Using PGP

InterWorks 2001
Session # 24

Ben Rothke, CISSP, CCO
Senior Security Consultant
Baltimore Technologies
ben.rothke@baltimore.com

About me...

- Who I am
  - Senior Security Consultant with Baltimore Technologies
  - Previously with Ernst & Young, Citibank
- Who Baltimore Technologies is
  - Leading e-security company
  - A global leader in e-security products, services & solutions
  - Over 1,200 employees worldwide

Session agenda

- This session is about:
  - An introduction to encryption and cryptography
  - Main uses and features of PGP Desktop Security
    - Version 6.5.8 for Windows is used in this talk

- This session is not about:
  - A comprehensive investigation of every available PGP option and configuration setting
  - PGP Certificate Server, PGP Disk and other add-ons
  - Heavy mathematics and science of cryptography
  - Moral, legal, privacy, social and political issues

Session agenda

- Intro to PGP
- Cryptography & digital signatures
- Keys and key sizes
- Passphrases
- Installation & steps to using PGP
  - Key generation & distribution
  - Exporting your public key
  - Public key servers
  - Importing/exporting keys
  - File encryption/decryption
  - Digital signature signing/verification
  - Freespace & file wiping
  - Key Management - validity & trust
- Wrap-up / Q&A

What is PGP?

- Pretty Good Privacy is a software package that provides strong cryptographic functionality for e-mail, file, and disk storage
- Originally developed as freeware, PGP has since become the de facto standard for e-mail security
  - Has made cryptography accessible for the on-line community
  - NAI has commercial and freeware versions
    - Commercial - http://www.nai.com/asp_set/products/tns/intro.asp
    - Freeware - http://web.mit.edu/network/pgp.html
    - Source code - http://www.pgpi.org/products/pgp/versions/freeware/win32/6.5.8/
- Multiple platform support
  - Windows 95/98/NT/2000, Solaris, AIX, HP/UX, Linux, Solaris, MS-DOS, MacOS
- Provides message encryption, digital signatures, data compression, and transfer of secure e-mail

PGP History

- 1991 – v1.0 written by Phil Zimmerman ships. RSA files suit against Zimmerman for patent infringement
- 1992 – v2.0 ships. Bass-O-Matic replaced by IDEA
- 1993 – FBI investigates Zimmerman for possible violation of federal export laws, namely ITAR
- 1994 – v2.4 – ViaCrypt starts commercial distribution
- 1997 – v5.0 released by PGP Inc.
- 1997 – PGP Inc. acquired by Network Associates
- 1998 – v6.0 ships
- 1999 – PGP, Inc. rolled out as separate division of NAI
- 2000 – v7.0 ships
- 2000 – RSA patents expired on September 20, 2000
- 2001 – Phil Zimmerman leaves NAI for Hush Communications
- 2001 - Describe flaw found

What PGP does

- Encrypt files
- Send and receive encrypted e-mail
  - Microsoft Outlook 97/98/2000
  - Microsoft Outlook Express 4.x and 5.x
  - Lotus Notes 4.5.x, 4.6.x and 5.x
  - Qualcomm Eudora 4.x
  - Claris E-mailer 2.x for Macintosh
- Create secret and public key pairs
- Manage keys
- Certify keys
- Sign documents with a digital signature
- Verify documents signed with a digital signature

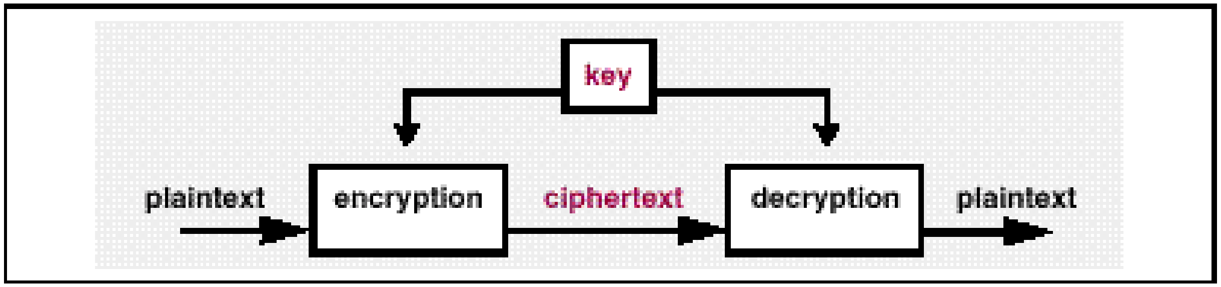
Cryptography

- Cryptography – science of using mathematics to encrypt and decrypt data
- Encryption – Conversion of data into a pattern, often called ciphertext, that can’t be read by unauthorized persons
- Decryption – Process of converting ciphertext data back into its original form, so it can be read

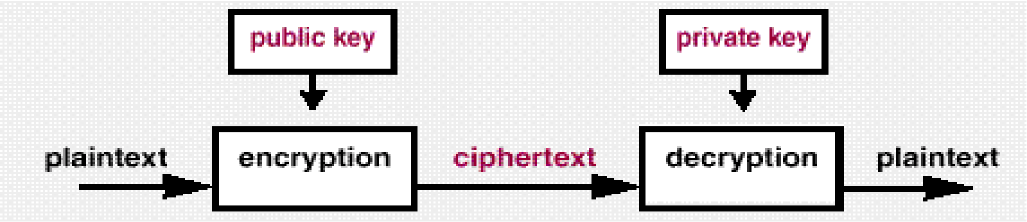
Public-key Cryptography

- 1976 - Conceptual ideas developed by Whitfield Diffie and Martin Hellman to solve key management problems
  - You need a secure channel to set up a secure channel
  - How do you get the key a a recipient without someone intercepting it?
- 1977 - First PKC designed by Ron Rivest, Adi Shamir & Len Adlelman (RSA)
- In a PKC, each user has a publicly known encryption key and a corresponding private key known only to that user
- When sending a message to someone, you encrypt the message with their *public* key. When they receive it, they decrypt it with their *private* key

Asymmetric vs. Symmetric cryptography



Secret-key encryption



Public-key encryption

Key management issues

- With symmetric cryptography, it is essentially impossible to provide effective key management for large networks.
- With symmetric cryptography, as the number of users increase, the number of keys required to provide secure communications among those users increases rapidly.
- For a group of *n* users, there needs to be 1/2 (n^2 - n) keys for total communications
- As the number of parties increases (i.e., *n* becomes larger), the number of symmetric keys becomes unreasonably large for practical use.
  - This is known as the *n^2 Problem*

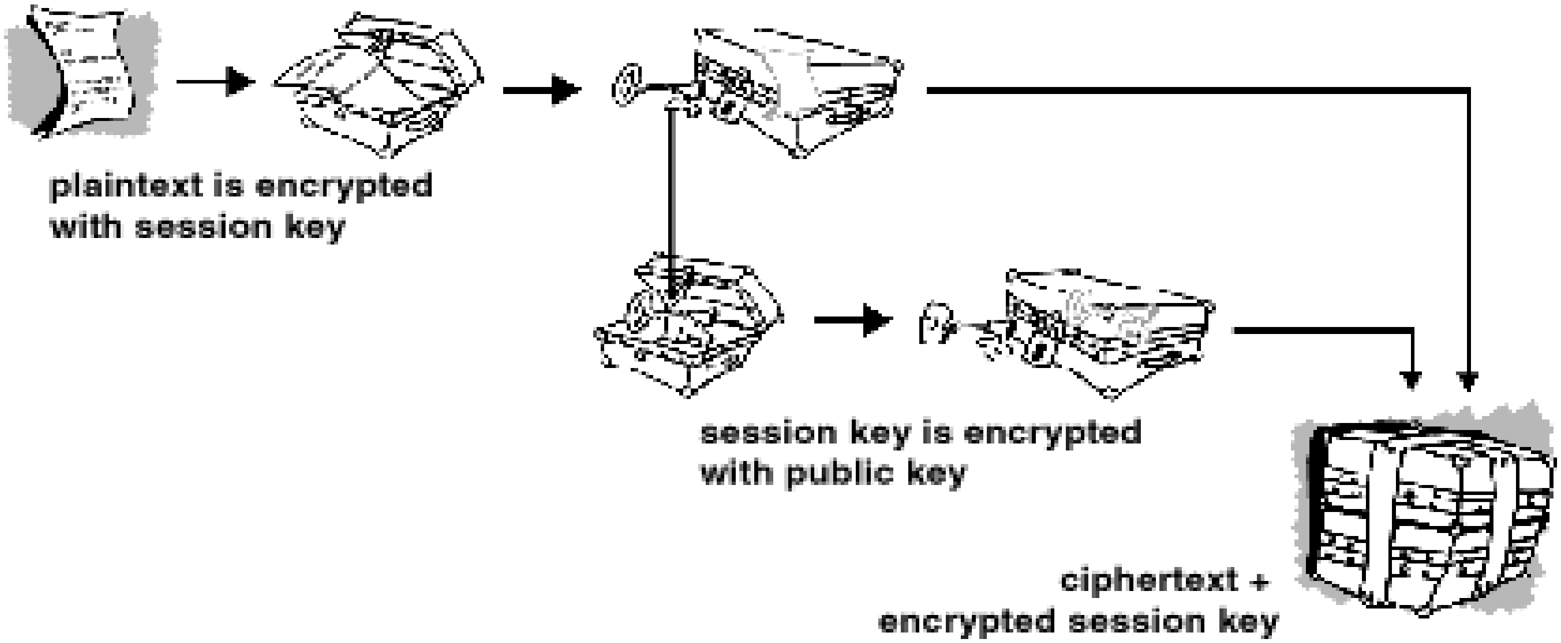
The n^2 Problem

Users	1/2 $(n^2 - n)$	Shared key pairs required
2	½ (4 - 2)	1
3	½ (9 – 3)	3
10	½ (100 – 10)	45
100	½ (10,000 – 100)	4,950
1000	½ (1,000,000 – 1,000)	499,500

PGP hybrid cryptography

- Encrypting an entire message can be extremely CPU intensive. PGP therefore uses both public and private-key cryptography.
- PGP first compresses the plaintext
- PGP then creates a session key, which is a one-time secret key. This key is a random number generated from the random movement of the mouse & keystrokes. The session key then encrypts the plaintext resulting in the ciphertext.
- Once the data is encrypted, the session key is then encrypted to the recipient's public key. The public-key encrypted session key is transmitted along with the ciphertext to the recipient
- Decryption works in the opposite way. The recipient uses their private key to recover the temporary session key, which PGP then uses to decrypt the conventionally encrypted ciphertext

PGP hybrid cryptography



Keys & key sizes

- *Key* – A value that works with a cryptographic algorithm to produce ciphertext
- Keys, measured in bits are basically huge numbers
  - PGP key sizes range from 1024 to 4096 bits
  - Default is 2048 bits
  - Too big a key, too time-consuming
  - Too small a key, too insecure
- Private key-size := Public-key size
  - 80-bit private-key == 1024-bit public-key
  - 128-bit private-key == 3000-bit public-key

Keys & key sizes

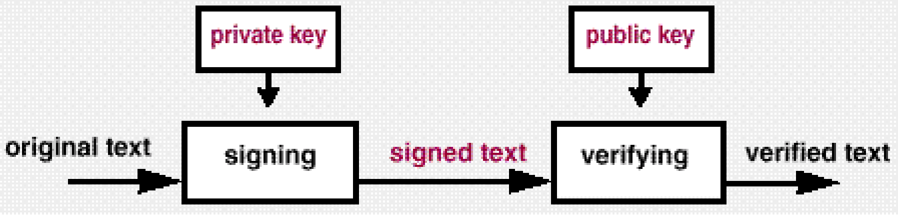
- Caveat: Key sizes are only one aspect of effective security
- Longer keys don’t always mean more security
  - Does a longer dead-bolt mean your house is more secure?
- Just as it’s possible to build a weak infrastructure using strong materials, it’s also possible to build a weak cryptographic system using long keys and strong algorithms and protocols.

How secure is PGP?

- If configured correctly and a good passphrase is utilized – very secure.
- Brute-force key search – IDEA encryption uses 128-bit keys for 2^{128} possible combinations.
- If a special purpose chip (FPGA) could perform a billion decryptions per second, and the server had a billion chips running in parallel, it would still require over 10^{12} years to try all of the possible keys, which is about a thousand times the age of the universe.

Digital Signatures

- Used to authenticate the identify of the message sender or the signer of a document and to ensure that the original content of the message or document has not been altered.
- PGP uses an algorithm that generates a hash code from the user's name and other signature information. The hash code is then encrypted with the sender's private-key. The receiver uses the sender's public-key to decrypt the hash code. If it matches the hash code, then the receiver is sure that the message has arrived securely from the stated sender.



Passphrase

- PGP security is built on the premise of a strong passphrase.
  - Your protection is ultimately only as good as the strength of your passphrase
  - Passphrase should include a combination of upper and lowercase alphabetic letters, numbers, punctuation marks and spaces.
  - Don’t use an easy to guess passphrase (DOB, SSN, etc.)
  - Backup your secret-key and store it in a secure location
- **DON’T FORGET YOUR PASSPHRASE!!!!**
  - If you do, you are totally, completely, utterly, absolutely, unconditionally, entirely, thoroughly and fully out of luck.

