Frequently Asked Questions
alt.security.pgp
Version 6
13-Jan-23

==========

IMPORTANT DISCLAIMER!

The use of PGP raises a number of political and legal issues. I AM NOT a lawyer and AM NOT qualified to give any legal opinions. Nothing in this document should be interpreted as legal advice. If you have any legal questions concerning the use of PGP, you should consult an attorney who specializes in patent and/or export law. In any case, the law will vary from country to country.

==========

Revision History

Ver	Date	Description
1	09-Dec-93	Proof Reading Copy - Limited Distribution
2	11-Dec-93	First Preliminary Posting (Many changes)
3	19-Dec-93	Second Preliminary Posting (Many changes)
4	01-Jan-94	Third Preliminary Posting (Many changes)

15-Jan-94 Changes for version 5:

Fixed a number of minor spelling, grammar, and typographical errors.
Corrected phone number for Colorado Catacombs BBS.
Added pgp-public-keys@pgp.iastate.edu to servers no longer in service.
Added talk.politics.crypto to related newsgroups.
Added new pgp support product: PGP-Front.
Added new pgp support product: PGPWinFront.
Updated pgp support product information: PBBS availability.
Updated pgp support product information: PGPSHELL.
Added section on how to obtain _Wired_ articles.
Added announcement of Commodore Amiga PGP 2.3a patchlevel 2.
Recommended reading list: Added notation of articles available online.

26-Jan-94 Changes for version 6:

Fixed a number of minor spelling, grammar, and typographical errors.
Modified Public Key Server List in section 8.2
Added note that 2.3a is the latest version for VAX/VMS in section 1.10.5.

Added Emacs Auto-PGP 1.02 to support product section in Appendix I.
Modified dates to a format that is not ambiguous for international use.

==========

Please read this FAQ over and let me know of any additions, deletions, or corrections. It should be noted that most of the questions and answers concerning PGP apply equally well to the ViaCrypt(tm) version as well. All additions, deletions, or corrections to this list should be directed to me at gbe@netcom.com (Gary Edstrom). I will acknowledge all e-mail.

This FAQ is slanted towards the DOS or Unix users of PGP and many of the examples given may only apply to them. For other systems, I would like to direct your attention to the following documents:

MAC: "Here's How to MacPGP!" by Xenon <an48138@anon.penet.fi>
Archimedes PGP comes with its own PGPhints file.
Send e-mail to pgpinfo@mantis.co.uk for a list of PGP tips.

I would like to thank Paul Allen <pla@sktb.demon.co.uk> for allowing me to use some of his PGHints file in this FAQ.

The files making up this FAQ are available via ftp at netcom.com:/pub/gbe. The file names are pgp-faq*.asc and are in clearsig pgp format. In addition, the file pgp-faq.doc is available which is in the original Microsoft Word for Windows format under which this FAQ was created..
--
Gary B. Edstrom | Sequoia Software | PGP fingerprint:
Internet: gbe@netcom.com | Programming Services | 2F F6 1B 28 6E A6 09 6C
CompuServe: 72677,564 | P.O. Box 9573 | B0 EA 9E 4C C4 C6 7D 46
Fax: 1-818-247-6046 | Glendale, CA 91226 | Key available via finger
What is PGP? Subscribe to alt.security.pgp and find out!

==========

Table of Contents

Part 1/4
µ

Part 2/4

Part 3/4

Part 4/4

Appendix VIII - Unites States Congress Phone and FAX List

==========

1. Introductory Questions

1.1. What is PGP?

PGP is a program that gives your electronic mail something that it otherwise doesn't have: Privacy. It does this by encrypting your mail so that nobody but the intended person can read it. When encrypted, the message looks like a meaningless jumble of random characters. PGP has proven itself quite capable of resisting even the most sophisticated forms of analysis aimed at reading the encrypted text.

PGP can also be used to apply a digital signature to a message without encrypting it. This is normally used in public postings where you don't want to hide what you are saying, but rather want to allow others to confirm that the message actually came from you. Once a digital signature is created, it is impossible for anyone to modify either the message or the signature without the modification being detected by PGP.

While PGP is easy to use, it does give you enough rope so that you can hang yourself. You should become thoroughly familiar with the various options in PGP before using it to send serious messages. For example, giving the command "PGP -sat <filename>" will only sign a message, it will not encrypt it. Even though the output looks like it is encrypted, it really isn't. Anybody in the world would be able to recover the original text.

1.2. Why should I encrypt my mail? I'm not doing anything illegal!

You should encrypt your e-mail for the same reason that you don't write all of your correspondence on the back of a post card. E-mail is actually far less secure than the postal system. With the post office, you at least put your letter inside an envelope to hide it from casual snooping. Take a look at the header area of any e-mail message that you receive and you will see that it has passed through a number of nodes on its way to you. Every one of these nodes presents the opportunity for snooping. Encryption in no way should imply illegal activity. It is simply intended to keep personal thoughts personal.

Xenon <an48138@anon.penet.fi> puts it like this:

Crime? If you are not a politician, research scientist, investor, CEO, lawyer, celebrity, libertarian in a repressive society, investor, or person having too much fun, and you do not send e-mail about your private sex life, financial/political/legal/scientific plans, or gossip then maybe you don't need PGP, but at least realize that privacy has nothing to do with crime and is in fact what keeps the world from falling apart. Besides, PGP is FUN. You never had a secret decoder ring? Boo! -Xenon (Copyright 1993, Xenon)

1.3. What are public keys and private keys?

With conventional encryption schemes, keys must be exchanged with everyone you wish to talk to by some other secure method such as face to face meetings, or via a trusted courier. The problem is that you need a secure channel before you can establish a secure channel! With conventional encryption, either the same key is used for both encryption and decryption or it is easy to convert either key to the other. With public key encryption, the encryption and decryption keys are different and it is impossible for anyone to convert one to the other. Therefore, the encryption key can be made public knowledge, and posted in a database somewhere. Anyone wanting to send you a message would obtain your encryption key from this database or some other source and encrypt his message to you. This message can't be

decrypted with the encryption key. Therefore nobody other than the intended receiver can decrypt the message. Even the person who encrypted it can not reverse the process. When you receive a message, you use your secret decryption key to decrypt the message. This secret key never leaves your computer. In fact, your secret key is itself encrypted to protect it from anyone snooping around your computer.

1.4. How much does PGP cost?

Nothing! (Compare to ViaCrypt PGP at $98!) It should be noted, however, that in the United States, the freeware version of PGP *MAY* be a violation of a patent held by Public Key Partners (PKP).

1.5. Is encryption legal?

In much of the civilized world, encryption is either legal, or at least tolerated. However, there are a some countries where such activities could put you in front of a firing squad! Check with the laws in your own country before using PGP or any other encryption product. A couple of the countries where encryption is illegal are Iran and Iraq.

