than they are for transmitting data through a modem. Modem errors usually involve flipped bits from line noise. Error detection methods for modems usually involve CRCs to be added, which are optimized for detecting line noise bursts. However, random sequences of spoken human words usually involves one of three kinds of errors: 1) transposition of two consecutive words, 2) duplicate words, or 3) omitted words. If we are to design an error detection scheme for this kind of data transmission channel, we should make one that is optimized for these three kinds of errors. Zhahai Stewart suggested a good scheme (in personal conversation with me in 1991) for error detection of these errors.

Stewart’s scheme for error detection while reading aloud long sequences of bytes via a word list entails using not one, but two lists of words. Each list contains 256 phonetically distinct words, each word representing a different byte value between 0 and 255. The two lists are used alternately for the even-offset bytes and the odd-offset bytes in the byte sequence.

For example, the first byte (offset 0 in the sequence) is used to select a word from the even list. The byte at offset 1 is used to select a byte from the odd list. The byte at offset 2 selects a word from the even list again, and the byte at offset 3 selects from the odd list again. Each byte value is actually represented by two different words, depending on whether that byte appears at an even or an odd offset from the beginning of the byte sequence. For example, suppose the word “adult” and the word “amulet” each appears in the same corresponding position in the two word lists, position 5. That means that the repeating 3-byte sequence 05 05 05 is represented by the 3-word sequence “adult, amulet, adult.”

This approach makes it easy to detect all three kinds of common errors in spoken data streams: transposition, duplication, and omission. A transposition will result in two consecutive words from the even list followed by two consecutive words from the odd list (or the other way around). A duplication will be detected by two consecutive duplicate words, a condition that cannot occur in a normal sequence. An omission will be detected by two consecutive words drawn from the same list.

To facilitate the immediate and obvious detection by a human of any of the three error syndromes described above, without computer assistance, we made the two lists have one obviously different property: The even list contains only two-syllable words, while the odd list contains only three-syllable words. That suggestion came from Patrick Juola, a computational linguist.

PGPfone was the application that precipitated the actual development of the word list by Juola and Zimmermann. PGPfone is an application that turns your computer into a secure telephone. We used it to authenticate PGPfone's initial Diffie-Hellman key exchange without using digital signatures and public key infrastructures. We knew we would end up using it for authenticating PGP key fingerprints when we applied it to PGP later.

The idea behind building the word lists was to develop a metric to measure the phonetic distance between two words, then use that as a goodness measure to develop a full list. Grady Ward provided us with a large collection of words and their pronunciations, and Patrick Juola used genetic algorithms to evolve the best subset of Ward's list. To briefly summarize what he did, he made a large population of guesses and let the population “sexually reproduce” by exchanging words with other guesses -- and, like biological evolution, the better guesses survived into the next generation. After about 200 generations, the list had mostly stabilized into a best guess, with far greater phonetic distance between the words than what we started with in the initial guess lists.

The first major hurdle was the development of the metric. Linguists have studied sound production and perception for decades, and there is a standard feature set used to describe sounds in English. For example, say the words “pun,” “fun,” “dun,” and “gun” (go ahead, try it), and notice how your tongue keeps moving back in your mouth on each word. Linguists call this the “place of articulation,” and noises that are very different in this feature sound different to English speakers. Combining the features of all the sounds in a word gives us a representation of the sound of the entire word -- and we can compute the phonetic distance between a pair of words.

Actually, it wasn't that simple. We didn't know how to weight the various features, certain word-level features like accents were hard to represent, and the feature-based analysis simply fails for certain sounds. There were also a few other more subtle criteria; for example, we wanted the words to be common enough to be universally recognizable, but not so common as to be boring --and we didn't want confusing words like “repeat” or “begin” or “error”. Some sound features are less perceptible to non-native-English speakers, for example, some Japanese speakers might hear and pronounce “r” and “l” the same way. It would be nice if the words were short enough that you could fit enough of them on a small LCD display. Large consonant clusters (“corkscrew” has five pronounced consonants in a row) are sometimes hard to say, especially to non-English speakers. One way or another, we tried to incorporate all these criteria into a filter on the initial dictionary list or into the distance metric itself.