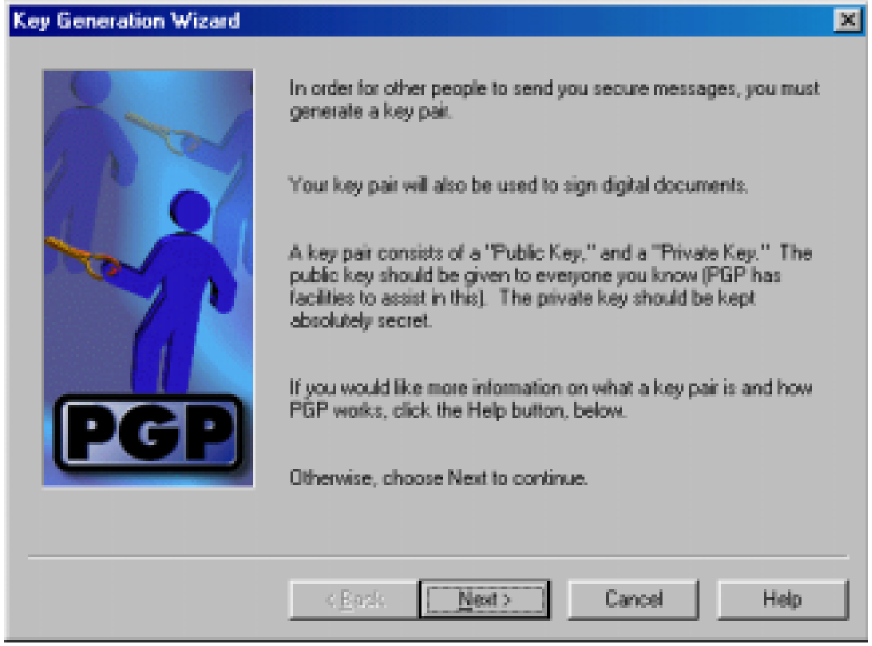
Steps to using PGP

- Installation
- Key generation
  - Your private and public keys
- Key exchange with others
- Public key validation for keys exchanged
- Encrypt files or e-mail
- Sign documents
- Decrypt & verify files
- Wipe files for permanent deletion

PGP Installation - Windows

- Standard Windows installation
- Run SETUP.EXE
- Choose directory location
- Select components
- At completion, Key Generation wizard starts
- Backup your keys

Key Generation



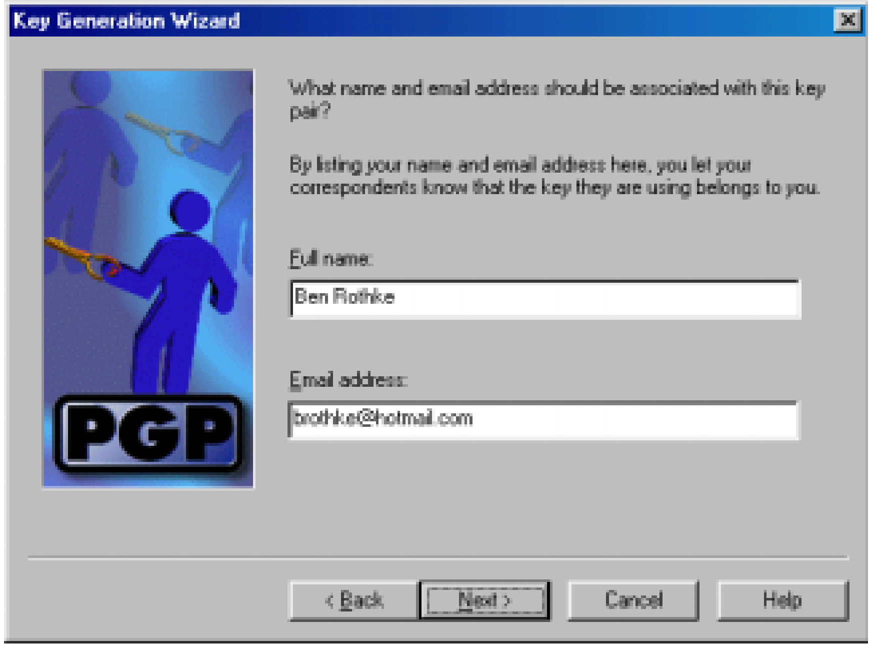
In order for other people to send you secure messages, you must generate a key pair.

Your key pair will also be used to sign digital documents.

A key pair consists of a "Public Key," and a "Private Key." The public key should be given to everyone you know (PGP has facilities to assist in this). The private key should be kept absolutely secret.

If you would like more information on what a key pair is and how PGP works, click the Help button, below.

Otherwise, choose Next to continue.



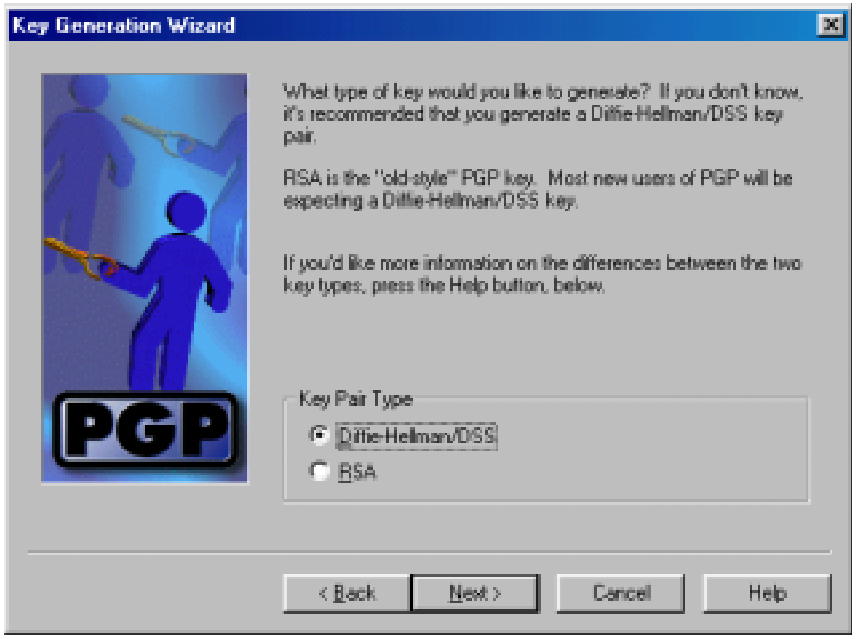
What name and email address should be associated with this key pair?

By listing your name and email address here, you let your correspondents know that the key they are using belongs to you.

Full name: Ben Rothke

Email address: brothke@hotmail.com

Key Generation



What type of key would you like to generate? If you don't know, it's recommended that you generate a Diffie-Hellman/DSS key pair.

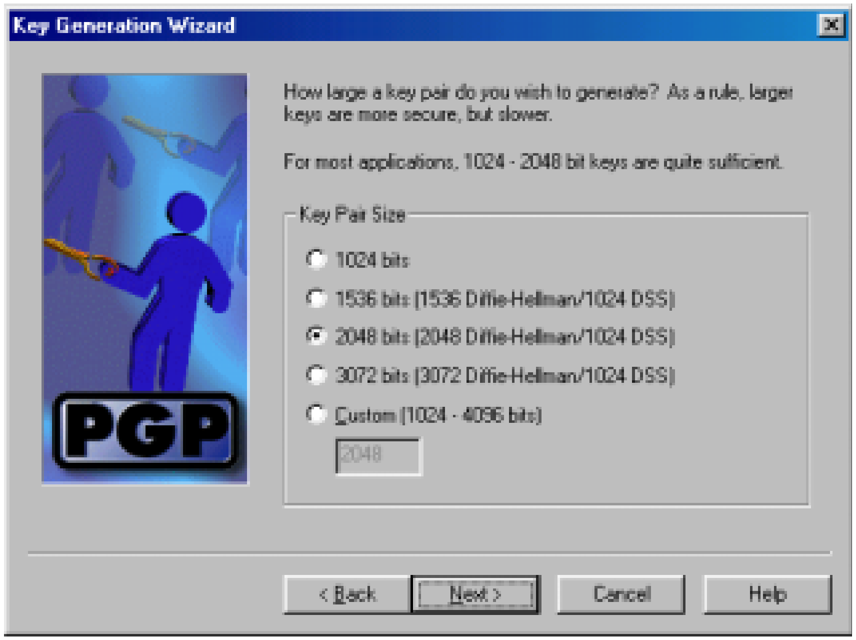
RSA is the "old-style" PGP key. Most new users of PGP will be expecting a Diffie-Hellman/DSS key.

If you'd like more information on the differences between the two key types, press the Help button, below.

Key Pair Type

● Diffie-Hellman/DSS

○ RSA



How large a key pair do you wish to generate? As a rule, larger keys are more secure, but slower.

For most applications, 1024 - 2048 bit keys are quite sufficient.

Key Pair Size

○ 1024 bits

○ 1536 bits (1536 Diffie-Hellman/1024 DSS)

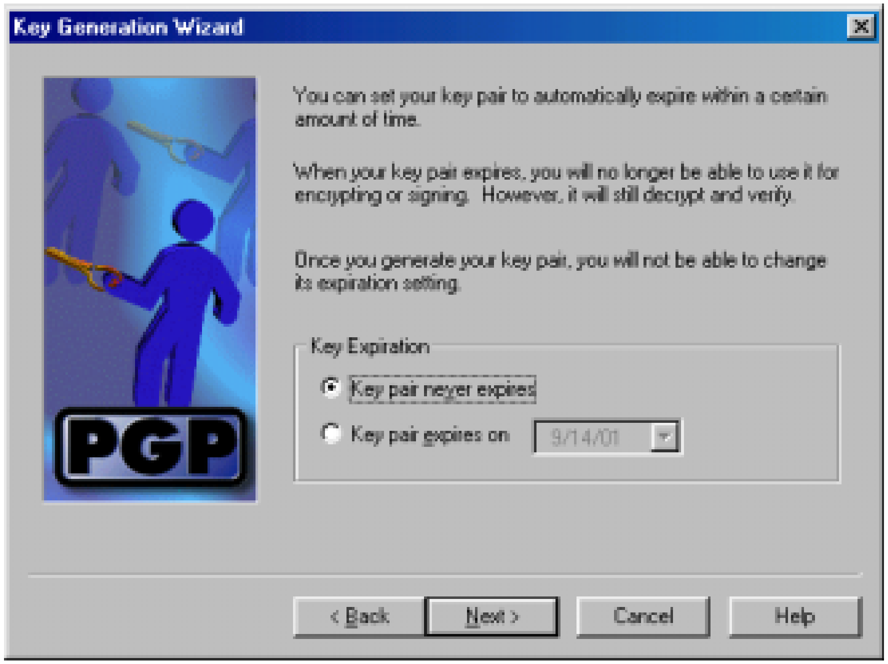
● 2048 bits (2048 Diffie-Hellman/1024 DSS)

○ 3072 bits (3072 Diffie-Hellman/1024 DSS)

○ Custom (1024 - 4096 bits)

2048

Key Generation



Key Generation Wizard

You can set your key pair to automatically expire within a certain amount of time.

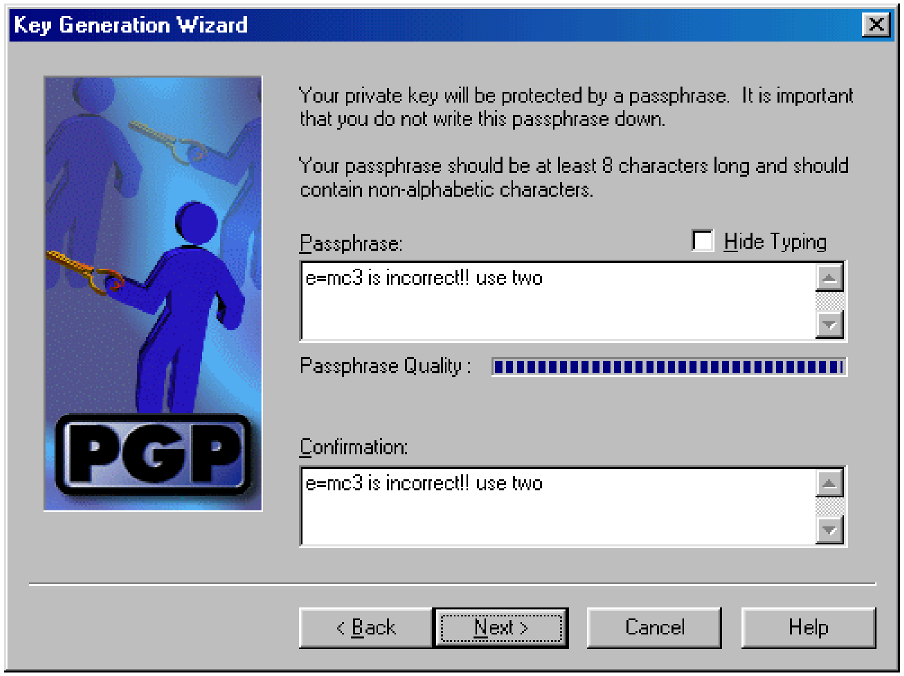
When your key pair expires, you will no longer be able to use it for encrypting or signing. However, it will still decrypt and verify.

Once you generate your key pair, you will not be able to change its expiration setting.

Key Expiration

● Key pair never expires

○ Key pair expires on 9/14/01



Key Generation Wizard

Your private key will be protected by a passphrase. It is important that you do not write this passphrase down.

Your passphrase should be at least 8 characters long and should contain non-alphabetic characters.