1.6. Is PGP legal?

In addition to the comments about encryption listed above, there are a couple of additional issues of importance to those individuals residing in the United States or Canada. First, there is a question as to whether or not PGP falls under ITAR regulations with govern the exporting of cryptographic technology from the United States and Canada. This despite the fact that technical articles on the subject of public key encryption have been available legally world wide for a number of years. Any competent programmer would have been able to translate those articles into a workable encryption program. There is the possibility that ITAR regulations may be relaxed to allow for encryption technology.

1.7. Where can I get translations of the PGP documentation and/or language.txt files?

Spanish: Armando Ramos <armando@clerval.org>
German: Marc Aurel <4-tea-2@bong.saar.de>
Lithuanian: Zygimantas Cepaitis, Bokera Ltd., Kaunas Lithuania.
e-mail: <zcepaitis@ktl.fi> or <zygis@bokera.lira.lt.ee>
ftp: ghost.dsi.unimi.it:/pub/crypt/pgp23ltk.zip
ftp: nic.funet.fi:/pub/crypt/ghost.dsi.unimi.it/pgp23ltk.zip

1.8. Is there an archive site for alt.security.pgp?

laszlo@instrlab.kth.se (Laszlo Baranyi) says:

"My memory says that ripem.msu.edu stores a backlog of both alt.security.pgp, and sci.crypt. But that site is ONLY open for ftp for those that are inside US."

1.9. Is there a commercial version of PGP available?

Yes, by arrangement with the author of PGP, a company called ViaCrypt is marketing a version of PGP that is almost identical to the version currently available on Internet. Each can read or write messages to the

other. The list price of ViaCrypt PGP is $98 (US) for a single user license and is NOT available for export from the United States. In addition, it is presently available only for MS-DOS. Versions for other platforms are under development. While the present product is 100% compatible with free PGP, it is not known if this will remain the case in the future. The address of ViaCrypt is:

ViaCrypt
David A. Barnhart
Product Manager
2104 West Peoria Avenue
Phoenix, Arizona 85029
Tel: (602) 944-0773
Fax: (602) 943-2601
E-Mail: 70304.41@compuserve.com
E-Mail: wk01965@worldlink.com
Credit card orders only. (800)536-2664 (8-5 MST M-F)

1.10. What platforms has PGP been ported to?

1.10.1. DOS: 2.3a

1.10.2. MAC: 2.3

1.10.3. OS/2: 2.3a

1.10.4. Unix: 2.3a (Variations exist for many different systems.)

1.10.5. VAX/VMS: 2.3a

1.10.6. Atari ST: 2.3a

1.10.7. Archimedes: 2.3a subversion 1.18b

1.10.8. Commodore Amiga: 2.3a patchlevel 2

From: simons@peti.GUN.de (Peter Simons)
Date: Fri, 31 Dec 1993 08:10:53 +0100
Newsgroups: alt.security.pgp
Subject: PGPAmiga 2.3a.2 available for FTP
TITLE
Pretty Good Privacy (PGP)
VERSION
Version 2.3a patchlevel 2
AUTHOR
Amiga port and enhancements by Peter Simons <simons@peti.GUN.de>

CHANGES

This version is re-compiled with SAS/C 6.50. A few minor bugs have been fixed. Additionally, the manual is now availabe in TexInfo style and can easily be converted into AmigaGuide, postscript, dvi or whatever format. AmigaGuide versions are included.

Also for the first time, the alt.security.pgp frequently asked questions (FAQ) are included in the archive.

NOTES

Please take note that the archive contains a readme file, with checksums for ALL files in the distribution and is signed with my key! Please be careful, if this file is missing or rigged!

A mailing list concerning PGPAmiga has been opened on peti.GUN.de. To subscribe, send e-mail to listserv@peti.GUN.de with "ADD your_address PGPAmiga" in the message body. You may add "HELP" in the next line to receive a command overview of ListSERV.

SPECIAL REQUIREMENTS

none

HOST NAME

Any Aminet host, i.e. ftp.uni-kl.de (131.246.9.95).

DIRECTORY

/pub/aminet/util/crypt/

FILE NAMES

PGPAmi23a_2.lha

PGPAmi23a2_src.lha

1.11. Where can I obtain PGP?

FTP sites:

soda.berkeley.edu
  /pub/cypherpunks/pgp (DOS, MAC)
  Verified: 21-Dec-93
ftp.demon.co.uk
  /pub/amiga/pgp
  /pub/archimedes
  /pub/pgp
  /pub/mac/MacPGP
ftp.informatik.tu-muenchen.de
ftp.funet.fi
ghost.dsi.unimi.it
  /pub/crypt
  Verified: 21-Dec-93
ftp.tu-clausthal.de (139.174.2.10)
wuarchive.wustl.edu
  /pub/aminet/util/crypt
src.doc.ic.ac.uk (Amiga)
  /aminet
  /amiga-boing
ftp.informatik.tu-muenchen.de
  /pub/comp/os/os2/crypt/pgp23os2A.zip (OS/2)
black.ox.ac.uk (129.67.1.165)
  /src/security (Unix)

iswuarchive.wustl.edu
pub/aminet/util/crypt (Amiga)
csn.org
/mpj (see README.MPJ for export restrictions)
nic.funet.fi (128.214.6.100)
van-bc.wimsey.bc.ca (192.48.234.1)
ftp.uni-kl.de (131.246.9.95)
qiclab.scn.rain.com (147.28.0.97)
pc.usl.edu (130.70.40.3)
leif.thep.lu.se (130.235.92.55)
goya.dit.upm.es (138.4.2.2)
tupac-amaru.informatik.rwth-aachen.de (137.226.112.31)
ftp.etsu.edu (192.43.199.20)
princeton.edu (128.112.228.1)
pencil.cs.missouri.edu (128.206.100.207)

Also, try an archie search for PGP using the command:

archie -s pgp23 (DOS Versions)
archie -s pgp2.3 (MAC Versions)

ftpmail:

For those individuals who do not have access to FTP, but do have access to e-mail, you can get FTP files mailed to you. For information on this service, send a message saying "Help" to ftpmail@decwrl.dec.com. You will be sent an instruction sheet on how to use the ftpmail service.

BBS sites:

Hieroglyphics Vodoo Machine (Colorado)
DOS version only
(303) 443-2457
Verified: 26-Dec-93

Colorado Catacombs BBS
(303) 938-9654

Exec-Net (New York)
Host BBS for the ILink net.
(914) 667-4567

The Grapvine BBS (Little Rock Arkansas)
No longer in operation

2. General Questions

2.1. Why can't a person using version 2.2 read my version 2.3 message?

Try adding "+pkcs_compat=0" to your command line as follows: "pgp -seat +pkcs_compat=0 <filename>" By default, version 2.3 of PGP uses a different header format that is not compatible with earlier versions

of PGP. Inserting this option into the command will force PGP to use the older header format. You can also set this option in your config.txt file, but this is not recommended.

2.2. Why does it take so long to encrypt/decrypt messages?

This problem can arise when you have placed the entire public key ring from one of the servers into the pubring.pgp file. PGP may have to search through several thousand keys to find the one that it is after. The solution to this dilemma is to maintain 2 public key rings. The first ring, the normal pubring.pgp file, should contain only those individuals that you send messages to quite often. The second key ring can contain ALL of the keys for those occasions when the key you need isn't in your short ring. You will, of course, need to specify the key file name whenever encrypting messages using keys in your secondary key ring. Now, when encrypting or decrypting messages to individuals in your short key ring, the process will be a LOT faster.