After the computer evolved the winning list, we looked at it. Yes, the words were phonetically distinct. But many of them looked like a computer picked them, not a human. A lot of them were just ugly and dumb. Some were repugnant, and some were bland and wimpy. So we applied some “wetware” augmentation to the list. Some words were deleted, and replaced by some human-chosen words. We had the computer check the new words against the list to see if they were phonetically distant from the rest of the list. We also tried to make the words not come too close to colliding phonetically with the other words in the larger dictionary, just so that they would not be mistaken for other words not on the list.

There were a variety of selection criteria that Juola used in his algorithms. He published a paper on it that goes into more detail. This document is just a brief overview of how we built the list.

I’m not entirely happy with the word list. I wish it had more cool words in it, and less bland words. I like words like “Aztec” and “Capricorn”, and the words in the standard military alphabet. While we'd like to reserve the right to revise the list at some future time, it's not likely, due to the legacy problems that this initial version will create. This version of the list was last modified in September 1998.

If you have any suggested words you'd like to see added or deleted, send them in to pgpfone-bugs@mit.edu, and while you're at it, send a copy to Patrick Juola at juola@mathcs.duq.edu. Here are the full word lists, both odd and even.

Two Syllable Word List

aardvark
absurd
accrue
acme
adrift
adult
afflict
ahead
aimless
Algol
allow
alone
ammo
ancient
apple
artist
assume
Athens
atlas
Aztec
baboon
backfield
backward
banjo
beaming
bedlamp
beehive
beeswax
befriend
Belfast
berserk
billiard
bison
blackjack
blockade
blowtorch
bluebird
bombast
bookshelf
brackish
breadline
breakup
brickyard
briefcase
Burbank
button
buzzard
cement
chairlift
chatter
checkup
chisel
choking
chopper
Christmas
clamshell
classic
classroom
cleanup
clockwork
cobra
commence
concert
cowbell
crackdown
cranky
crowfoot
crucial
crumpled
crusade
cubic
dashboard
deadbolt
deckhand
dogsled
dragnet
drainage
dreadful
drifter
dropper
drumbeat
drunken
Dupont
dwelling
eating
edict
egghead
eightball
endorse
endow
enlist
erase
escape
exceed
eyeglass
eyetooth
facial
fallout
flagpole
flatfoot
flytrap
fracture
framework
freedom
frighten
gazelle
Geiger
glitter
glucose
goggles
goldfish
gremlin
guidance
hamlet
highchair
hockey
indoors
indulge
inverse
involve
island
jawbone
keyboard
kickoff
kiwi
klaxon
locale
lockup
merit
minnow
miser
Mohawk
mural
music
necklace
Neptune
newborn
nightbird
Oakland
obtuse
offload
optic
orca
payday
peachy
pheasant
physique
playhouse
Pluto
preclude
prefer
preshrunk
printer
prowler
pupil
puppy
python
quadrant
quiver
quota
ragtime
ratchet
rebirth
reform
regain
reindeer
rematch
repay
retouch
revenge
reward
rhythm
ribcage
ringbolt
robust
rocker
ruffled
sailboat
sawdust
scallion
scenic
scorecard
Scotland
seabird
select
sentence
shadow
shamrock
showgirl
skullcap
skydive
slingshot
slowdown
snapline
snapshot
snowcap
snowslide
solo
southward
soybean
spaniel
spearhead
spellbind
spheroid
spigot
spindle
spyglass
stagehand
stagnate
stairway
standard
stapler
steamship
sterling
stockman
stopwatch
stormy
sugar
surmount
suspense
sweatband
swelter
tactics
talon
tapeworm
tempest
tiger
tissue
tonic
topmost
tracker
transit
trauma
treadmill
Trojan
trouble
tumor
tunnel
tycoon
uncut
unearth
unwind
uproot
upset
upshot
vapor
village
virus
Vulcan
waffle
wallet
watchword
wayside
willow
woodlark
Zulu