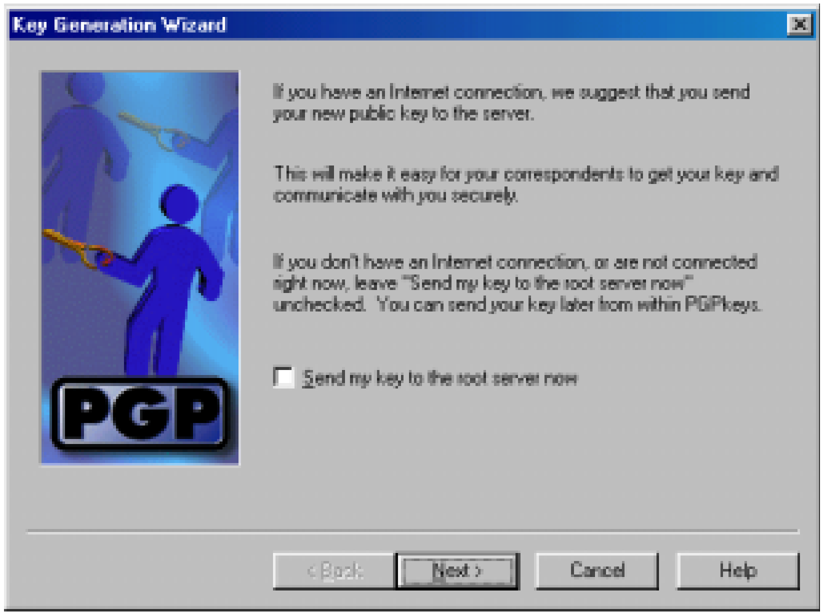
Passphrase: e=mc3 is incorrect!! use two

□ Hide Typing

Passphrase Quality :

Confirmation: e=mc3 is incorrect!! use two

Key Generation

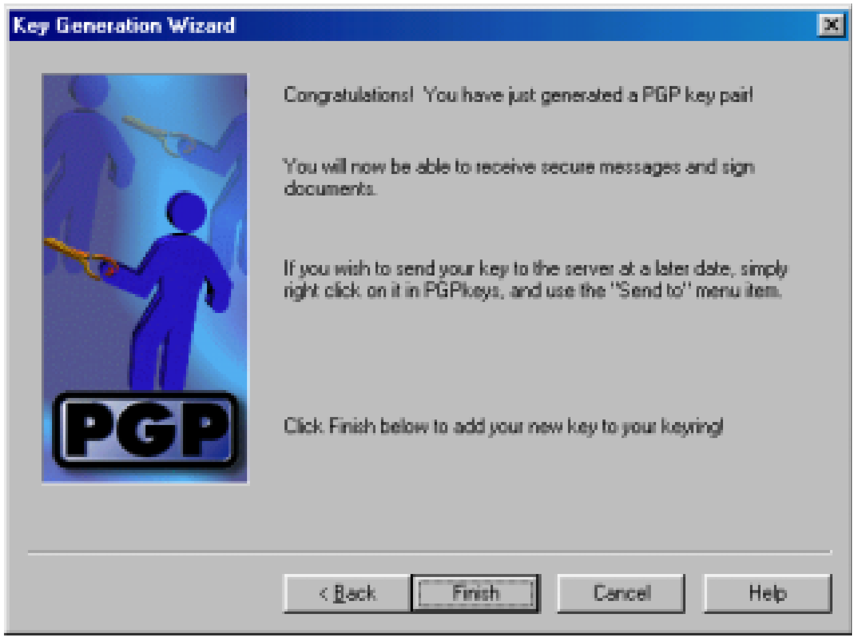


If you have an Internet connection, we suggest that you send your new public key to the server.

This will make it easy for your correspondents to get your key and communicate with you securely.

If you don't have an Internet connection, or are not connected right now, leave "Send my key to the root server now" unchecked. You can send your key later from within PGPkeys.

□ Send my key to the root server now



Congratulations! You have just generated a PGP key pair!

You will now be able to receive secure messages and sign documents.

If you wish to send your key to the server at a later date, simply right click on it in PGPkeys, and use the "Send to" menu item.

Click Finish below to add your new key to your keyring!

Key Distribution

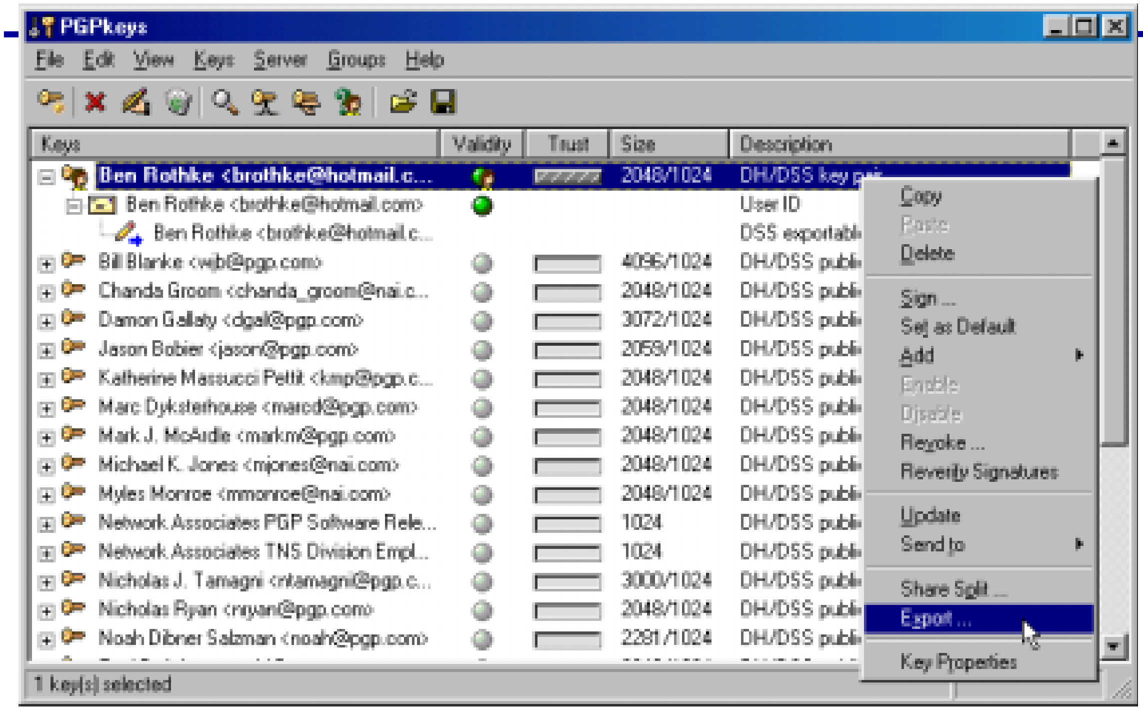
Making your public key available:

- Post to a public key server
- Export to a file
- Attach to your e-mail signature

Obtaining someone else's public key:

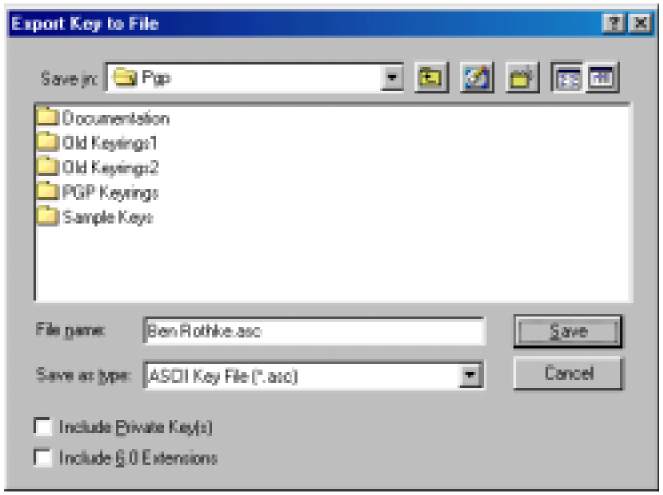
- Get the key from a public key server
  - And then add their key to your key ring
- Import the key from a file
- Copy it from their e-mail signature

Exporting your public key



Keys	Validity	Trust	Size	Description
Ben Rothke <brothke@hotmail.c...	●	▨	2048/1024	DH/DSS key pai
Ben Rothke <brothke@hotmail.com>	●			User ID
Ben Rothke <brothke@hotmail.c...				DSS exportabl
Bill Blanke <wjb@pgp.com>	●	▭	4096/1024	DH/DSS publi
Chanda Groom <chanda_groom@nai.c...	●	▭	2048/1024	DH/DSS publi
Damon Gallaty <dgal@pgp.com>	●	▭	3072/1024	DH/DSS publi
Jason Bobier <jason@pgp.com>	●	▭	2059/1024	DH/DSS publi
Katherine Massucci Pettit <kmp@pgp.c...	●	▭	2048/1024	DH/DSS publi
Marc Dyksterhouse <marcd@pgp.com>	●	▭	2048/1024	DH/DSS publi
Mark J. McArdle <markm@pgp.com>	●	▭	2048/1024	DH/DSS publi
Michael K. Jones <mjones@nai.com>	●	▭	2048/1024	DH/DSS publi
Myles Monroe <mmonroe@nai.com>	●	▭	2048/1024	DH/DSS publi
Network Associates PGP Software Rele...	●	▭	1024	DH/DSS publi
Network Associates TNS Division Empl...	●	▭	1024	DH/DSS publi
Nicholas J. Tamagni <ntamagni@pgp.c...	●	▭	3000/1024	DH/DSS publi
Nicholas Ryan <nryan@pgp.com>	●	▭	2048/1024	DH/DSS publi
Noah Dibner Salzman <noah@pgp.com>	●	▭	2281/1024	DH/DSS publi

1 key(s) selected



Save in: Pgp

Documentation
Old Keyrings1
Old Keyrings2
PGP Keyrings
Sample Keys

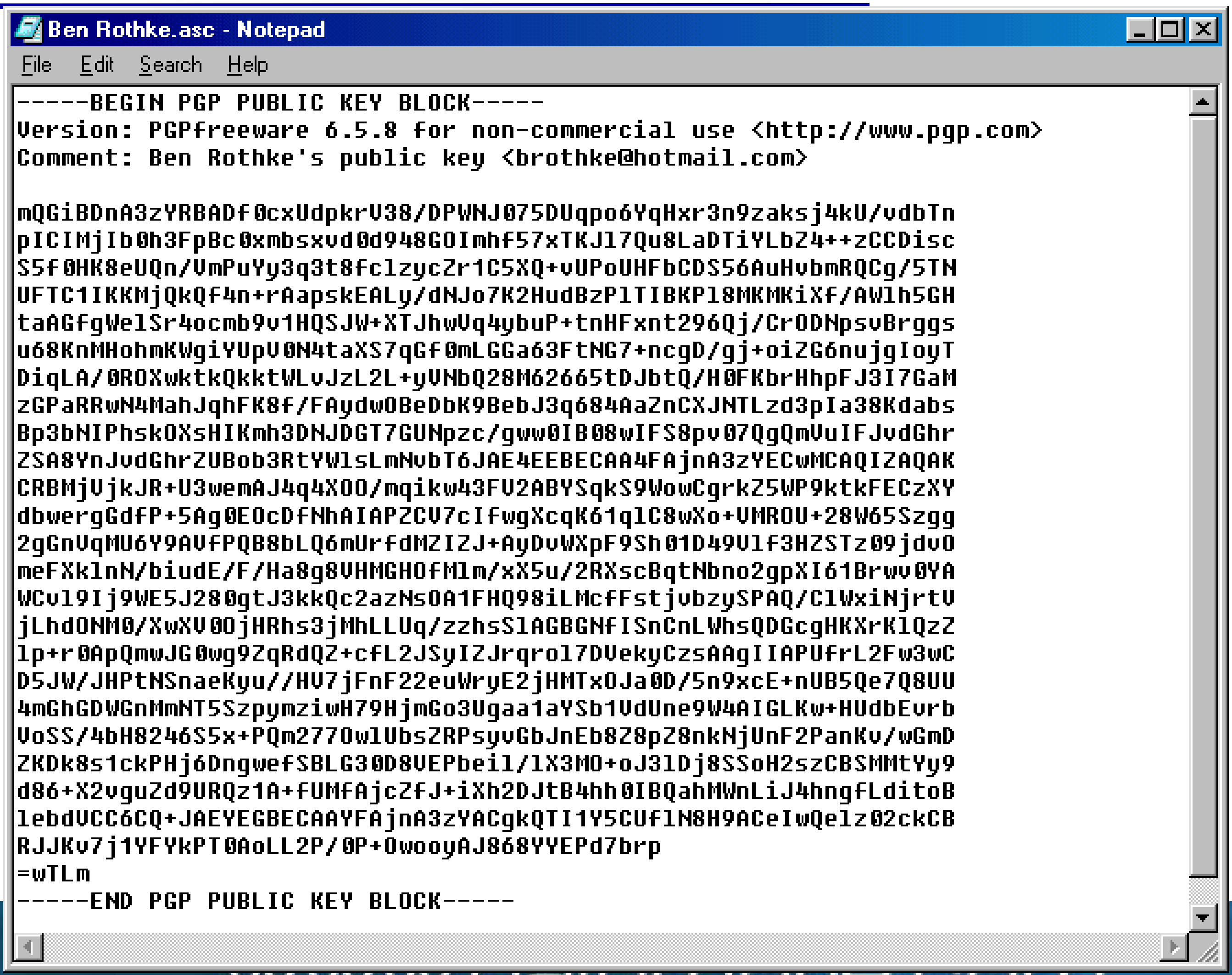
File name: Ben Rothke.asc

Save as type: ASCII Key File (*.asc)

□ Include Private Key(s)