2.3. How do I create a secondary key file?

First, let's assume that you have all of the mammoth public key ring in your default pubring.pgp file. First, you will need to extract all of your commonly used keys into separate key files using the -kx option. Next, rename pubring.pgp to some other name. For this example, I will use the name pubring.big. Next, add each of the individual key files that you previously created to a new pubring.pgp using the -ka option. You now have your 2 key rings. To encrypt a message to someone in the short default file, use the command "pgp -e <userid>". To encrypt a message to someone in the long ring, use the command "pgp -e <userid> c:\pgp\pubring.big". Note that you need to specify the complete path and file name for the secondary key ring. It will not be found if you only specify the file name.

2.4. How does PGP handle multiple addresses?

When encrypting a message to multiple addresses, you will notice that the length of the encrypted file only increases by a small amount for each additional address. The reason that the message only grows by a small amount for each additional key is that the body of the message is only encrypted once using a random session key and IDEA. It is only necessary then to encrypt this session key once for each address and place it in the header of the message. Therefore, the total length of a message only increases by the size of a header segment for each additional address. (To avoid a known weakness in RSA when encrypting the same message to multiple recipients, the IDEA session key is padded with different random data each time it is RSA-encrypted.)

2.5. How can I use PGP to create a return receipt for a message?

I was planning on including a section on this question. However, while following a similar thread in alt.security.pgp, I realized that there were too many unresolved issues to include an answer here. I may try to include the subject in a future release of the FAQ.

2.6. Where can I obtain scripts to integrate pgp with my email or news reading system?

The scripts that come with the source code of PGP are rather out of date. Newer versions of some of the scripts are available via anonymous ftp at
ftp.informatik.uni-hamburg.de:/pub/virus/misc/contrib.zip

3. Keys

3.1. Which key size should I use?

PGP gives you 4 choices of key size: 384, 512, 1024, or a user selected number of bits. The larger the key, the more secure the RSA portion of the encryption is. The only place where the key size makes a large change in the running time of the program is during key generation. A 1024 bit key can take 8 times longer to generate than a 384 bit key. Fortunately, this is a one time process that doesn't need to be repeated unless you wish to generate another key pair. During encryption, only the RSA portion of the encryption process is affected by key size. The RSA portion is only used for encrypting the session key used by the IDEA. The main body of the message is totally unaffected by the choice of RSA key size. So unless you have a very good reason for doing otherwise, select the 1024 bit key size. Using currently available algorithms for factoring, the 384 bit key is just not far enough out of reach to be a good choice.

3.2. Why does PGP take so long to add new keys to my key ring?

The time required to check signatures and add keys to your public key ring tends to grow as the square of the size of your existing public key ring. This can reach extreme proportions. I just recently added the entire 850KB public key ring form one of the key servers to my local public key ring. Even on my 66MHz 486 system, the process took over 10 hours.

3.3. How can I extract multiple keys into a single armored file?

A number of people have more than one public key that they would like to make available. One way of doing this is executing the "-kxa" command for each key you wish to extract from the key ring into separate armored files, then appending all the individual files into a single long file with multiple armored blocks. This is not as convenient as having all of your keys in a single armored block.

Unfortunately, the present version of PGP does not allow you to do this directly. Fortunately, there is an indirect way to do it. First, extract each of the desired keys into separate armored key files using the command "pgp -kxa <key>". Next, create a temporary key ring by adding the individual key files one by one using the command "pgp -ka <keyfile> <temp-key-ring>". This new temporary key ring will contain only the keys that you are interested in. Finally, execute the command "pgp -kxa * <new-armored-file> <temp-key-ring> to extract all of the keys in the temporary ring to an armored file. Note the "*" in the previous command. It is not described in the PGP documentation but apparently means "all keys". This armored file now contains all of the desired keys just as if pgp had had a built in command to do it in the first place.

A Unix script to perform the extraction with a singled command would be as follows:

```
foreach name (name1 name2 name3 ...)
pgp -kx $name /tmp/keys.pgp <keyring>
end
```

An equilivent DOS command would be:

for %a in (name1 name2 name3 ...) do pgp -kx %a <keyring>

3.4. I tried encrypting the same message to the same address two different times and got completely different outputs. Why is this?

Every time you run pgp, a different session key is generated. This session key is used as the key for IDEA. As a result, the entire header and body of the message changes. You will never see the same output twice, no matter how many times you encrypt the same message to the same address. This adds to the overall security of PGP.

3.5. How do I specify which key to use when an individual has 2 or more public keys and the very same user ID on each, or when 2 different users have the same name?

Instead of specifying the user's name in the ID field of the PGP command, you can use the key ID number. The format is 0xNNNNNN where NNNNNN is the user's 6 character key ID number. It should be noted that you don't need to enter the entire ID number, a few consecutive digits from anywhere in the ID should do the trick. Be careful: If you enter "0x123", you will be matching key IDs 0x123937, 0x931237, or 0x912373. Any key ID that contains "123" anywhere in it will produce a match. They don't need to be the starting characters of the key ID. You will recognize that this is the format for entering hex numbers in the C programming language. For example, any of the following commands could be used to encrypt a file to me.

```
pgp -e <filename> "Gary Edstrom"
pgp -e <filename> gbe@netcom.com
pgp -e <filename> 0x90A9C9
```

This same method of key identification can be used in the config.txt file in the "MyName" variable to specify exactly which of the keys in the secret key ring should be used for encrypting a message.

3.6. What does the message "Unknown signator, can't be checked" mean?

It means that the key used to create that signature does not exist in your database. If at sometime in the future, you happen to add that key to your database, then the signature line will read normally. It is completely harmless to leave these non checkable signatures in your database. They neither add to nor take away from the validity of the key in question.

3.7. How do I get PGP to display the trust parameters on a key?

You can only do this when you run the -kc option by itself on the entire database. The parameters will NOT be shown if you give a specific ID on the command line. The correct command is: "pgp -kc". The command "pgp -kc smith" will NOT show the trust parameters for smith.

4. Security Questions

4.1. How secure is PGP?

The big unknown in any encryption scheme based on RSA is whether or not there is an efficient way to factor huge numbers, or if there is some backdoor algorithm that can break the code without solving the

factoring problem. Even if no such algorithm exists, it is still believed that RSA is the weakest link in the PGP chain.

4.2. Can't you break PGP by trying all of the possible keys?

This is one of the first questions that people ask when they are first introduced to cryptography. They do not understand the size of the problem. For the IDEA encryption scheme, a 128 bit key is required. Any one of the 2^128 possible combinations would be legal as a key, and only that one key would successfully decrypt all message blocks. Let's say that you had developed a special purpose chip that could try a billion keys per second. This is FAR beyond anything that could really be developed today. Let's also say that you could afford to throw a billion such chips at the problem at the same time. It would still require over 10,000,000,000,000 years to try all of the possible 128 bit keys. That is something like a thousand times the age of the known universe! While the speed of computers continues to increase and their cost decrease at a very rapid pace, it will probably never get to the point that IDEA could be broken by the brute force attack.