Three Syllable Word List

adroitness
adviser
aftermath
aggregate
alkali
almighty
amulet
amusement
antenna
applicant
Apollo
armistice
article
asteroid
Atlantic
atmosphere
autopsy
Babylon
backwater
barbecue
belowground
bifocals
bodyguard
bookseller
borderline
bottomless
Bradbury
bravado
Brazilian
breakaway
Burlington
businessman
butterfat
Camelot
candidate
cannonball
Capricorn
caravan
caretaker
celebrate
cellulose
certify
chambermaid
Cherokee
Chicago
clergyman
coherence
combustion
commando
company
component
concurrent
confidence
conformist
congregate
consensus
consulting
corporate
corrosion
councilman
crossover
crucifix
cumbersome
customer
Dakota
decadence
December
decimal
designing
detector
detergent
determine
dictator
dinosaur
direction
disable
disbelief
disruptive
distortion
document
embezzle
enchanting
enrollment
enterprise
equation
equipment
escapade
Eskimo
everyday
examine
existence
exodus
fascinate
filament
finicky
forever
fortitude
frequency
gadgetry
Galveston
getaway
glossary
gossamer
graduate
gravity
guitarist
hamburger
Hamilton
handiwork
hazardous
headwaters
hemisphere
hesitate
hideaway
holiness
hurricane
hydraulic
impartial
impetus
inception
indigo
inertia
infancy
inferno
informant
insincere
insurgent
integrate
intention
inventive
Istanbul
Jamaica
Jupiter
leprosy
letterhead
liberty
maritime
matchmaker
maverick
Medusa
megaton
microscope
microwave
midsummer
millionaire
miracle
misnomer
molasses
molecule
Montana
monument
mosquito
narrative
nebula
newsletter
Norwegian
October
Ohio
onlooker
opulent
Orlando
outfielder
Pacific
pandemic
Pandora
paperweight
paragon
paragraph
paramount
passenger
pedigree
Pegasus
penetrate
perceptive
performance
pharmacy
phonetic
photograph
pioneer
pocketful
politeness
positive
potato
processor
provincial
proximate
puberty
publisher
pyramid
quantity
racketeer
rebellion
recipe
recover
repellent
replica
reproduce
resistor
responsive
retraction
retrieval
retrospect
revenue
revival
revolver
sandalwood
sardonic
Saturday
savagery
scavenger
sensation
sociable
souvenir
specialist
speculate
stethoscope
stupendous
supportive
surrender
suspicious
sympathy
tambourine
telephone
therapist
tobacco
tolerance
tomorrow
torpedo
tradition
travesty
trombonist
truncated
typewriter
ultimate
undaunted
underfoot
unicorn
unify
universe
unravel
upcoming
vacancy
vagabond
vertigo
Virginia
visitor
vocalist
voyager
warranty
Waterloo
whimsical
Wichita
Wilmington
Wyoming
yesteryear
Yucatan

Glossary

AES (Advanced Encryption Standard) NIST approved standards, usually used for the next 20 - 30 years.

Algorithm (encryption) a set of mathematical rules (logic) used in the processes of encryption and decryption.

Algorithm (hash) a set of mathematical rules (logic) used in the processes of message digest creation and key / signature generation.

Anonymity of unknown or undeclared origin or authorship, concealing an entity’s identification.

ANSI (American National Standards Institute) develops standards through various Accredited Standards Committees (ASC). The X9 committee focuses on security standards for the financial services industry.

ASCII-armored text binary information that has been encoded using a standard, printable, 7-bit ASCII character set, for convenience in transporting the information through communication systems. In the PGP program, ASCII armored text files are given the default filename extension, and they are encoded and decoded in the ASCII radix-64 format.

Asymmetric keys a separate but integrated user key-pair, comprised of one public key and one private key. Each key is one way, meaning that a key used to encrypt information can not be used to decrypt the same data.

Authentication the determination of the origin of encrypted information through the verification of someone’s digital signature or someone’s public key by checking its unique fingerprint.

Authorization certificate an electronic document to prove one’s access or privilege rights, also to prove one is who they say they are.

Authorization to convey official sanction, access or legal power to an entity.

Blind signature

ability to sign documents without knowledge of content, similar to a notary public.

Block cipher

a symmetric cipher operating on blocks of plain text and cipher text, usually 64 bits.

CA (Certificate Authority)

a trusted third party (TTP) who creates certificates that consist of assertions on various attributes and binds them to an entity and / or to their public key.

CAPI (Crypto API)

Microsoft’s crypto API for Windows-based operating systems and applications.