□ Include 6.0 Extensions

Exporting your public key



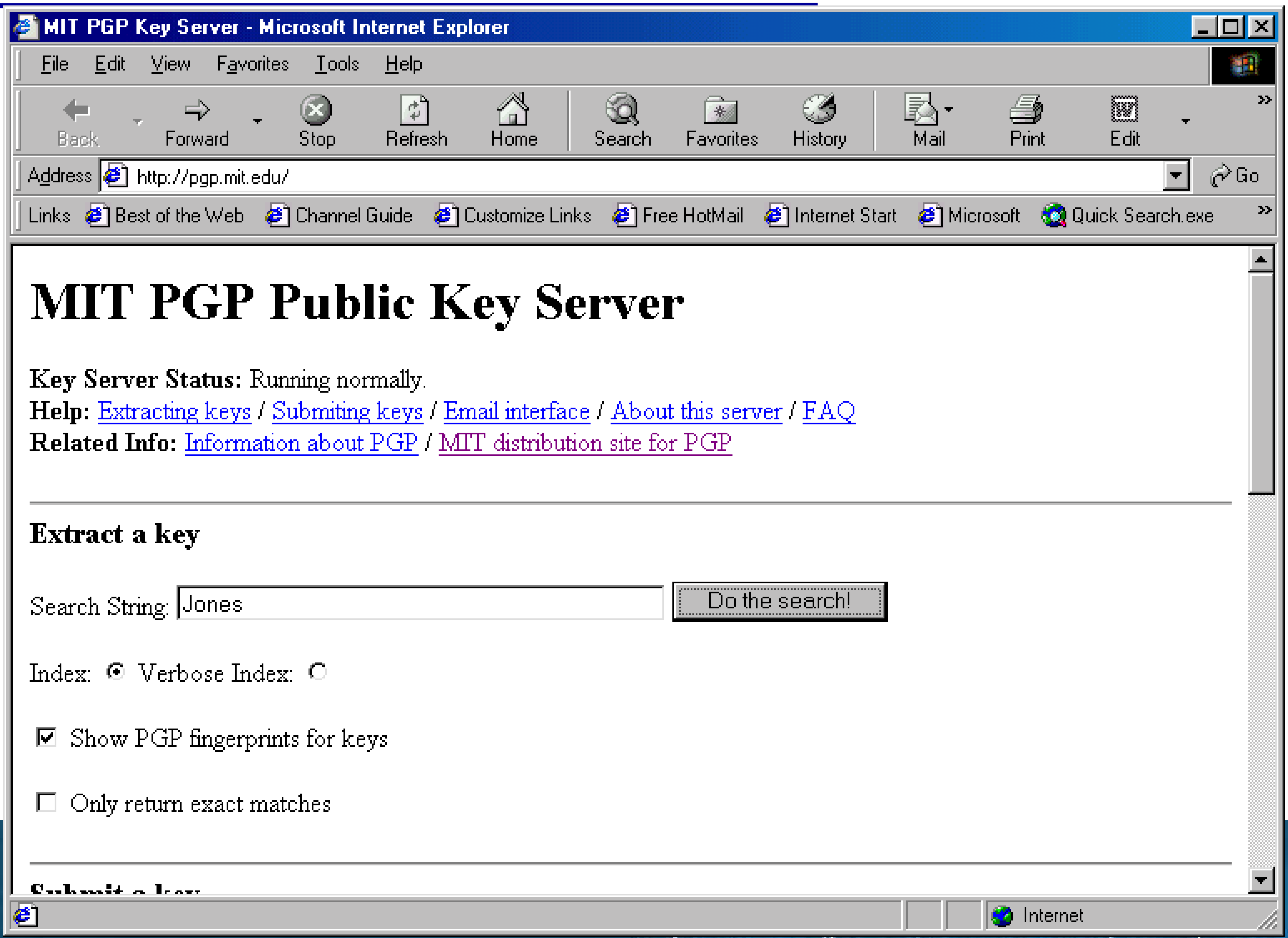
```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 6.5.8 for non-commercial use <http://www.pgp.com>
Comment: Ben Rothke's public key <brothke@hotmail.com>

mQGiBDnA3zYRBADf0cxUdpkrV38/DPWNJ075DUqpo6YqHxr3n9zaksj4kU/vdbTn
pICIMjIb0h3FpBc0xmbsxvd0d948GOImhf57xTKJ17Qu8LaDTiYLbZ4++zCCDisc
S5f0HK8eUQn/VmPuYy3q3t8fclzycZr1C5XQ+vUPoUHFbCDS56AuHvbmRQCg/5TN
UFTC1IKKMjQkQf4n+rAapskEALy/dNJo7K2HudBzPlTIBKPl8MKMKiXf/AWlh5GH
taAGfgWelSr4ocmb9v1HQSJW+XTJhwVq4ybuP+tnHFxnt296Qj/CrODNpsvBrggs
u68KnMHohmKWgiYUpV0N4taXS7qGf0mLGGa63FtNG7+ncgD/gj+oiZG6nujgIoyT
DiqLA/0ROXwktkQkktWLvJzL2L+yVNbQ28M62665tDJbtQ/H0FKbrHhpFJ3I7GaM
zGPaRRwN4MahJqhFK8f/FAydwOBeDbK9BebJ3q684AaZnCXJNTLzd3pIa38Kdabs
Bp3bNIPhskOXsHIKmh3DNJDGT7GUNpzc/gww0IB08wIFS8pv07QgQmVuIFJvdGhr
ZSA8YnJvdGhrZUBob3RtYWlsLmNvbT6JAE4EEBECAA4FAjnA3zYECwMCAQIZAQAK
CRBMjVjkJR+U3wemAJ4q4XOO/mqikw43FV2ABYSqkS9WowCgrkZ5WP9ktkFECzXY
dbwergGdfP+5Ag0EOcDfNhAIAPZCV7cIfwgXcqK61qlC8wXo+VMROU+28W65Szgg
2gGnVqMU6Y9AVfPQB8bLQ6mUrfdMZIZJ+AyDvWXpF9Sh01D49Vlf3HZSTz09jdvO
meFXklnN/biudE/F/Ha8g8VHMGHOfMlm/xX5u/2RXscBqtNbno2gpXI61Brwv0YA
WCvl9Ij9WE5J280gtJ3kkQc2azNsOA1FHQ98iLMcfFstjvbzySPAQ/ClWxiNjrtV
jLhdONM0/XwXV0OjHRhs3jMhLLUq/zzhsSlAGBGNfISnCnLWhsQDGcgHKXrKlQzZ
lp+r0ApQmwJG0wg9ZqRdQZ+cfL2JSyIZJrqrol7DVekyCzsAAgIIAPUfrL2Fw3wC
D5JW/JHPtNSnaeKyu//HV7jFnF22euWryE2jHMTxOJa0D/5n9xcE+nUB5Qe7Q8UU
4mGhGDWGnMmNT5SzpymziwH79HjmGo3Ugaa1aYSb1VdUne9W4AIGLKw+HUdbEvrb
VoSS/4bH8246S5x+PQm277OwlUbsZRPsyvGbJnEb8Z8pZ8nkNjUnF2PanKv/wGmD
ZKDk8s1ckPHj6DngwefSBLG30D8VEPbeil/lX3MO+oJ3lDj8SSoH2szCBSMMtYy9
d86+X2vguZd9URQz1A+fUMfAjcZfJ+iXh2DJtB4hh0IBQahMWnLiJ4hngfLditoB
lebdVCC6CQ+JAEYEGBECAAYFAjnA3zYACgkQTI1Y5CUflN8H9ACeIwQelz02ckCB
RJJKv7j1YFYkPT0AoLL2P/0P+OwooyAJ868VYEPd7brp
=wTLm
-----END PGP PUBLIC KEY BLOCK-----
```

Key servers

- Public-key servers allow keys to reside on a common public server for downloading
- A comprehensive list of key servers can be found at
  - www.hal-pc.org/~bunbytes/karlsson/pgp/index.html#keyserver
- NAI key server
  - http://certserver.pgp.com
- Brian LaMacchia's public key server

  http://pgp.mit.edu/

Key servers



Address: http://pgp.mit.edu/

MIT PGP Public Key Server

Key Server Status: Running normally.
Help: Extracting keys / Submiting keys / Email interface / About this server / FAQ
Related Info: Information about PGP / MIT distribution site for PGP

Extract a key

Search String: Jones [Do the search!]

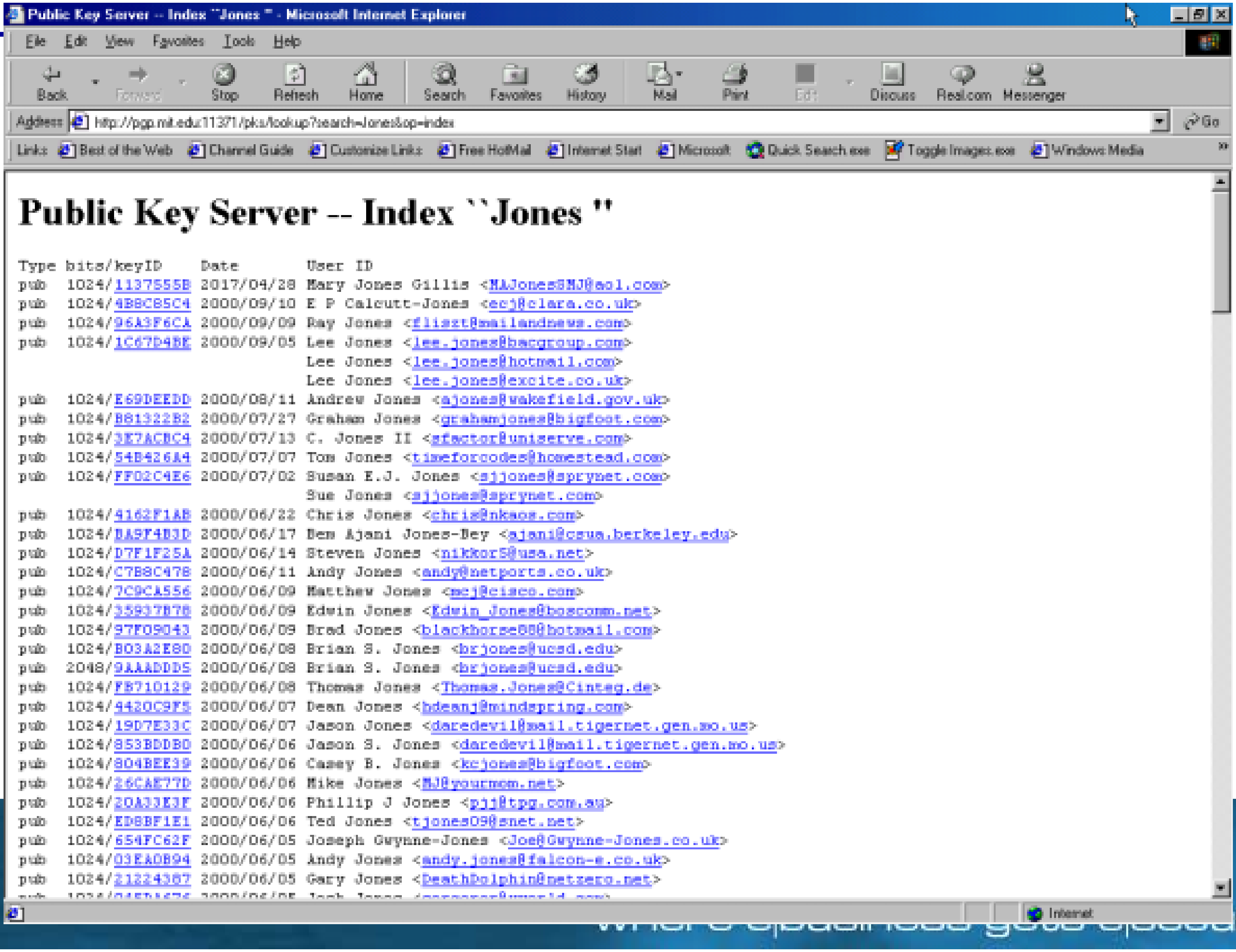
Index: ● Verbose Index: ○

☑ Show PGP fingerprints for keys

□ Only return exact matches

Submit a key

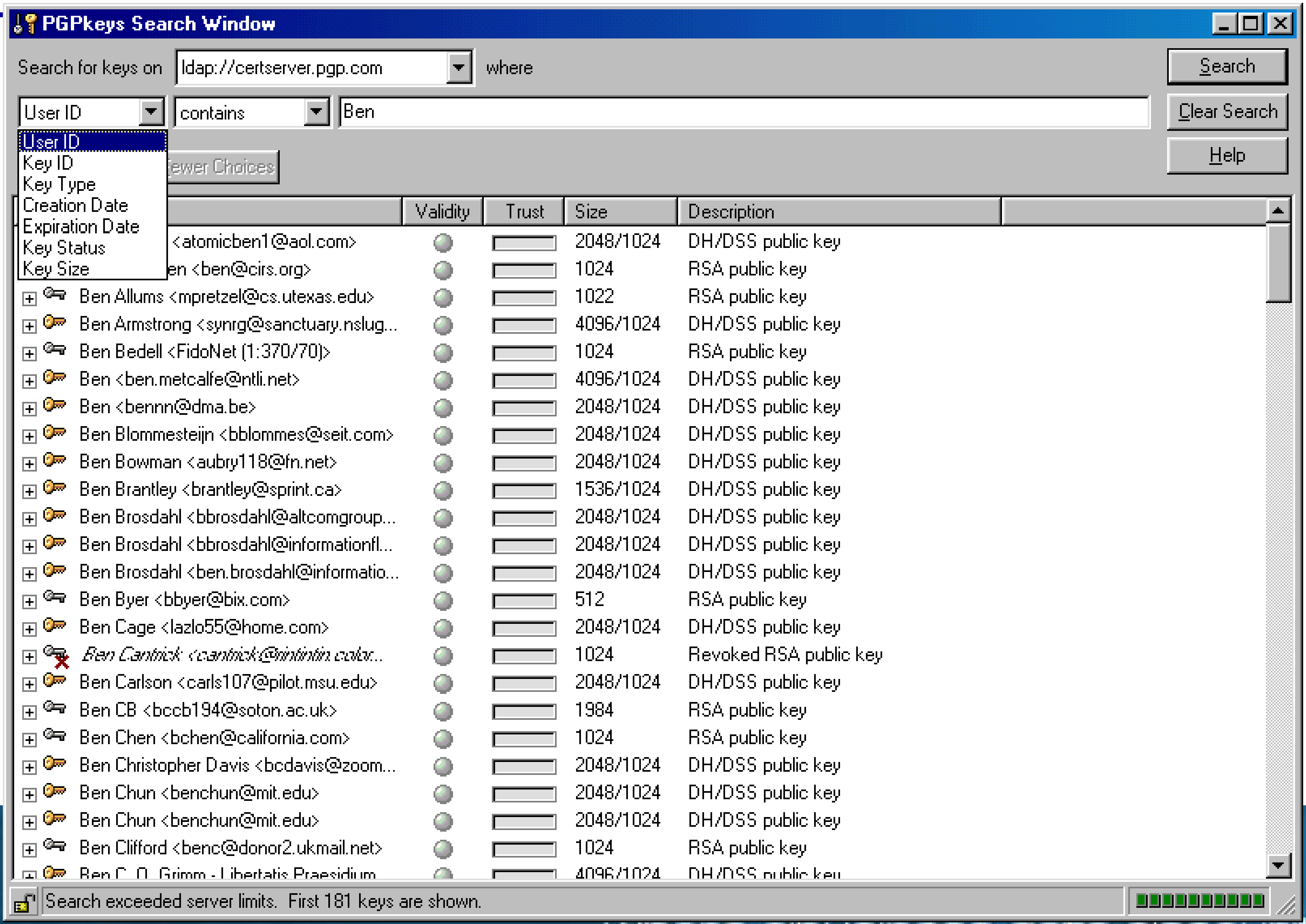
Key servers


Public Key Server -- Index ``Jones '' - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Real.com Messenger
Address http://pgp.mit.edu:11371/pks/lookup?search=Jones&op=index
Go
Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft Quick Search.exe Toggle Images.exe Windows Media
Internet