The only type of attack that might succeed is one that tries to solve the problem from a mathematical standpoint by analyzing the transformations that take place between plain text blocks, and their cipher text equivalents. IDEA is still a fairly new algorithm, and work still needs to be done on it as it relates to complexity theory, but so far, it appears that there is no algorithm much better suited to solving an IDEA cipher than the brute force attack, which we have already shown is unworkable. The nonlinear transformation that takes place in IDEA puts it in a class of extremely difficult to solve mathmatical problems.

4.3. How secure is the conventional cryptography (-c) option?

Assuming that you are using a good strong random pass phrase, it is actually much stronger than the normal mode of encryption because you have removed RSA which is believed to be the weakest link in the chain. Of course, in this mode, you will need to exchange secret keys ahead of time with each of the recipients using some other secure method of communication, such as an in-person meeting or trusted courier.

4.4. Can the NSA crack RSA?

This question has been asked many times. If the NSA were able to crack RSA, you would probably never hear about it from them. The best defense against this is the fact the algorithm for RSA is known world wide. There are many competent mathematicians and cryptographers outside the NSA and there is much research being done in the field right now. If any of them were to discover a hole in RSA, I'm sure that we would hear about it from them. I think that it would be hard to hide such a discovery. For this reason, when you read messages on USENET saying that "someone told them" that the NSA is able to break pgp, take it with a grain of salt and ask for some documentation on exactly where the information is coming from.

4.5. How secure is the "for your eyes only" option (-m)?

It is not secure at all. There are many ways to defeat it. Probably the easiest way is to simply redirect your screen output to a file as follows:

pgp [filename] > [diskfile]

The -m option was not intended as a fail-safe option to prevent plain text files from being generated, but to serve simply as a warning to the person decrypting the file that he probably shouldn't keep a copy of the plain text on his system.

4.6. What if I forget my pass phrase?

In a word: DON'T. If you forget your pass phrase, there is absolutely no way to recover any encrypted files. I use the following technique: I have a backup copy of my secret key ring on floppy, along with a sealed envelope containing the pass phrase. I keep these two items in separate safe locations, neither of which is my home or office. The pass phrase used on this backup copy is different from the one that I normally use on my computer. That way, even if some stumbles onto the hidden pass phrase and can figure out who it belongs to, it still doesn't do them any good, because it is not the one required to unlock the key on my computer.

4.7. Why do you use the term "pass phrase" instead of "password"?

This is because most people, when asked to choose a password, select some simple common word. This can be cracked by a program that uses a dictionary to try out passwords on a system. Since most people really don't want to select a truly random password, where the letters and digits are mixed in a nonsense pattern, the term pass phrase is used to urge people to at least use several unrelated words in sequence as the pass phrase.

4.8. If my secret key ring is stolen, can my messages be read?

No, not unless they have also stolen your secret pass phrase, or if your pass phrase is susceptible to a brute-force attack. Neither part is useful without the other. You should, however, revoke that key and generate a fresh key pair using a different pass phrase. Before revoking your old key, you might want to add another user ID that states what your new key id is so that others can know of your new address.

4.9. How do I choose a pass phrase?

All of the security that is available in PGP can be made absolutely useless if you don't choose a good pass phrase to encrypt your secret key ring. Too many people use their birthday, their telephone number, the name of a loved one, or some easy to guess common word. While there are a number of suggestions for generating good pass phrases, the ultimate in security is obtained when the characters of the pass phrase are chosen completely at random. It may be a little harder to remember, but the added security is worth it. As an absolute minimum pass phrase, I would suggest a random combination of at least 8 letters and digits, with 12 being a better choice. With a 12 character pass phrase made up of the lower case letters a-z plus the digits 0-9, you have about 62 bits of key, which is 6 bits better than the 56 bit DES keys. If you wish, you can mix upper and lower case letters in your pass phrase to cut down the number of characters that are required to achieve the same level of security. I don't do this myself because I hate having to manipulate the shift key while entering a pass phrase.

A pass phrase which is composed of ordinary words without punctuation or special characters is susceptible to a dictionary attack. Transposing characters or mis-spelling words makes your pass phrase less vulnerable, but a professional dictionary attack will cater for this sort of thing.

4.10. How do I remember my pass phrase?

This can be quite a problem especially if you are like me and have about a dozen different pass phrases that are required in your every day life. Writing them down someplace so that you can remember them would defeat the whole purpose of pass phrases in the first place. There is really no good way around this. Either remember it, or write it down someplace and risk having it compromised.

4.11. How do I verify that my copy of PGP has not been tampered with?

If you do not presently own any copy of PGP, use great care on where you obtain your first copy. What I would suggest is that you get two or more copies from different sources that you feel that you can trust. Compare the copies to see if they are absolutely identical. This won't eliminate the possibility of having a bad copy, but it will greatly reduce the chances.

If you already own a trusted version of PGP, it is easy to check the validity of any future version. There is a file called PGPSIG.ASC included with all new releases. It is a stand-alone signature file for the contents of PGP.EXE. The signature file was created by the author of the program. Since nobody except the author has access to his secret key, nobody can tamper with either PGP.EXE or PGPSIG.ASC without it being detected. To check the signature, you MUST be careful that you are executing the OLD version of PGP to check the NEW. If not, the entire check is useless. Let's say that your existing copy of PGP is in subdirectory C:\PGP and your new copy is in C:\NEW. You should execute the following command:

```
\PGP\PGP C:\NEW\PGPSIG.ASC C:\NEW\PGP.EXE
```

This will force your old copy of PGP to be the one that is executed. If you simply changed to the C:\NEW directory and executed the command "PGP PGPSIG.ASC PGP.EXE" you would be using the new version to check itself, and this is an absolutely worthless check.

Once you have properly checked the signature of your new copy of PGP, you can copy all of the files to your C:\PGP directory.

4.12. How do I know that there is no trap door in the program?

The fact that the entire source code for PGP is available makes it just about impossible for there to be some hidden trap door. The source code has been examined by countless individuals and no such trap door has been found. To make sure that your executable file actually represents the given source code, all you need to do is to re-compile the entire program. I did this with the DOS version 2.3a and the Borland C++ 3.1 compiler and found that the output exactly matched byte for byte the distributed executable file.

4.13. Can I put PGP on a multi-user system like a network or a mainframe?

You can, but you should not, because this greatly reduces the security of your secret key/pass phrase. This is because your pass phrase may be passed over the network in the clear where it could be intercepted by network monitoring equipment. Also, while it is being used by PGP on the host system, it could be caught by some Trojan Horse program. Also, even though your secret key ring is encrypted, it would not be good practice to leave it lying around for anyone else to look at.