CAST

a 64-bit block cipher using 64-bit key, six S-boxes with 8-bit input and 32-bit output, developed in Canada by Carlisle Adams and Stafford Tavares.

Certificate (digital certificate)

an electronic document attached to a public key by a trusted third party, which provides proof that the public key belongs to a legitimate owner and has not been compromised.

Certification

endorsement of information by a trusted entity.

Certify

to sign another person's public key.

Certifying authority

one or more trusted individuals who are assigned the responsibility of certifying the origin of keys and adding them to a common database.

Ciphertext

plaintext converted into a secretive format through the use of an encryption algorithm. An encryption key can unlock the original plaintext from ciphertext.

Clear text

characters in a human readable form or bits in a machine-readable form (also called *plain text*).

Corporate signing key

a public key that is designated by the security officer of a corporation as the system-wide key that all corporate users trust to sign other keys.

Conventional encryption encryption that relies on a common passphrase instead of public key cryptography. The file is encrypted using a session key, which encrypts using a passphrase that you will be asked to choose

Cryptanalysis the art or science of transferring cipher text into plain text without initial knowledge of the key used to encrypt the plain text.

CRYPTOKI same as PKCS #11.

Cryptography the art and science of creating messages that have some combination of being private, signed, unmodified with non-repudiation.

Cryptosystem a system comprised of cryptographic algorithms, all possible plain text, cipher text, and keys.

Data integrity a method of ensuring information has not been altered by unauthorized or unknown means.

Decryption a method of unscrambling encrypted information so that it becomes legible again. The recipient’s private key is used for decryption.

DES (Data Encryption Standard) a 64-bit block cipher, symmetric algorithm also known as Data Encryption Algorithm (DEA) by ANSI and DEA-1 by ISO. Widely used for over 20 years, adopted in 1976 as FIPS 46.

Dictionary attack a calculated brute force attack to reveal a password by trying obvious and logical combinations of words.

Diffie-Hellman the first public key algorithm, invented in 1976, using discrete logarithms in a finite field.

Digital cash electronic money that is stored and transferred through a variety of complex protocols.

Direct trust an establishment of peer-to-peer confidence.

Digital signature see signature.

DSA (Digital Signature Algorithm)
a public key digital signature algorithm proposed by NIST for use in DSS.

DSS (Digital Signature Standard)
a NIST proposed standard (FIPS) for digital signatures using DSA.

ECC (Elliptic Curve Cryptosystem)
a unique method for creating public key algorithms based on mathematical curves over finite fields or with large prime numbers.

EES (Escrowed Encryption Standard)
a proposed U.S. government standard for escrowing private keys.

Elgamal scheme
used for both digital signatures and encryption based on discrete logarithms in a finite field; can be used with the DSA function.

Encryption
a method of scrambling information to render it unreadable to anyone except the intended recipient, who must decrypt it to read it.

Fingerprint
a uniquely identifying string of numbers and characters used to authenticate public keys. This is the primary means for checking the authenticity of a key. See *Key Fingerprint*.

FIPS (Federal Information Processing Standard)
a U.S. government standard published by NIST.

Firewall
a combination of hardware and software that protects the perimeter of the public/private network against certain attacks to ensure some degree of security.

Hash function
a one-way hash function - a function that produces a message digest that cannot be reversed to produced the original.

Hierarchical trust
a graded series of entities that distribute trust in an organized fashion, commonly used in ANSI X.509 issuing certifying authorities.

HTTP (HyperText Transfer Protocol)
a common protocol used to transfer documents between servers or from a server to a client.

Hexadecimal

hexadecimal describes a base-16 number system. That is, it describes a numbering system containing 16 sequential numbers as base units (including 0) before adding a new position for the next number. (Note that we're using "16" here as a decimal number to explain a number that would be "10" in hexadecimal.) The hexadecimal numbers are 0-9 and then use the letters A-F.

IDEA (International Data Encryption Standard)

a 64-bit block symmetric cipher using 128-bit keys based on mixing operations from different algebraic groups. Considered one of the strongest algorithms.

IKE (Internet Key Exchange)

provides a secure means of key exchange over the Internet. IKE is also a candidate for IPSec security archetecture.