Public Key Server -- Index ``Jones ''

```
Type bits/keyID    Date       User ID
pub  1024/1137555B 2017/04/28 Mary Jones Gillis <MAJonesSMJ@aol.com>
pub  1024/4B8C85C4 2000/09/10 E P Calcutt-Jones <ecj@clara.co.uk>
pub  1024/96A3F6CA 2000/09/09 Ray Jones <fliszt@mailandnews.com>
pub  1024/1C67D4BE 2000/09/05 Lee Jones <lee.jones@bacgroup.com>
                              Lee Jones <lee.jones@hotmail.com>
                              Lee Jones <lee.jones@excite.co.uk>
pub  1024/E69DEEDD 2000/08/11 Andrew Jones <ajones@wakefield.gov.uk>
pub  1024/B81322B2 2000/07/27 Graham Jones <grahamjones@bigfoot.com>
pub  1024/3E7ACBC4 2000/07/13 C. Jones II <sfactor@uniserve.com>
pub  1024/54B426A4 2000/07/07 Tom Jones <timeforcodes@homestead.com>
pub  1024/FF02C4E6 2000/07/02 Susan E.J. Jones <sjjones@sprynet.com>
                              Sue Jones <sjjones@sprynet.com>
pub  1024/4162F1AB 2000/06/22 Chris Jones <chris@nkaos.com>
pub  1024/BA9F4B3D 2000/06/17 Ben Ajani Jones-Bey <ajani@csua.berkeley.edu>
pub  1024/D7F1F25A 2000/06/14 Steven Jones <nikkor5@usa.net>
pub  1024/C7B8C478 2000/06/11 Andy Jones <andy@netports.co.uk>
pub  1024/7C9CA556 2000/06/09 Matthew Jones <mcj@cisco.com>
pub  1024/35937878 2000/06/09 Edwin Jones <Edwin_Jones@boscomm.net>
pub  1024/97F09043 2000/06/09 Brad Jones <blackhorse88@hotmail.com>
pub  1024/B03A2E8D 2000/06/08 Brian S. Jones <brjones@ucsd.edu>
pub  2048/9AAADDD5 2000/06/08 Brian S. Jones <brjones@ucsd.edu>
pub  1024/FB710129 2000/06/08 Thomas Jones <Thomas.Jones@Cinteg.de>
pub  1024/4420C9F5 2000/06/07 Dean Jones <hdeanj@mindspring.com>
pub  1024/19D7E33C 2000/06/07 Jason Jones <daredevil@mail.tigernet.gen.mo.us>
pub  1024/853BDDBD 2000/06/06 Jason S. Jones <daredevil@mail.tigernet.gen.mo.us>
pub  1024/804BEE39 2000/06/06 Casey B. Jones <kcjones@bigfoot.com>
pub  1024/26CAE77D 2000/06/06 Mike Jones <MJ@yourmom.net>
pub  1024/20A33E3F 2000/06/06 Phillip J Jones <pjj@tpg.com.au>
pub  1024/ED8BF1E1 2000/06/06 Ted Jones <tjones09@snet.net>
pub  1024/654FC62F 2000/06/05 Joseph Gwynne-Jones <Joe@Gwynne-Jones.co.uk>
pub  1024/03EA0B94 2000/06/05 Andy Jones <andy.jones@falcon-e.co.uk>
pub  1024/21224387 2000/06/05 Gary Jones <DeathDolphin@netzero.net>
```

Key servers



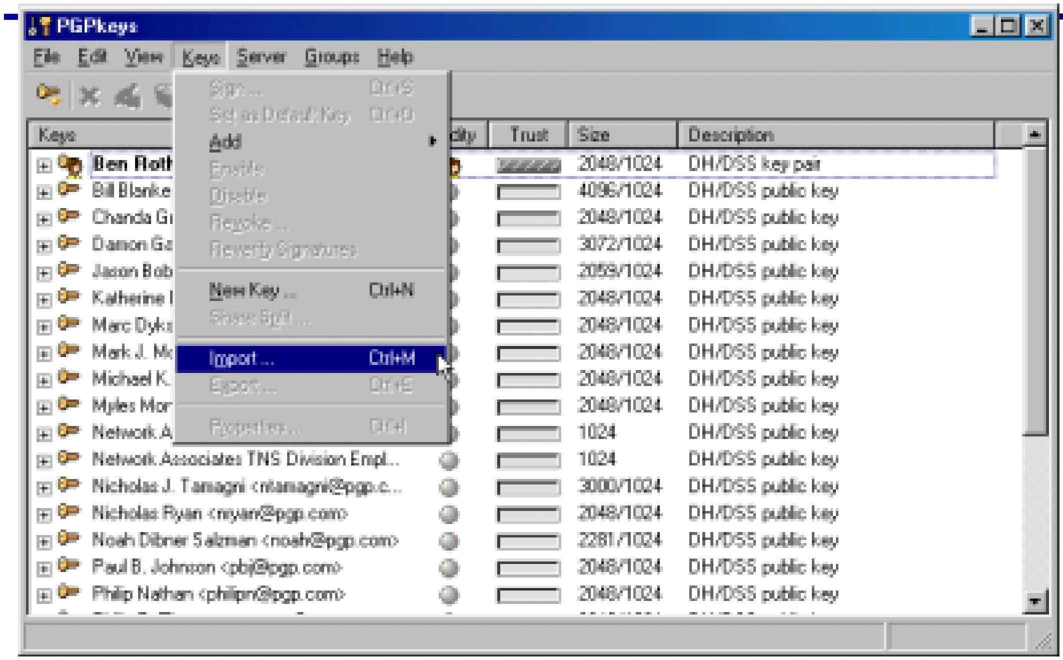
Search for keys on ldap://certserver.pgp.com where

User ID contains Ben

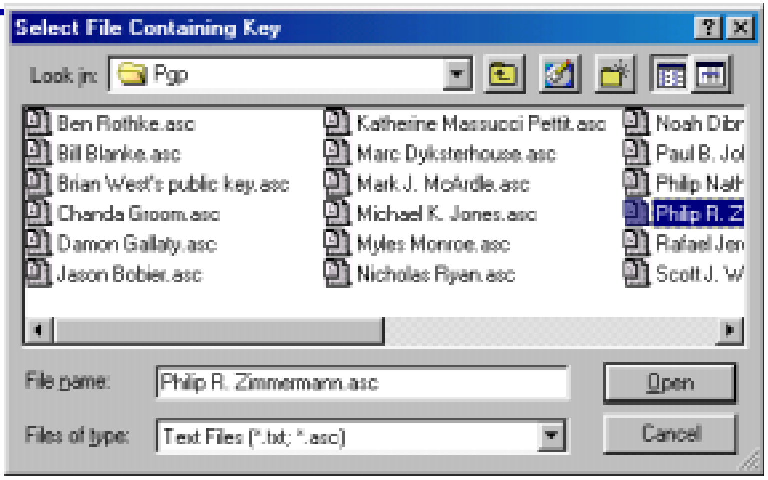
	Validity	Trust	Size	Description
<atomicben1@aol.com>	●	▭	2048/1024	DH/DSS public key
en <ben@cirs.org>	●	▭	1024	RSA public key
Ben Allums <mpretzel@cs.utexas.edu>	●	▭	1022	RSA public key
Ben Armstrong <synrg@sanctuary.nslug...	●	▭	4096/1024	DH/DSS public key
Ben Bedell <FidoNet (1:370/70)>	●	▭	1024	RSA public key
Ben <ben.metcalfe@ntli.net>	●	▭	4096/1024	DH/DSS public key
Ben <bennn@dma.be>	●	▭	2048/1024	DH/DSS public key
Ben Blommesteijn <bblommes@seit.com>	●	▭	2048/1024	DH/DSS public key
Ben Bowman <aubry118@fn.net>	●	▭	2048/1024	DH/DSS public key
Ben Brantley <brantley@sprint.ca>	●	▭	1536/1024	DH/DSS public key
Ben Brosdahl <bbrosdahl@altcomgroup...	●	▭	2048/1024	DH/DSS public key
Ben Brosdahl <bbrosdahl@informationfl...	●	▭	2048/1024	DH/DSS public key
Ben Brosdahl <ben.brosdahl@informatio...	●	▭	2048/1024	DH/DSS public key
Ben Byer <bbyer@bix.com>	●	▭	512	RSA public key
Ben Cage <lazlo55@home.com>	●	▭	2048/1024	DH/DSS public key
Ben Cantrick <cantrick@intintin.colo...	●	▭	1024	Revoked RSA public key
Ben Carlson <carls107@pilot.msu.edu>	●	▭	2048/1024	DH/DSS public key
Ben CB <bccb194@soton.ac.uk>	●	▭	1984	RSA public key
Ben Chen <bchen@california.com>	●	▭	1024	RSA public key
Ben Christopher Davis <bcdavis@zoom...	●	▭	2048/1024	DH/DSS public key
Ben Chun <benchun@mit.edu>	●	▭	2048/1024	DH/DSS public key
Ben Chun <benchun@mit.edu>	●	▭	2048/1024	DH/DSS public key
Ben Clifford <benc@donor2.ukmail.net>	●	▭	1024	RSA public key
Ben C. O. Grimm - Libertatis Praesidium	●	▭	4096/1024	DH/DSS public key

Search exceeded server limits. First 181 keys are shown.

Importing a public key



Keys	Validity	Trust	Size	Description
Ben Roth			2048/1024	DH/DSS key pair
Bill Blanke			4096/1024	DH/DSS public key
Chanda Gr			2048/1024	DH/DSS public key
Damon Ga			3072/1024	DH/DSS public key
Jason Bob			2059/1024	DH/DSS public key
Katherine I			2048/1024	DH/DSS public key
Marc Dyk			2048/1024	DH/DSS public key
Mark J. Mc			2048/1024	DH/DSS public key
Michael K.			2048/1024	DH/DSS public key
Myles Mor			2048/1024	DH/DSS public key
Network A			1024	DH/DSS public key
Network Associates TNS Division Empl...	●		1024	DH/DSS public key
Nicholas J. Tamagni <ntamagni@pgp.c...	●		3000/1024	DH/DSS public key
Nicholas Ryan <nryan@pgp.com>	●		2048/1024	DH/DSS public key
Noah Dibner Salzman <noah@pgp.com>	●		2281/1024	DH/DSS public key
Paul B. Johnson <pbj@pgp.com>	●		2048/1024	DH/DSS public key
Philip Nathan <philipn@pgp.com>	●		2048/1024	DH/DSS public key

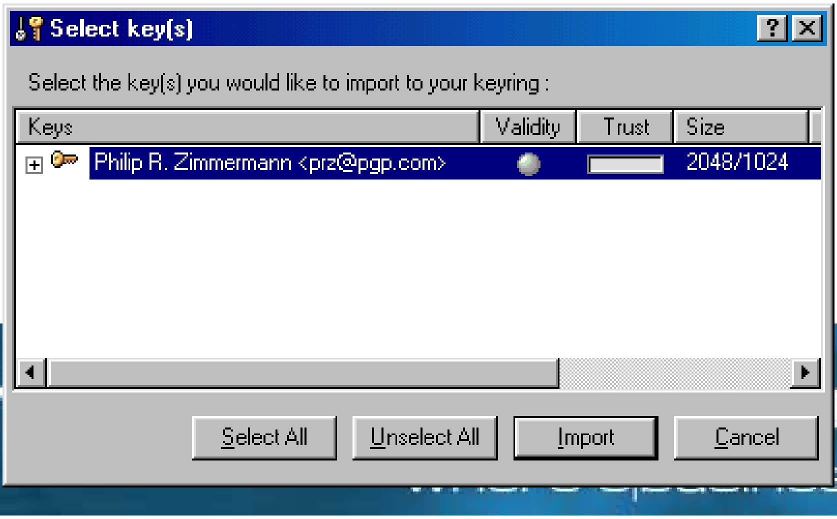


Look in: Pgp

Ben Rothke.asc
Bill Blanke.asc
Brian West's public key.asc
Chanda Groom.asc
Damon Gallaty.asc
Jason Bobier.asc
Katherine Massucci Pettit.asc
Marc Dykstterhouse.asc
Mark J. McArdle.asc
Michael K. Jones.asc
Myles Monroe.asc
Nicholas Ryan.asc
Noah Dibr
Paul B. Jol
Philip Nath
Philip R. Z
Rafael Jen
Scott J. W

File name: Philip R. Zimmermann.asc

Files of type: Text Files (*.txt; *.asc)