4.14. Why not use RSA alone rather than a hybrid mix of IDEA, MD5, & RSA?

Two reasons: First, the IDEA encryption algorithm used in PGP is actually MUCH stronger than RSA given

the same key length. Even with a 1024 bit RSA key, it is believed that IDEA encryption is still stronger, and, since a chain is no stronger than it's weakest link, it is believed that RSA is actually the weakest part of the RSA - IDEA approach. Second, RSA encryption is MUCH slower than IDEA. The only purpose of RSA in most public key schemes is for the transfer of session keys to be used in the conventional secret key algorithm, or to encode signatures.

4.15. Aren't all of these security procedures a little paranoid?

That all depends on how much your privacy means to you! Even apart from the government, there are many people out there who would just love to read your private mail. And many of these individuals would be willing to go to great lengths to compromise your mail. Look at the amount of work that has been put into some of the virus programs that have found their way into various computer systems. Even when it doesn't involve money, some people are obsessed with breaking into systems. Just about week ago, I saw a posting on alt.security.pgp where the return address had been altered to say "president@whitehouse.gov". In this case, the content of the message showed that it was obviously fake, but what about some of those other not so obvious cases.

4.16. Can I be forced to reveal my pass phrase in any legal proceedings?

The following information applies only to citizens of the United States in U.S. Courts. The laws in other countries may vary. Please see the disclaimer at the top of part 1.

There have been several threads on Internet concerning the question of whether or not the fifth amendment right about not being forced to give testimony against yourself can be applied to the subject of being forced to reveal your pass phrase. Not wanting to settle for the many conflicting opinions of armchair lawyers on usenet, I asked for input from individuals who were more qualified in the area. The results were somewhat mixed. There apparently has NOT been much case history to set precedence in this area. So if you find yourself in this situation, you should be prepared for a long and costly legal fight on the matter. Do you have the time and money for such a fight? Also remember that judges have great freedom in the use of "Contempt of Court". They might choose to lock you up until you decide to reveal the pass phrase and it could take your lawyer some time to get you out. (If only you just had a poor memory!)

5. Message Signatures

5.1. What is message signing?

Let's imagine that you received a letter in the mail from someone you know named John Smith. How do you know that John was really the person who sent you the letter and that someone else simply forged his name? With PGP, it is possible to apply a digital signature to a message that is impossible to forge. If you already have a trusted copy of John's public encryption key, you can use it to check the signature on the message. It would be impossible for anybody but John to have created the signature, since he is the only person with access to the secret key necessary to create the signature. In addition, if anybody has tampered with an otherwise valid message, the digital signature will detect the fact. It protects the entire message.

5.2. How do I sign a message while still leaving it readable?

Sometimes you are not interested in keeping the contents of a message secret, you only want to make sure that nobody tampers with it, and to allow others to verify that the message is really from you. For this, you can use clear signing. Clear signing only works on text files, it will NOT work on binary files. The command format is:

pgp -sat +clearsig=on <filename>

The output file will contain your original unmodified text, along with section headers and an armored PGP signature. In this case, PGP is not required to read the file, only to verify the signature.

6. Key Signatures

6.1. What is key signing?

OK, you just got a copy of John Smith's public encryption key. How do you know that the key really belongs to John Smith and not to some impostor? The answer to this is key signatures. They are similar to message signatures in that they can't be forged. Let's say that you don't know that you have John Smith's real key. But let's say that you DO have a trusted key from Joe Blow. Let's say that you trust Joe Blow and that he has added his signature to John Smith's key. By inference, you can now trust that you have a valid copy of John Smith's key. That is what key signing is all about. This chain of trust can be carried to several levels, such as A trusts B who trusts C who trusts D, therefore A can trust D. You have control in the PGP configuration file over exactly how many levels this chain of trust is allowed to proceed. Be careful about keys that are several levels removed from your immediate trust.

6.2. How do I sign a key?

From the command prompt, execute the following command:

PGP -ks [-u userid] <keyid>

A signature will be appended to already existing on the specified key. Next, you should extract a copy of this updated key along with its signatures using the "-kxa" option. An armored text file will be created. Give this file to the owner of the key so that he may propagate the new signature to whomever he chooses.

Be very careful with your secret keyring. Never be tempted to put a copy in somebody else's machine so you can sign their public key - they could have modified PGP to copy your secret key and grab your pass phrase.

It is not considered proper to send his updated key to a key server yourself unless he has given you explicit permission to do so. After all, he may not wish to have his key appear on a public server. By the same token, you should expect that any key that you give out will probably find its way onto the public key servers, even if you really didn't want it there, since anyone having your public key can upload it.

6.3. Should I sign my own key?

Yes, you should sign each personal ID on your key. This will help to prevent anyone from placing a phony address in the ID field of the key and possibly having your mail diverted to them. Anyone changing a user id to your key will be unable to sign the entry, making it stand out like a sore thumb since all of

the other entries are signed. Do this even if you are the only person signing your key. For example, my entry in the public key ring now appears as follows if you use the "-kvv" command:

```
Type bits/keyID   Date       User ID
pub  1024/90A9C9 1993/09/13  Gary Edstrom <gbe@netcom.com>
sig       90A9C9             Gary Edstrom <gbe@netcom.com>
                          Gary Edstrom <72677.564@compuserve.com>
sig       90A9C9             Gary Edstrom <gbe@netcom.com>
```

6.4. Should I sign X's key?

Signing someone's key is your indication to the world that you believe that key to rightfully belong to that person, and that person is who he purports to be. Other people may rely on your signature to decide whether or not a key is valid, so you should not sign capriciously.

Some countries require respected professionals such as doctors or engineers to endorse passport photographs as proof of identity for a passport application - you should consider signing someone's key in the same light. Alternatively, when you come to sign someone's key, ask yourself if you would be prepared to swear in a court of law as to that person's identity.

6.5. How do I verify someone's identity?

It all depends on how well you know them. Relatives, friends and colleagues are easy. People you meet at conventions or key-signing sessions require some proof like a driver's license or credit card.

6.6. How do I know someone hasn't sent me a bogus key to sign?

It is very easy for someone to generate a key with a false ID and send e-mail with fraudulent headers, or for a node which routes the e-mail to you to substitute a different key. Finger servers are harder to tamper with, but not impossible. The problem is that whilst public key exchange does not require a secure channel (eavesdropping is not a problem) it does require a tamper-proof channel (key-substitution is a problem).

If it is a key from someone you know well and whose voice you recognize then it is sufficient to give them a phone call and have them read their key's fingerprint (obtained with PGP -kvc <userid>).

If you don't know the person very well then the only recourse is to exchange keys face-to-face and ask for some proof of identity. Don't be tempted to put your public key disk in their machine so they can add their key - they could maliciously replace your key at the same time. If the user ID includes an e-mail address, verify that address by exchanging an agreed encrypted message before signing. Don't sign any user IDs on that key except those you have verified.