Implicit trust

Implicit trust is reserved for key *pairs* located on your local keyring. If the private portion of a key pair is found on your keyring, PGP assumes that you are the owner of the key pair and that you implicity trust yourself.

Integrity

assurance that data is not modified (by unauthorized persons) during storage or transmittal.

Introducer

a person or organization who is allowed to vouch for the authenticity of someone's public key. You designate an introducer by signing their public key.

IPSec

a TCP/IP layer encryption scheme under consideration within the IETF.

ISO (International Organization for Standardization)

responsible for a wide range of standards, like the OSI model and international relationship with ANSI on X.509.

Key

a digital code used to encrypt and sign and decrypt and verify messages and files. Keys come in key pairs and are stored on keyrings.

Key escrow/recovery

a practice where a user of a public key encryption system surrenders their private key to a third party thus permitting them to monitor encrypted communications.

Key exchange a scheme for two or more nodes to transfer a secret session key across an unsecured channel.

Key fingerprint a uniquely identifying string of numbers and characters used to authenticate public keys. For example, you can telephone the owner of a public key and have him or her read the fingerprint associated with their key so you can compare it with the fingerprint on your copy of their public key to see if they match. If the fingerprint does not match, then you know you have a bogus key.

Key ID a legible code that uniquely identifies a key pair. Two key pairs may have the same user ID, but they will have different Key IDs.

Key length the number of bits representing the key size; the longer the key, the stronger it is.

Key management the process and procedure for safely storing and distributing accurate cryptographic keys; the overall process of generating and distributing cryptographic key to authorized recipients in a secure manner.

Key pair a public key and its complimentary private key. In public-key cryptosystems, like the PGP program, each user has at least one key pair.

Keyring a set of keys. Each user has two types of keyrings: a private keyring and a public keyring.

Key splitting or "secret sharing" the process of dividing up a private key into multiple pieces, and share those pieces among a group of people. A designated number of those people must bring their shares of the key together to use the key.

LDAP (Lightweight Directory Access Protocol) a simple protocol that supports access and search operations on directories containing information such as names, phone numbers, and addresses across otherwise incompatible systems over the Internet.

Message digest a compact “distillate” of your message or file checksum. It represents your message, such that if the message were altered in any way, a different message digest would be computed from it.

Meta-introducer a trusted introducer of trusted introducers.

MIC (Message Integrity Check) originally defined in PEM for authentication using MD2 or MD5. Micalg (message integrity calculation) is used in secure MIME implementations.

MIME (Multipurpose Internet Mail Extensions) a freely available set of specifications that offers a way to interchange text in languages with different character sets, and multimedia email among many different computer systems that use Internet mail standards.

Non-repudiation preventing the denial of previous commitments or actions.

One-way hash a function of a variable string to create a fixed length value representing the original pre-image, also called message digest, fingerprint, message integrity check (MIC).

Passphrase an easy-to-remember phrase used for better security than a single password; key crunching converts it into a random key.

Password a sequence of characters or a word that a subject submits to a system for purposes of authentication, validation, or verification.

PGP/MIME an IETF standard (RFC 2015) that provides privacy and authentication using the Multipurpose Internet Mail Extensions (MIME) security content types described in RFC1847, currently deployed in PGP 5.0 and later versions.

PKCS (Public Key Crypto Standards) a set of *de facto* standards for public key cryptography developed in cooperation with an informal consortium (Apple, DEC, Lotus, Microsoft, MIT, RSA, and Sun) that includes algorithm-specific and algorithm-independent implementation standards. Specifications defining message syntax and other protocols controlled by RSA Data Security Inc.

PKI (Public Key Infrastructure) a widely available and accessible certificate system for obtaining an entity’s public key with some degree of certainty that you have the “right” key and that it has not been revoked.

Plaintext normal, legible, un-encrypted, unsigned text.

Private key the secret portion of a key pair-used to sign and decrypt information. A user's private key should be kept secret, known only to the user.

Private keyring a set of one or more private keys, all of which belong to the owner of the private keyring.

Public key one of two keys in a key pair-used to encrypt information and verify signatures. A user's public key can be widely disseminated to colleagues or strangers. Knowing a person's public key does not help anyone discover the corresponding private key.