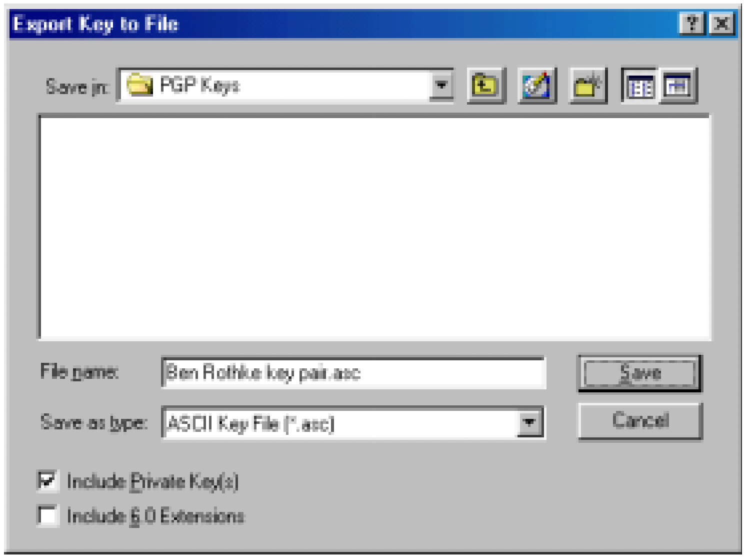


Select the key(s) you would like to import to your keyring :

Keys	Validity	Trust	Size
Philip R. Zimmermann <prz@pgp.com>	●	▭	2048/1024

Backing up your key pair

- Export your own key pair and select *Include Private Keys* checkbox.
- If you lose your private key, there is no way to recreate it, even if you use the same passphrase



Export Key to File

Save in: PGP Keys

File name: Ben Rothke key pair.asc

Save as type: ASCII Key File (*.asc)

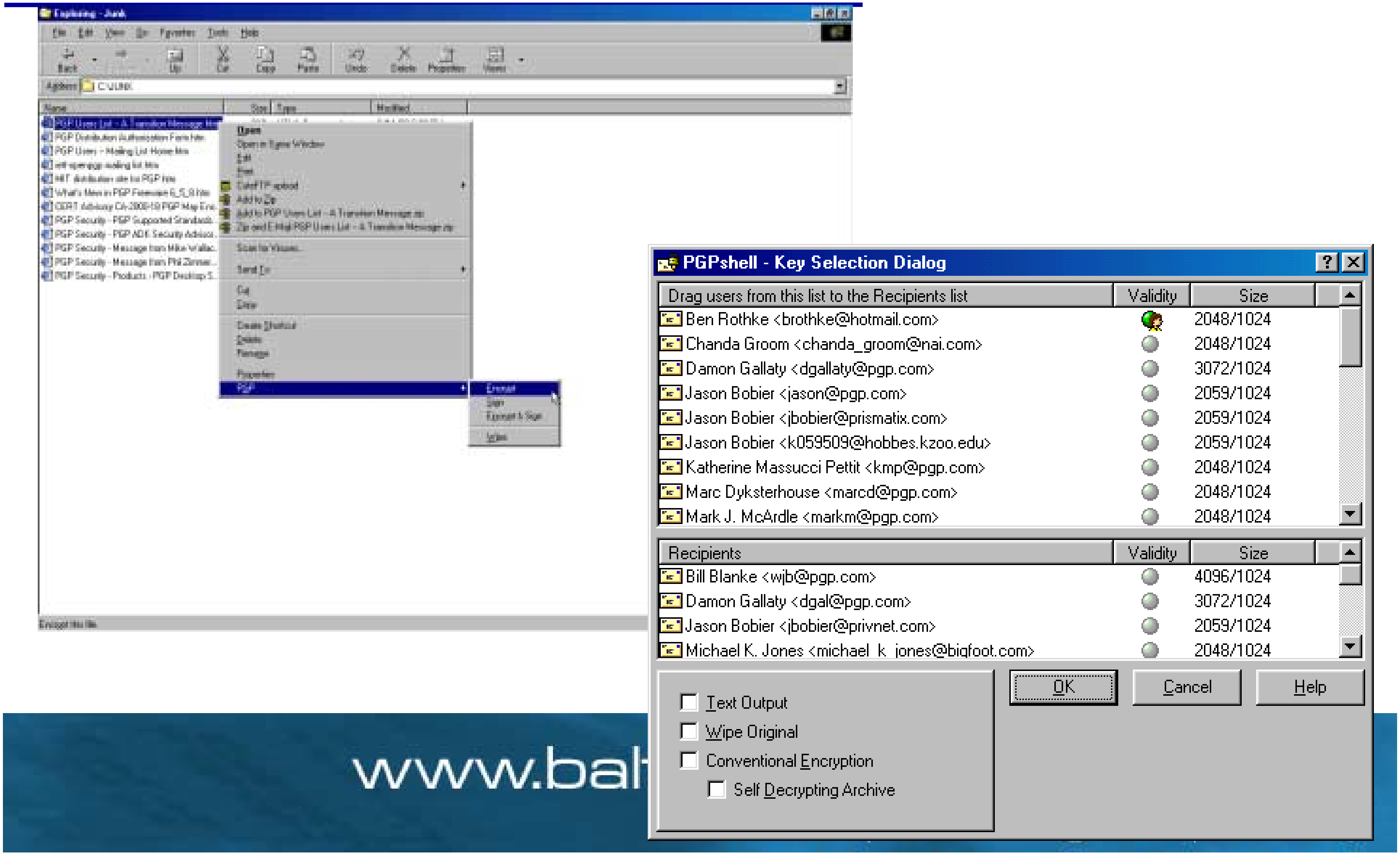
☑ Include Private Key(s)

□ Include 6.0 Extensions

File encryption/decryption

- Encrypt
  - Use Windows Explorer & drag file into PGPtools *Encrypt* or use right-click from mouse
  - Click on *Recipient(s)* & add to recipients lists to add their public key
- Decrypt
  - Use Windows Explorer to drag file into PGPtools *Decrypt/Verify* or double-click from Windows Explorer
  - Enter passphrase for your private key

Encrypting a file

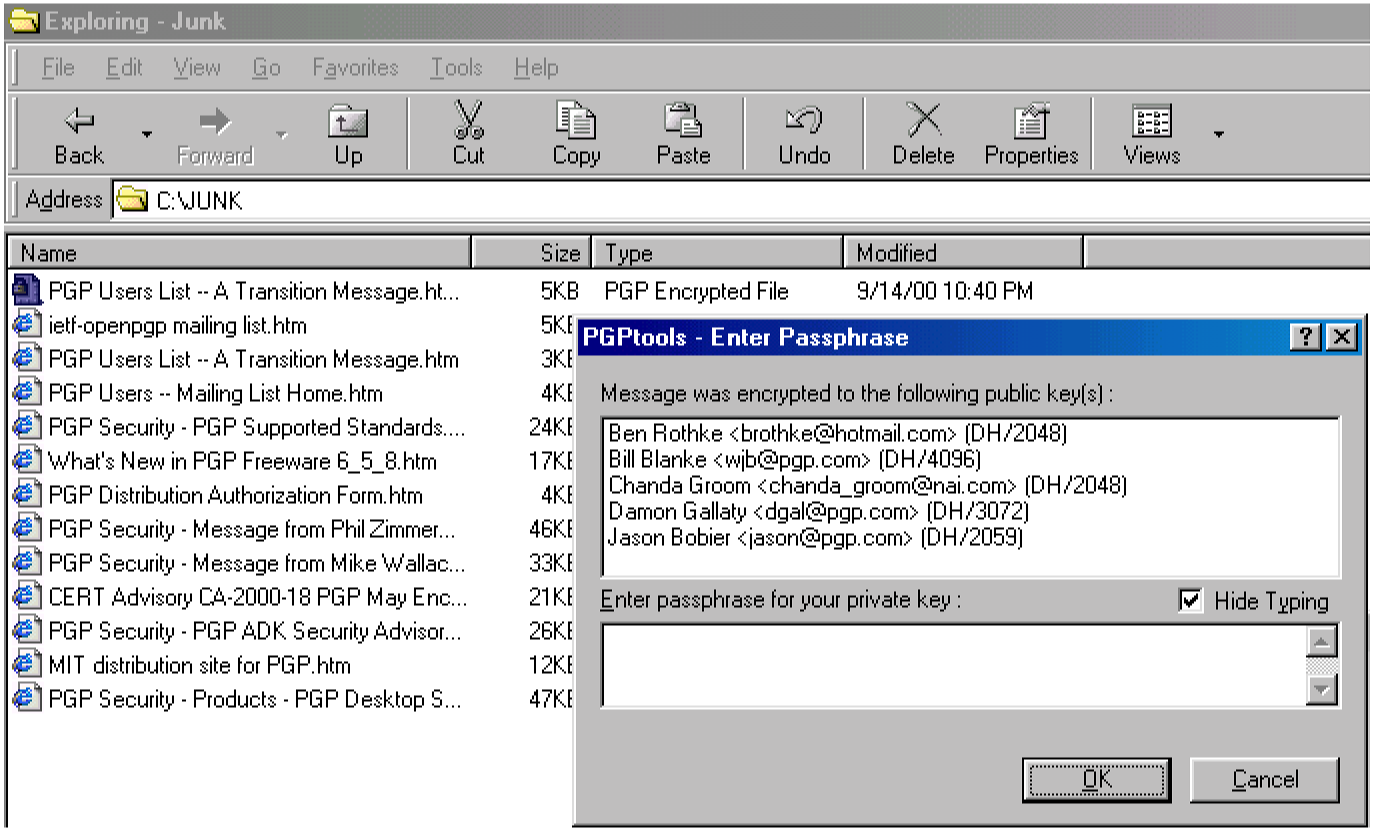

Exploring - Junk
File Edit View Go Favorites Tools Help
Back Up Cut Copy Paste Undo Delete Properties Views
Address C:\JUNK
Name Size Type Modified
PGP Users List - A Transition Message.htm
PGP Distribution Authorization Form.htm
PGP Users - Mailing List Home.htm
mit-openpgp mailing list.htm
MIT distribution site for PGP.htm
What's New in PGP Freeware 6_5_8.htm
CERT Advisory CA-2000-18 PGP May Enc...
PGP Security - PGP Supported Standards...
PGP Security - PGP ADK Security Advisor...
PGP Security - Message from Mike Wallac...
PGP Security - Message from Phil Zimmer...
PGP Security - Products - PGP Desktop S...
Open
Open in Same Window
Edit
Print
CuteFTP upload
Add to Zip
Add to PGP Users List - A Transition Message.zip
Zip and E-Mail PGP Users List - A Transition Message.zip
Scan for Viruses...
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
PGP
Encrypt
Sign
Encrypt & Sign
Wipe
Encrypt this file.
PGPshell - Key Selection Dialog
OK
Cancel
Help

Drag users from this list to the Recipients list	Validity	Size
Ben Rothke <brothke@hotmail.com>	●	2048/1024
Chanda Groom <chanda_groom@nai.com>	●	2048/1024
Damon Gallaty <dgallaty@pgp.com>	●	3072/1024
Jason Bobier <jason@pgp.com>	●	2059/1024
Jason Bobier <jbobier@prismatix.com>	●	2059/1024
Jason Bobier <k059509@hobbes.kzoo.edu>	●	2059/1024
Katherine Massucci Pettit <kmp@pgp.com>	●	2048/1024
Marc Dykstehouse <marcd@pgp.com>	●	2048/1024
Mark J. McArdle <markm@pgp.com>	●	2048/1024

Recipients	Validity	Size
Bill Blanke <wjb@pgp.com>	●	4096/1024
Damon Gallaty <dgal@pgp.com>	●	3072/1024
Jason Bobier <jbobier@privnet.com>	●	2059/1024
Michael K. Jones <michael_k_jones@bigfoot.com>	●	2048/1024

☐ Text Output
☐ Wipe Original
☐ Conventional Encryption
☐ Self Decrypting Archive

Decrypting a file



Name	Size	Type	Modified
PGP Users List -- A Transition Message.ht...	5KB	PGP Encrypted File	9/14/00 10:40 PM
ietf-openpgp mailing list.htm	5KE		
PGP Users List -- A Transition Message.htm	3KE		
PGP Users -- Mailing List Home.htm	4KE		
PGP Security - PGP Supported Standards....	24KE		
What's New in PGP Freeware 6_5_8.htm	17KE		
PGP Distribution Authorization Form.htm	4KE		
PGP Security - Message from Phil Zimmer...	46KE		
PGP Security - Message from Mike Wallac...	33KE		
CERT Advisory CA-2000-18 PGP May Enc...	21KE		
PGP Security - PGP ADK Security Advisor...	26KE		
MIT distribution site for PGP.htm	12KE		
PGP Security - Products - PGP Desktop S...	47KE		

Message was encrypted to the following public key(s) :

Ben Rothke <brothke@hotmail.com> (DH/2048)
Bill Blanke <wjb@pgp.com> (DH/4096)
Chanda Groom <chanda_groom@nai.com> (DH/2048)
Damon Gallaty <dgal@pgp.com> (DH/3072)
Jason Bobier <jason@pgp.com> (DH/2059)