7. Revoking a key

7.1. My secret key ring has been stolen or lost, what do I do?

Assuming that you selected a good solid random pass phrase to encrypt your secret key ring, you are probably still safe. It takes two parts to decrypt a message, the secret key ring, and its pass phrase. Assuming you have a backup copy of your secret key ring, you should generate a key revocation certificate and upload the revocation to one of the public key servers. Prior to uploading the revocation certificate, you might add a new ID to the old key that tells what your new key ID will be. If you don't have a backup copy of your secret key ring, then it will be impossible to create a revocation certificate under

the present version of pgp. This is another good reason for keeping a backup copy of your secret key ring.

7.2. I forgot my pass phrase. Can I create a key revocation certificate?

YOU CAN'T, since the pass phrase is required to create the certificate! The way to avoid this dilemma is to create a key revocation certificate at the same time that you generate your key pair. Put the revocation certificate away in a safe place and you will have it available should the need arise. You need to be careful how you do this, however, or you will end up revoking the key pair that you just generated and a revocation can not be reversed. After you have generated your key pair initially, extract your key to an ASCII file using the -kxa option. Next, create a key revocation certificate and extract the revoked key to another ASCII file using the -kxa option again. Finally, delete the revoked key from your public key ring using the -kr option and put your non-revoked version back in the ring using the -ka option. Save the revocation certificate on a floppy so that you don't lose it if you crash your hard disk sometime.

8. Public Key Servers

8.1. What are the Public Key Servers?

Public Key Servers exist for the purpose of making your public key available in a common database where everybody can have access to it for the purpose of encrypting messages to you. While a number of key servers exist, it is only necessary to send your key to one of them. The key server will take care of the job of sending your key to all other known servers. As of 06-Dec-93 there are about 2,600 keys on the key servers. The rate of growth is increasing rapidly.

8.2. What public key servers are available?

The following is a list of all of the known public key servers active as of the publication date of this FAQ. I try to keep this list current by requesting keys from a different server every few days on a rotating basis. Any changes to this list should be posted to alt.security.pgp and a copy forwarded to me for inclusion in future releases of the PGP FAQ.

Changes:

24-Jan-94 Added message announcing WWW access to public keyserver on martigny.ai.mit.edu
24-Jan-94 Verified the existance of pgp-public-keys@sw.oz.au and corrected its address.
21-Jan-94 Added pgp-public-keys@ext221.sra.co.jp to list.
20-Jan-94 Added pgp-public-keys@kub.nl to list.
17-Jan-94 Added pgp-public-keys@jpunix.com to key servers no longer operational.

Internet sites:

pgp-public-keys@demon.co.uk
Mark Turner <mark@demon.co.uk>
FTP: ftp.demon.co.uk:/pub/pgp/pubring.pgp

Verified: 19-Jan-94

pgp-public-keys@fbihh.informatik.uni-hamburg.de
Vesselin V. Bontchev <bontchev@fbihh.informatik.uni-hamburg.de>
FTP: ftp.informatik.uni-hamburg.de:/pub/virus/misc/pubkring.pgp
Verified: 03-Jan-94

public-key-server@martigny.ai.mit.edu
Brian A. LaMacchia <public-key-server-request@martigny.ai.mit.edu>
FTP: None
Verified: 16-Jan-94

pgp-public-keys@pgp.ox.ac.uk
Paul Leyland <pcl@ox.ac.uk>
FTP: None
Verified: 18-Jan-94

pgp-public-keys@dsi.unimi.it
David Vincenzetti <vince@dsi.unimi.it>
FTP: ghost.dsi.unimi.it:/pub/crypt/public-keys.pgp
Verified: 18-Jan-94

pgp-public-keys@kub.nl
Teun Nijssen <teun@kub.nl>
FTP: None
Verified: 18-Jan-94

pgp-public-keys@ext221.sra.co.jp
Hironobu Suzuki <hironobu@sra.co.jp>
FTP: None
Verified: 20-Jan-94

pgp-public-keys@sw.oz.au
Jeremy Fitzhardinge <jeremy@sw.oz.au>
FTP: Unknown
Verified: 24-Jan-94

I have previously verified the existance of the following key server, but have been unable to reach it since the date indicated. If anyone has any information concerning it, please forward it to me so that I can update this list.

pgp-public-keys@kiae.su
FTP: Unknown
Last Attempt: 19-Jan-94
Last Verified: 11-Dec-93

The following key servers are no longer in operation:

pgp-public-keys@junkbox.cc.iastate.edu

pgp-public-keys@toxicwaste.mit.edu
pgp-public-keys@phil.utmb.edu
pgp-public-keys@pgp.iastate.edu
pgp-public-keys@jpunix.com

BBS sites:

Unknown

==========

From: bal@zurich.ai.mit.edu (Brian A. LaMacchia)
Newsgroups: alt.security.pgp
Subject: Announcing WWW access to public keyserver on martigny.ai.mit.edu
Date: 22 Jan 94 00:19:37

Announcing a new way to access public keyservers...

The public keyserver running on martigny.ai.mit.edu may now be accessed via a World Wide Web client with forms support (such as Mosaic). In your favorite WWW client, open the following URL to start:

http://martigny.ai.mit.edu/~bal/pks-toplev.html

Access to keys on the server is immediate. You can also submit new keys and/or signatures in ASCII-armored format to the server. New keys are processed every 10 minutes (along with server requests that arrive by e-mail).

The martigny.ai.mit.edu keyserver currently syncs directly with these other keyservers:

pgp-public-keys@demon.co.uk
pgp-public-keys@pgp.ox.ac.uk
pgp-public-keys@ext221.sra.co.jp
pgp-public-keys@kub.nl

NOTE! This service is experimental, and has limited options at present. I expect to be making changes to the server over the next few weeks to make it more useful. I would appreciate any bug reports, comments or suggestions you might have.

--Brian LaMacchia
bal@martigny.ai.mit.edu
public-key-server-request@martigny.ai.mit.edu

==========

8.3. What is the syntax of the key server commands?

The remailer expects to see one of the following commands placed in the subject field. Note that only the ADD command uses the body of the message.

ADD Your PGP public key (key to add is body of msg) (-ka)
INDEX List all PGP keys the server knows about (-kv)
VERBOSE INDEX List all PGP keys, verbose format (-kvv)
GET Get the whole public key ring (-kxa *)
GET <userid> Get just that one key (-kxa <userid>)
MGET <userid> Get all keys which match <userid>
LAST <n> Get all keys uploaded during last <n> days

If you wish to get the entire key ring and have access to FTP, it would be a lot more efficient to use FTP rather than e-mail. Using e-mail, the entire key ring can generate a many part message, which you will have to reconstruct into a single file before adding it to your key ring.

9. Bugs

> Where should I send bug reports?

Post all of your bug reports concerning PGP to alt.security.pgp and forward a copy to me for possible inclusion in future releases of the FAQ. Please be a aware that the authors of PGP might not acknowledge bug reports sent directly to them. Posting them on USENET will give them the widest possible distribution in the shortest amount of time. The following list of bugs is limited to version 2.2 and later. For bugs in earlier versions, refer to the documentation included with the program.

> Version 2.3 for DOS has a problem with clear signing messages. Anyone using version 2.3 for DOS should upgrade to version 2.3a.