Public keyring a set of public keys. Your public keyring includes your own public key(s).

Public-key cryptography cryptography in which a public and private key pair is used, and no security is needed in the channel itself.

Random number an important aspect to many cryptosystems, and a necessary element in generating a unique key(s) that are unpredictable to an adversary. True random numbers are usually derived from analog sources, and usually involve the use of special hardware.

Revocation retraction of certification or authorization.

RFC (Request for Comment) an IETF document, either FYI (For Your Information) RFC sub-series that are overviews and introductory or STD RFC sub-series that identify specify Internet standards. Each RFC has an RFC number by which it is indexed and by which it can be retrieved (www.ietf.org).

RSA short for RSA Data Security, Inc.; or referring to the principals - Ron Rivest, Adi Shamir, and Len Adleman; or referring to the algorithm they invented. The RSA algorithm is used in public key cryptography and is based on the fact that it is easy to multiply two large prime numbers together, but hard to factor them out of the product.

secret sharing see Key Splitting.

secure channel a means of conveying information from one entity to another such that an adversary does not have the ability to reorder, delete, insert, or read (SSL, IPSec, whispering in someone's ear).

self-signed key a public key that has been signed by the corresponding private key for proof of ownership.

session key the secret (symmetric) key used to encrypt each set of data on a transaction basis. A different session key is used for each communication session.

sign to apply a signature.

signature a digital code created with a private key. Signatures allow authentication of information by the process of signature verification. When you sign a message or file, the PGP program uses your private key to create a digital code that is unique to both the contents of the message and your private key. Anyone can use your public key to verify your signature.

S/MIME (Secure Multipurpose Mail Extension) a proposed standard developed by Deming software and RSA Data Security for encrypting and/or authenticating MIME data. S/MIME defines a format for the MIME data, the algorithms that must be used for interoperability (RSA, RC2, SHA-1), and the additional operational concerns such as ANSI X.509 certificates and transport over the Internet.

SSL (Secure Socket Layer) developed by Netscape to provide security and privacy over the Internet. Supports server and client authentication and maintains the security and integrity of the transmission channel. Operates at the transport layer and mimics the “sockets library,” allowing it to be application independent. Encrypts the entire communication channel and does not support digital signatures at the message level.

symmetric algorithm a.k.a., conventional, secret key, and single key algorithms; the encryption and decryption key are either the same or can be calculated from one another. Two sub-categories exist - Block and Stream.

subkey a subkey is a Diffie-Hellman encryption key that is added as a subset to your master key. Once a subkey is created, you can expire or revoke it without affecting your master key or the signatures collected on it.

Text standard, printable, 7-bit ASCII text.

Timestamping recording the time of creation or existence of information.

TLS (Transport Layer Security) an IETF draft, version 1 is based on the Secure Sockets Layer (SSL) version 3.0 protocol, and provides communications privacy over the Internet.

TLSP (Transport Layer Security Protocol) ISO 10736, draft international standard.

Triple DES an encryption configuration in which the DES algorithm is used three times with three different keys.

Trusted a public key is said to be trusted by you if it has been validated by you or by someone you have designated as an introducer.

Trusted introducer someone whom you trust to provide you with keys that are valid. When a trusted introducer signs another person’s key, you trust that the person’s key is valid, and you do not need to verify the key before using it.

User ID a text phrase that identifies a key pair. For example, one common format for a user ID is the owner's name and email address. The user ID helps users (both the owner and colleagues) identify the owner of the key pair.

Validity indicates the level of confidence that the key actually belongs to the alleged owner.

Verification the act of comparing a signature created with a private key to its public key. Verification proves that the information was actually sent by the signer, and that the message has not been subsequently altered by anyone else.

VPN (Virtual Private Network) allows private networks to span from the end-user, across a public network (Internet) directly to the Home Gateway of choice, such as your company's Intranet.

Web of trust a distributed trust model used by PGP to validate the ownership of a public key where the level of trust is cumulative, based on the individuals' knowledge of the introducers.

X.509 an ITU-T digital certificate that is an internationally recognized electronic document used to prove identity and public key ownership over a communication network. It contains the issuer's name, the user's identifying information, and the issuer's digital signature, as well as other possible extensions.

Index