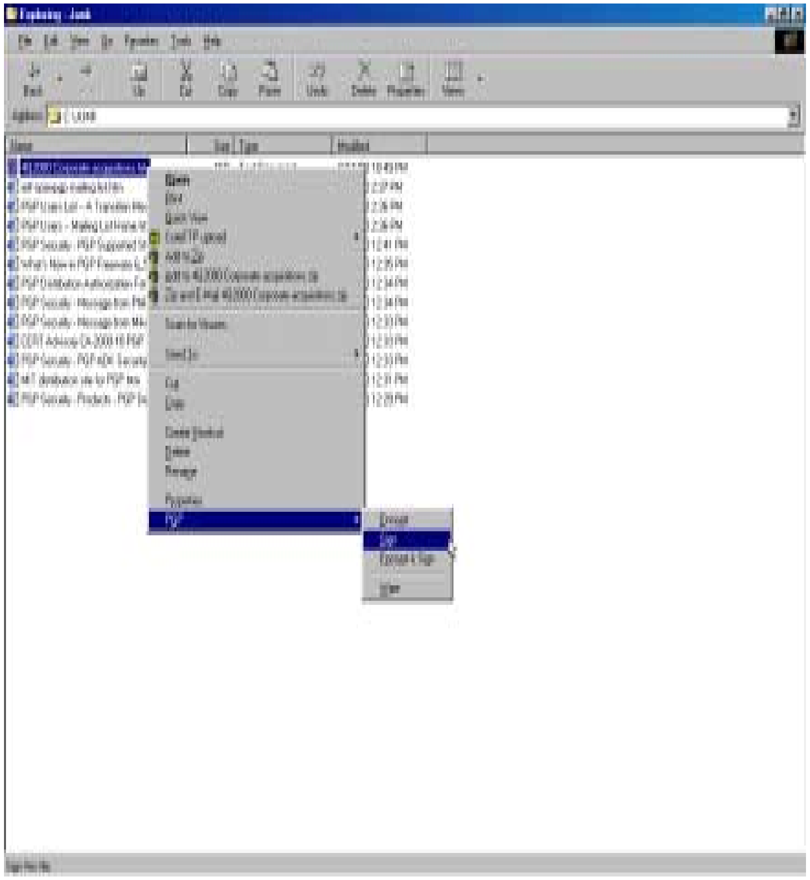
Enter passphrase for your private key : ☑ Hide Typing

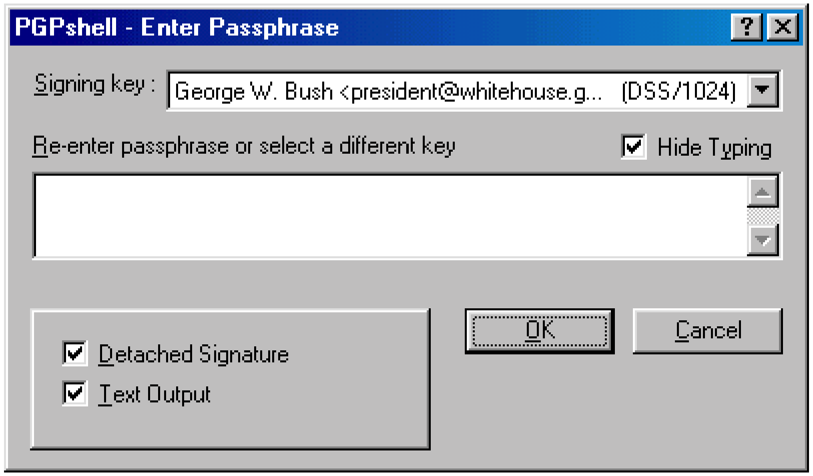
Digital signature signing/verification

- Sign
  - Use Windows Explorer to drag file into PGPtools *Sign*, or right-click on file name.
  - Enter passphrase for your private key.
- Verify
  - Use Windows Explorer to drag file into PGPtools *Decrypt/Verify*, or right-click on file name.

BALTIMORE™
www.baltimore.com

Digital signature signing/verification




PGPshell - Enter Passphrase

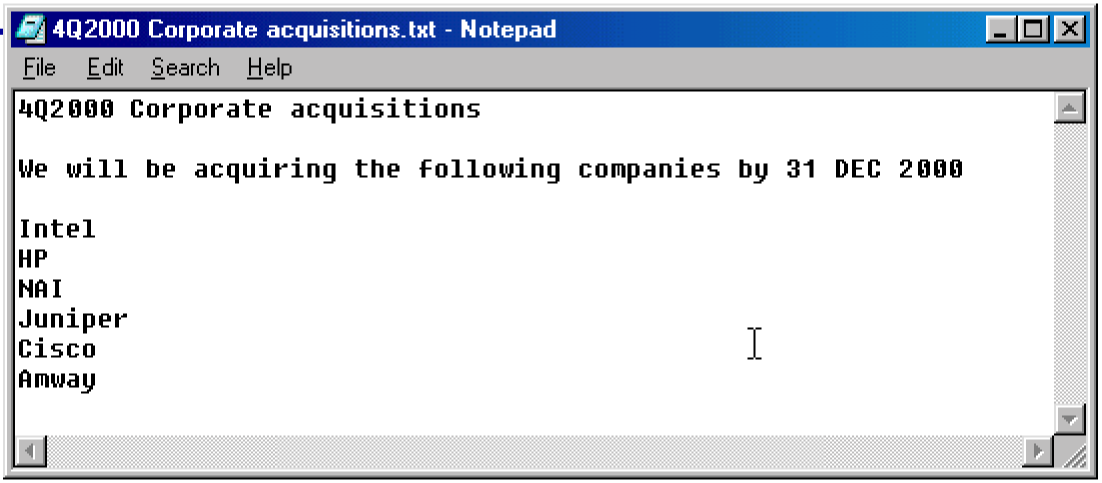
Signing key : George W. Bush <president@whitehouse.g... (DSS/1024)

Re-enter passphrase or select a different key ☑ Hide Typing

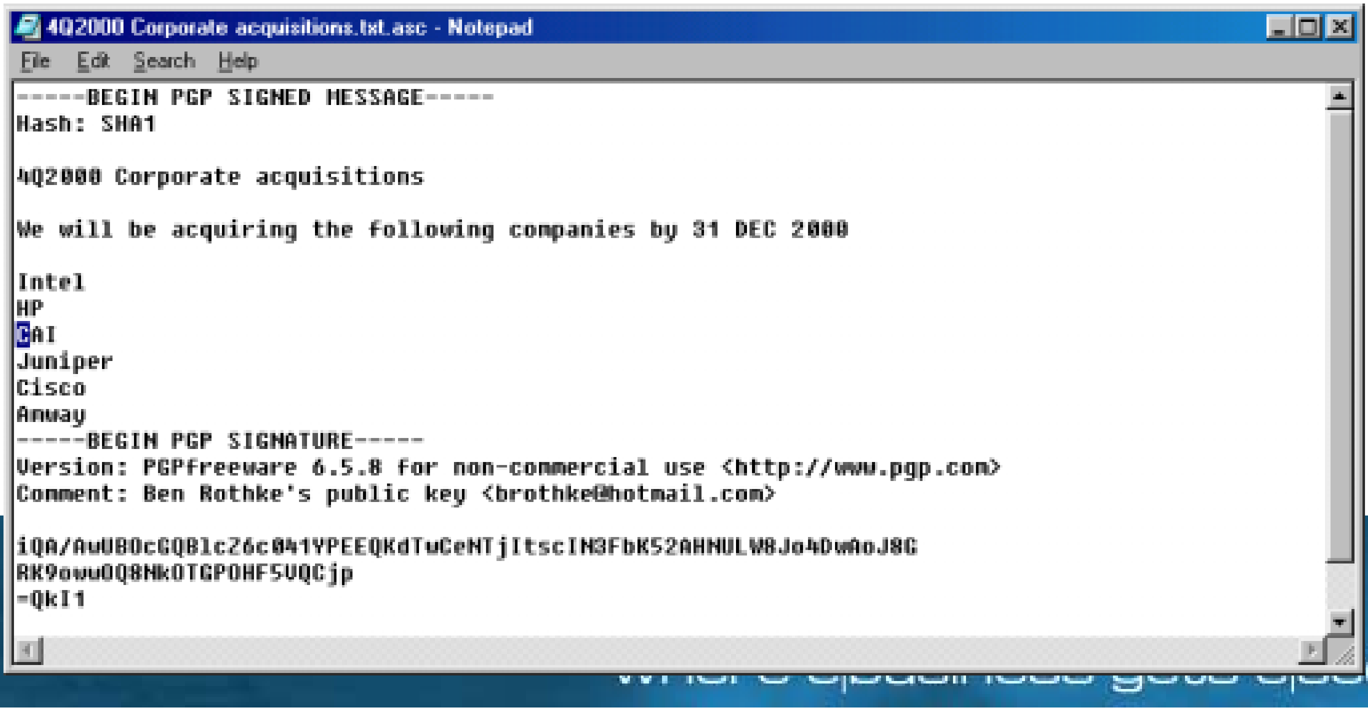
☑ Detached Signature

☑ Text Output

Digital signature verification

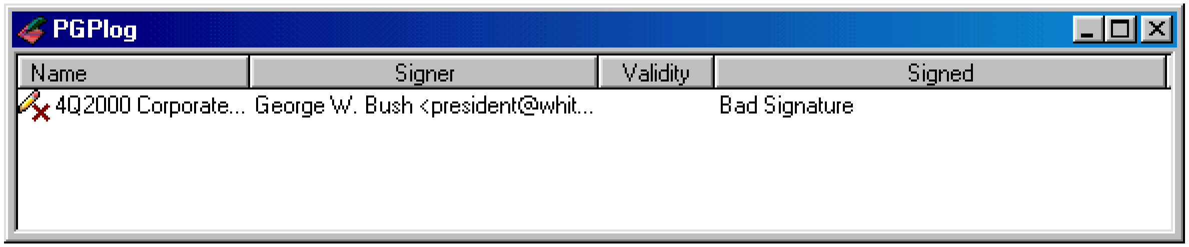


```
4Q2000 Corporate acquisitions

We will be acquiring the following companies by 31 DEC 2000

Intel
HP
NAI
Juniper
Cisco
Amway
```

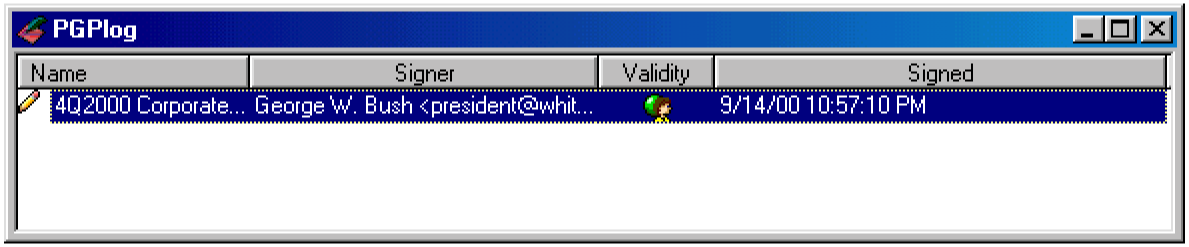


```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

4Q2000 Corporate acquisitions

We will be acquiring the following companies by 31 DEC 2000

Intel
HP
CAI
Juniper
Cisco
Amway
-----BEGIN PGP SIGNATURE-----
Version: PGPfreeware 6.5.8 for non-commercial use <http://www.pgp.com>
Comment: Ben Rothke's public key <brothke@hotmail.com>

iQA/AwUBOcGQBlcZ6c041YPEEQKdTwCeNTjItscIN3FbK52AHNULW8Jo4DwAoJ8G
RK9owwOQ8NkOTGPOHF5VQCjp
=QkI1
```

Digital signature verification



Name	Signer	Validity	Signed
4Q2000 Corporate...	George W. Bush <president@whit...		Bad Signature

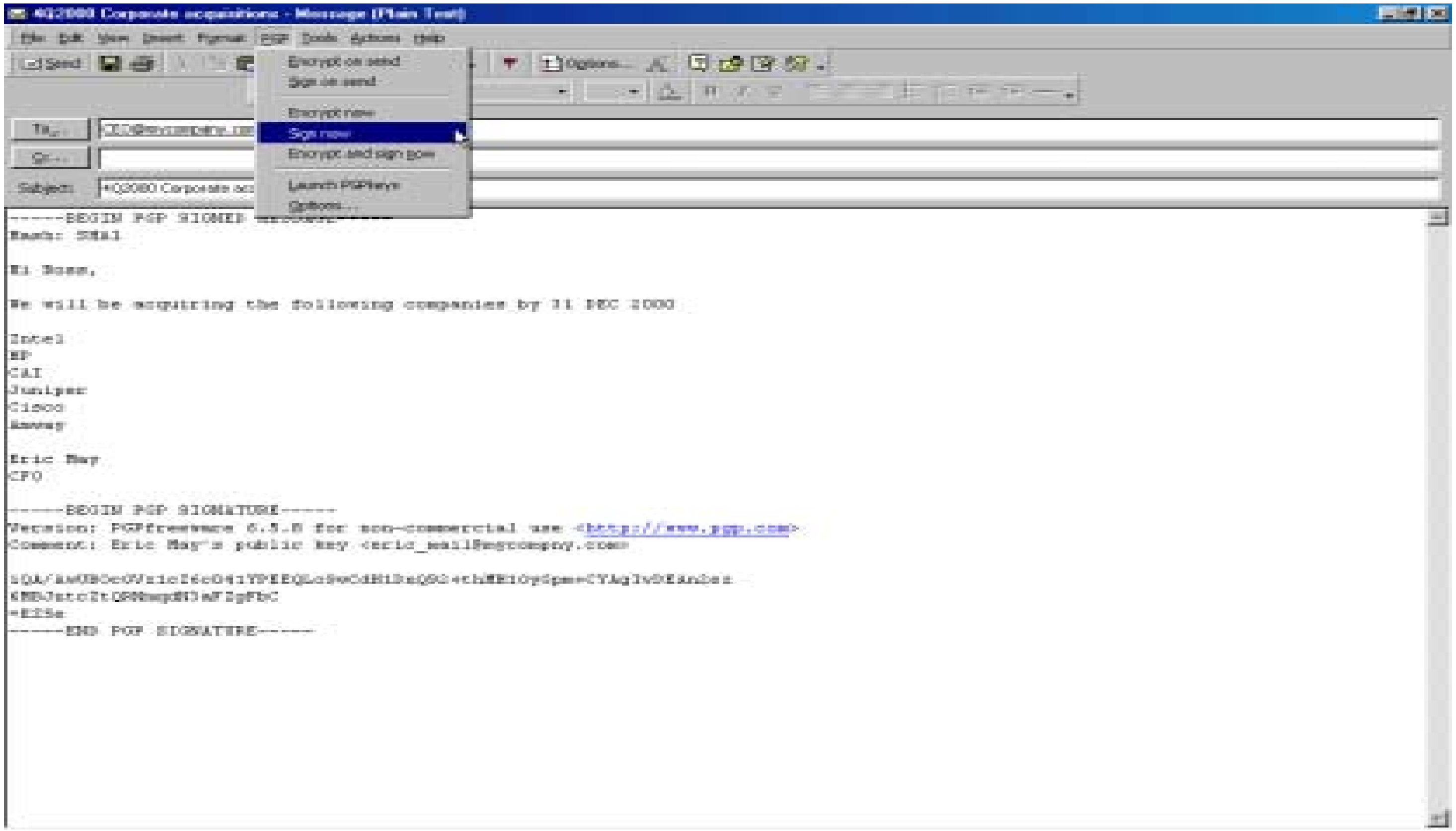


Name	Signer	Validity	Signed
4Q2000 Corporate...	George W. Bush <president@whit...	●	9/14/00 10:57:10 PM

PGP Trust architecture

- No central authority or hierarchical trust. PGP instead uses a *web of trust* architecture.
- Individuals sign each others keys, which progressively forms a web of individual keys, interconnected by links for by these signatures.
  - Any PGP user can act as a quasi certifying authority, and can also validate other PGP users' public-keys.