> Version 2.2 for DOS has a problem of randomly corrupting memory, which can (and sometimes does) make DOS trash your hard disk.

10. Related News Groups

alt.privacy.clipper	Clipper, Capstone, Skipjack, Key Escrow
alt.security	general security discussions
alt.security.index	index to alt.security
alt.security.pgp	discussion of PGP
alt.security.ripem	discussion of RIPEM
alt.society.civil-liberty	general civil liberties, including privacy
comp.compression	discussion of compression algorithms
comp.org.eff.news	News reports from EFF
comp.org.eff.talk	discussion of EFF related issues
comp.patents	discussion of S/W patents, including RSA
comp.risks	some mention of crypto and wiretapping
comp.society.privacy	general privacy issues
comp.security.announce	announcements of security holes
misc.legal.computing	software patents, copyrights, computer laws
sci.crypt	methods of data encryption/decryption
sci.math	general math discussion
talk.politics.crypto	general talk on crypto politics

11. Recommended Reading

> The Code Breakers

The Story of Secret Writing
By David Kahn
The MacMillan Publishing Company (1968)
866 Third Avenue, New York, NY 10022
Library of Congress Catalog Card Number: 63-16109
ISBN: 0-02-560460-0

This has been the unofficial standard reference book on the history of cryptography for the last 25 years. It covers the development of cryptography from ancient times, up to 1967. It is interesting to read about the cat and mouse games that governments have been playing with each other even to this day. I have been informed by Mats Lofkvist <d87-mal@nada.kth.se> that the book has been reissued since its original printing. He found out about it from the 'Baker & Taylor Books' database. I obtained my original edition from a used book store. It is quite exhaustive in its coverage with 1164 pages. When I was serving in the United States Navy in the early 1970's as a cryptographic repair technician, this book was considered contraband and not welcome around my work place, even though it was freely available at the local public library. This was apparently because it mentioned several of the pieces of secret cryptographic equipment that were then in use in the military.

> The following list was taken from the PGP documentation:

Dorothy Denning, "Cryptography and Data Security", Addison-Wesley, Reading, MA 1982
Dorothy Denning, "Protecting Public Keys and Signature Keys", IEEE Computer, Feb 1983
Martin E. Hellman, "The Mathematics of Public-Key Cryptography," Scientific American, Aug 1979
Steven Levy, "Crypto Rebels", WIRED, May/Jun 1993, page 54. (This is a "must-read" article on PGP and other related topics.)
Ronald Rivest, "The MD5 Message Digest Algorithm", MIT Laboratory for Computer Science, 1991

Availible from the net as RFC1321.

Also avalible at ghost.dsi.unimi.it and it's mirror at nic.funet.fi:/pub/crypt/ghost.dsi.unimi.iti is:
IDEA_chapter.3.ZIP, a postscript text from the IDEA designer about IDEA.

Xuejia Lai, "On the Design and Security of Block Ciphers", Institute for Signal and Information Processing, ETH-Zentrum, Zurich, Switzerland, 1992
Xuejia Lai, James L. Massey, Sean Murphy, "Markov Ciphers and Differential Cryptanalysis", Advances in Cryptology- EUROCRYPT'91
Philip Zimmermann, "A Proposed Standard Format for RSA Cryptosystems", Advances in Computer Security, Vol III, edited by Rein Turn, Artech House, 1988
Bruce Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", John Wiley & Sons, 1993 (coming in November)
Paul Wallich, "Electronic Envelopes", Scientific American, Feb 1993, page 30. (This is an article on PGP)

12. General Tips

> Some BBS sysops may not permit you to place encrypted mail or files on their boards. Just because they have PGP in their file area, that doesn't necessarily mean they tolerate you uploading encrypted mail or files - so *do* check first.

> Fido net mail is even more sensitive. You should only send encrypted net mail after checking that:

a) Your sysop permits it.
b) Your recipient's sysop permits it.
c) The mail is routed through nodes whose sysops also permit it.

> Get your public key signed by as many individuals as possible. It increases the chances of another person finding a path of trust from himself to you.

> Don't sign someone's key just because someone else that you know has signed it. Confirm the identity of the individual yourself. Remember, you are putting your reputation on the line when you sign a key.

==========

Appendix I - PGP add-ons and Related Programs

==========

Much of this section was taken from an old FAQ supplied to me for the development of this list. This section will hopefully grow to contain a list of every utility that has been written. I would appreciate it if the authors of the various utilities could send me mail about their latest version, a description, if source code is available, and where to get it. I will then include the information in the next release of the FAQ.

If you have a utility, but don't know how to make it widely available, send mail to David Vincenzetti <vince@dsi.unimi.it> who is crypto collection maintainer at ghost.dsi.unimi.it. That ftp-site is weekly mirrored at nic.funet.fi in area: /pub/crypt/ghost.dsi.unimi.it

==========

> There are utilities in the source code for PGP. Get pgp23srcA.zip and unpack with 'pkunzip -d pgp23srcA.zip' to get them all come up nicely sorted in subdirectories.

Archimedes

> PGPwimp

From: Peter Gaunt
Current Version: 0.12
Where Available: ftp.demon.co.uk:/pub/archimedes
Information Updated: 21-Dec-93

A multi-tasking WIMP front-end for PGP (requires RISC OS 3). Operates on files - it has no hooks to allow integration with mailers/newsreaders.

> RNscripts4PGP

From: pla@sktb.demon.co.uk (Paul L. Allen)
Current Version: 1.1
Where Available: ftp.demon.co.uk:/pub/archimedes
Information Updated: 12-Dec-93

A collection of scripts and a small BASIC program which integrate PGP with the ReadNews mailer/newsreader. Provides encryp, decrypt, sign signature-check, add key.

DOS / MS Windows

> HPACK79 PGP-compatible archiver
  114243 Nov 20 07:08 garbo.uwasa.fi:/pc/arcers/hpack79.zip
  146470 Dec 3 01:01 garbo.uwasa.fi:/pc/doc-soft/hpack79d.zip
  511827 Dec 3 14:46 garbo.uwasa.fi:/pc/source/hpack79s.zip
  667464 Dec 5 16:43 garbo.uwasa.fi:/unix/arcers/hpack79src.tar.Z