Secure e-mail

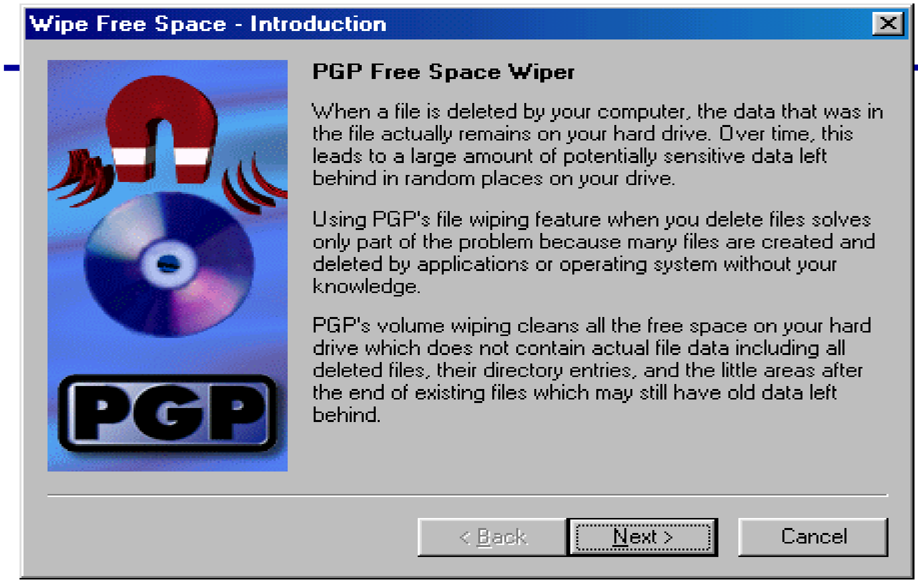


To...: CEO@mycompany.co

Subject: 4Q2000 Corporate ac

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hi Boss,

We will be acquiring the following companies by 31 DEC 2000

Intel
HP
CAI
Juniper
Cisco
Amway

Eric May
CFO

-----BEGIN PGP SIGNATURE-----
Version: PGPfreeware 6.5.8 for non-commercial use <http://www.pgp.com>
Comment: Eric May's public key <eric_mail@mycompany.com>

iQA/AwUBOeOVz1cI6cO4iYPEEQLoSwCdH1DzQSG+thMH1OySpm=CYAg1vDEAnbez
KMBJztcZtQRNmpdH3aF2gFbC
=EZ5e
-----END PGP SIGNATURE-----
```

Freespace Wipe



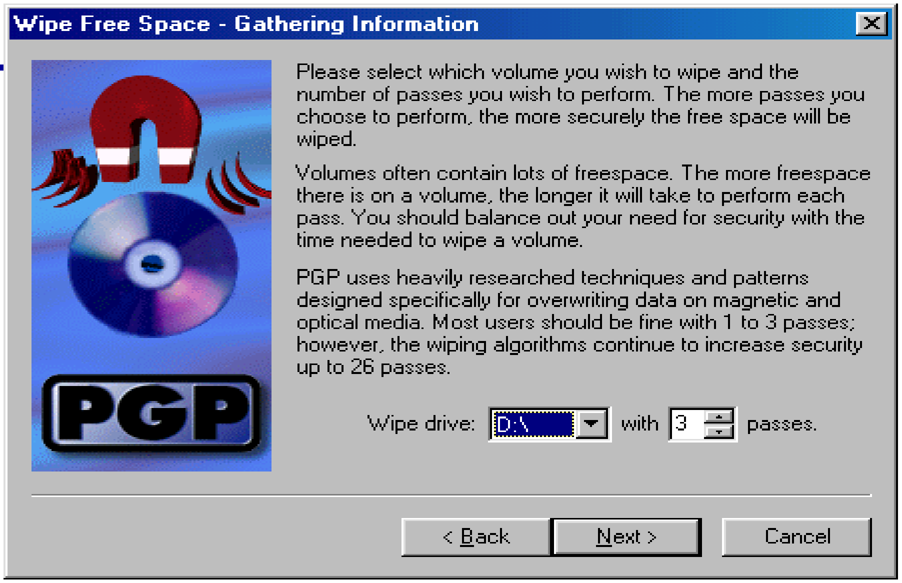
Wipe Free Space - Introduction

PGP Free Space Wiper

When a file is deleted by your computer, the data that was in the file actually remains on your hard drive. Over time, this leads to a large amount of potentially sensitive data left behind in random places on your drive.

Using PGP's file wiping feature when you delete files solves only part of the problem because many files are created and deleted by applications or operating system without your knowledge.

PGP's volume wiping cleans all the free space on your hard drive which does not contain actual file data including all deleted files, their directory entries, and the little areas after the end of existing files which may still have old data left behind.



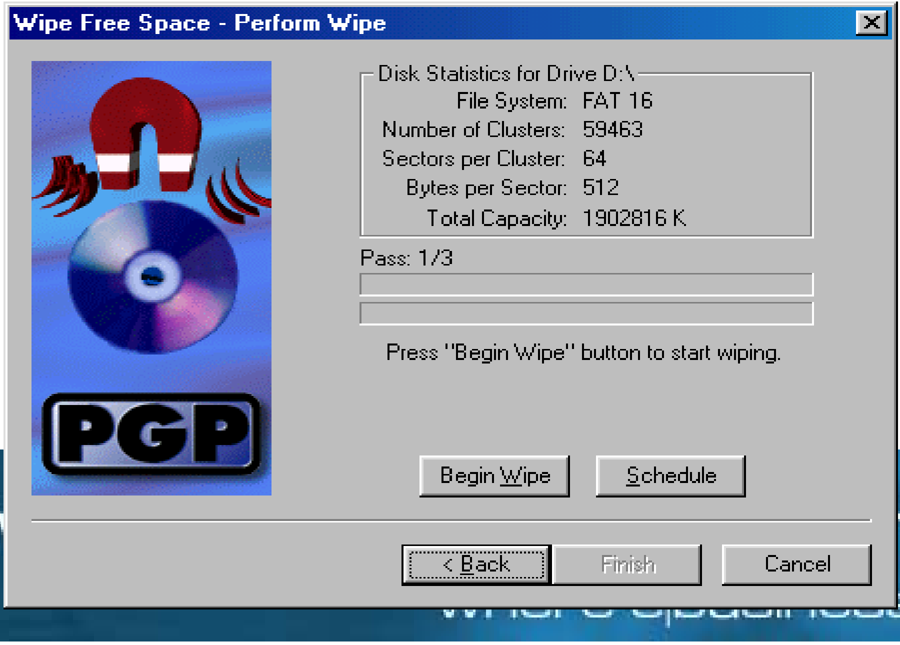
Wipe Free Space - Gathering Information

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 freespace. The more freespace 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 to 3 passes; however, the wiping algorithms continue to increase security up to 26 passes.

Wipe drive: D:\ with 3 passes.



Wipe Free Space - Perform Wipe

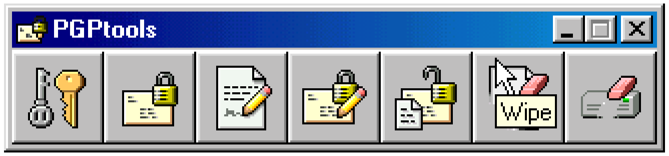
Disk Statistics for Drive D:\

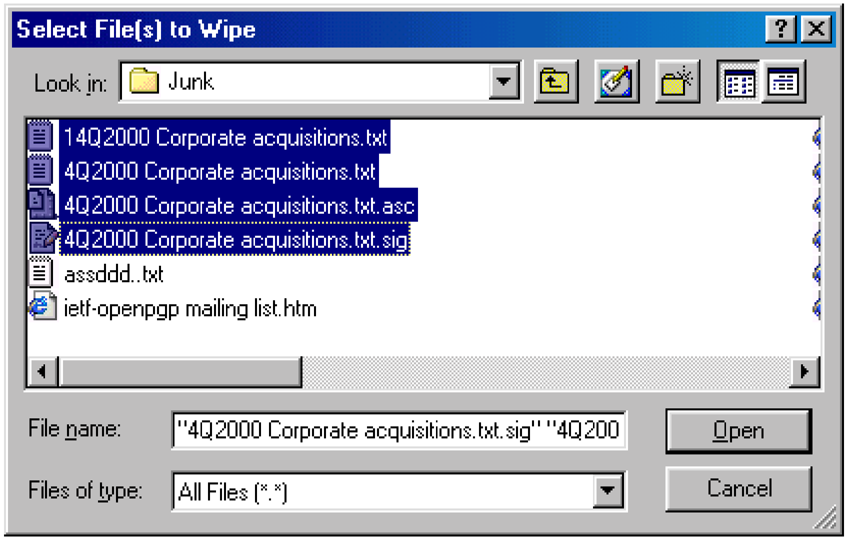
File System:	FAT 16
Number of Clusters:	59463
Sectors per Cluster:	64
Bytes per Sector:	512
Total Capacity:	1902816 K

Pass: 1/3

Press "Begin Wipe" button to start wiping.

Secure file wipe



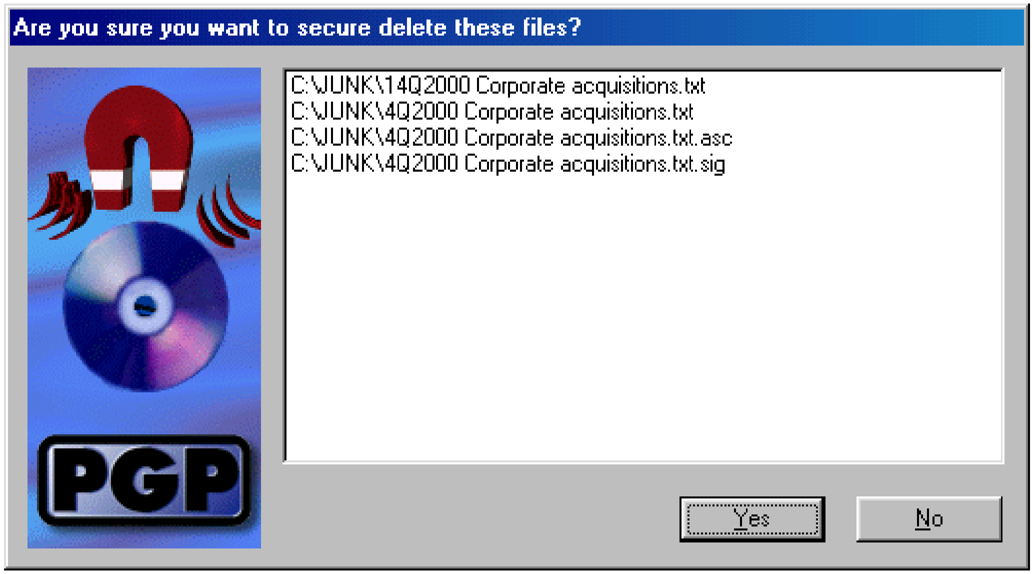


Look in: Junk

14Q2000 Corporate acquisitions.txt
4Q2000 Corporate acquisitions.txt
4Q2000 Corporate acquisitions.txt.asc
4Q2000 Corporate acquisitions.txt.sig
assddd..txt
ietf-openpgp mailing list.htm

File name: "4Q2000 Corporate acquisitions.txt.sig" "4Q200

Files of type: All Files (*.*)



Are you sure you want to secure delete these files?

C:\JUNK\14Q2000 Corporate acquisitions.txt
C:\JUNK\4Q2000 Corporate acquisitions.txt
C:\JUNK\4Q2000 Corporate acquisitions.txt.asc
C:\JUNK\4Q2000 Corporate acquisitions.txt.sig

Additional resources

- PGP-Users Mailing list
  - pgp-users-listbot@cryptorights.org 'subscribe‘
- IETF-OpenPGP mailing list
  - ietf-openpgp-request@imc.org ‘subscribe’
  - Archives available at www.imc.org/ietf-openpgp/mail-archive/
- Yahoo PGP resources
  - http://dir.yahoo.com/computers_and_internet/security_and_encryption/pgp___pretty_good_privacy/
- International PGP group
  - www.pgpi.org/

Thanks for attending

- Any questions? comments? jokes?
- Please fill out your evaluation sheets

Ben Rothke, CISSP, CCO
Senior Security Consultant
Baltimore Technologies
ben.rothke@baltimore.com
973/202-7921