  Where hpack79.zip is the MSDOS executable, hpack79d.zip is the Postscript documentation, hpack79s.zip is the source code, and hpack79src.tar.Z is the source code again but in tar.Z format (note that the latter is a tiny bit more recent that hpack79s.zip and contains changes for the NeXT). There is a (rather primitive) Macintosh executable somewhere on garbo as well, possibly /mac/arcers/hpack79mac.cpt. OS/2 32-bit versions of
  HPACK available for anonymous FTP from the UK. `ftp.demon.co.uk' [158.152.1.65] in ~/pub/ibmpc/pgp
  pgut1@cs.aukuni.ac.nz
  p_gutmann@cs.aukuni.ac.nz
  gutmann_p@kosmos.wcc.govt.nz
  peterg@kcbbs.gen.nz
  peter@nacjack.gen.nz
  peter@phlarnschlorpht.nacjack.gen.nz
  (In order of preference - one of 'ems bound to work)
> MENU.ZIP
  Menushell for MSDOS. (Requires 4DOS or Norton's NDOS) You can customize the menu for your own preferences. The name 'MENU' violates file naming conventions on ftp-sites, so I guess it's hard to find this program somewhere else. Exists at ghost.dsi.unimi.it area: /pub/crypt/ (ask archie about 4DOS, a comand.com replacement)
> PBBS (Scheduled for release summer 1994)
  Public Bulletin Board System (PBBS) ver 1.0 is a privacy-oriented host BBS application designed with the "anonymous movement's" diverse needs in mind. PBBS is a compact application at 75K, allowing it to be run off of a floppy disk if desired, and requires no telecommunications experience to operate. Installation of PBBS takes about 2 minutes flat, and is easy to set up and maintain. Don't let the size fool you however, it packs a powerful set of Zmodem, Ymodem, and Xmodem assembly-language protocols, supports speeds up to 57,600 bps, door support, full ANSI-emulation, and many more features!

  Public BBS is an eclectic and powerful BBS and also the first bulletin board system designed to work with Pretty Good Privacy (PGP), the public-key encryption program. A unique Post Office within PBBS allows users to send each other private "postcards" or to upload and download PGP-encrypted messages to other user's mail boxes. PBBS also contains a comprehensive public message base with "anonymous" read, write, and reply options. PBBS has a built in emergency self-destruct sequence for the sysop that desires an extra level of security. The ESD option will completely shred all PBBS-related files on disk, assuring the sysop that his or her BBS will not be compromised in any way. Look for Public BBS to be released on all Internet sites and FidoNet BBS's as PBBS10.ZIP. PBBS will change the face of cyber-fringe telecommunications forever! Questions or comments please e-mail James Still at <still@kailua.colorado.edu>.
> PGP-Front
  From: Walter H. van Holst <121233@pc-lab.fbk.eur.nl>
  Current Version:

Where Available: ghost.dsi.unimi.it:/pub/crypt
nic.funet.fi:/pub/crypt
Information Updated: 09-Jan-94

"PGP-Front is an interactive shell for Phill Zimmerman's Pretty Good Privacy and is available since November 1993 on some of the biggest FTP-sites. It features an easy to use interface for those who don't want to learn all PGP flags by heart but still want to make use of its versatility. The most used options of PGP are supported, including most key-management options. An improved version is under development and will feature support for some of the advanced options of PGP and a lot of extra configuration options for PGP-Front itself. System requirements for this beta-version:

- 80286 or better (will be lifted in version 1.00)
- MS/PC-DOS 3.11 or better
- Enough memory to run PGP plus an extra 512 bytes for PGP-Front, thanks to Ralph Brown.

Any feedback on this project will be appreciated,

Walter H. van Holst <121233@pc-lab.fbk.eur.nl>"

> PGP-NG.ZIP

At nic.funet.fi; /pub/crypt/pgp-ng.zip. A norton Guide database for PGP ver 2.0. Easy to find info for programmers about all the functions in the source code, and users can more easily find their subject. Is any update for the current version planned? Ask archie about the 2 Norton guide clones that are out on the net.

> PGPSHELL

Date: 12-Jan-94
From: James Still <still@kailua.colorado.edu>
Subject: PGPShell Version 3.0

FOR IMMEDIATE RELEASE

PGPSHELL VERSION 3.0 PROGRAM RELEASE

PGPShell, a front-end DOS program for use with Philip Zimmermann's Pretty Good Privacy (PGP) public-key encryption software, has just been upgraded and released as version 3.0.

PGPShell incorporates easy to use, mouse-driven menus and a unique Key Management Screen to easily display all public key ring information in a flash. PGP encryption will never be the same again! Breeze through PGP UserID's, KeyID's, Fingerprints, E-mail addresses, Signature's, Trust Parameter's, and PGP's Validity ratings all in one screen, at one place, and with a single mouse-click.

PGPShell is archived as pgpshe30.zip at many Internet sites including garbo.uwasa.fi:/pc/crypt and oak.oakland.edu:/pub/msdos/security and has been posted to the FidoNet Software Distribution Network (SDN) and should be on all nodes carrying SDN in a week or so.

To immediately acquire version 3.0 by modem you can call the Hieroglyphic Voodoo Machine BBS at +1

303 443 2457 or the GrapeVine BBS at +1 501 791 0124.

Questions or comments? Ping me at --> still@kailua.colorado.edu

> PGPUTILS.ZIP at ghost.dsi.unimi.it /pub/crypt/ is a collection of BAT-files, and PIF-files for windows.

> PGPWinFront (PFW20.ZIP)

Date: Thu, 13 Jan 1994 11:06:31 -0500 (EST)
From: Ross Barclay <RBARCLAY@TrentU.ca>
Subject: FAQ addition
To: gbe@netcom.com

Hello,

I have a program called PGPWinFront that is a Windows front-end for PGP. It is really quite good and has things like automatic message creation, key management, editable command line, one button access to PGP documentation, etc...

It is almost out in its second revision. It will be out on FTP sites very soon, and is available currently, and will always be available, by my automatic mail system.

If people send me (rbarclay@trentu.ca) a message with the subject GET PWF it will be sent to them, in PGP's radix-64 format. Like I said, it will also be available within the week on FTP sites. by the way my program is FREEWARE. Check it out if you like. If you use Windows, I think you'll find it very useful.

Ross Barclay Internet: Barclay@TrentU.Ca
Ontario, Canada CI$ (rarely): 72172,31
Send me a message with the subject GET KEY to get my PGP public key.

> Subject: Front End Announcement: PGP with TAPCIS
Sender: usenet@ttinews.tti.com (Usenet Admin)
Reply-To: 72027.3210@compuserve.com
Date: Tue, 3 Aug 1993 00:58:17 GMT

TAPCIS is a popular navigator/offline message reader used on PCs to access CompuServe. An add-on program, TAPPKE (TAPcis Public Key Encryption), has been uploaded to the CompuServe TAPCIS Support Forum library under "scripts and tools;" this program is an interface between TAPCIS message-writing facilities and PGP.

When you compose messages in TAPCIS, they get collected into a batch in a .SND file along with some control information about where and how the messages are to be posted or mailed; next time you go on-line to CompuServe, TAPCIS processes any messages waiting in its .SND files. The TAPPKE add-on can be run before you do this transmission step. TAPPKE scans messages in a .SND file, and any message that contains a keyword (##PRIVATE## or ##SIGNATURE##) is extracted and just that message is handed to PGP for encryption or signature, then reinserted into the .SND file for transmission.

All this is a simplified interface to make it more convenient to encrypt/sign messages while still using the normal (and familiar)message composition features of TAPCIS. TAPPKE doesn't do any encryption itself, it merely invokes an external encryption engine to perform the indicated tasks; you can even use it with encryption programs other than PGP if you set up a few environment variables so TAPPKE will know what encryption program to run and what command-line arguments to feed it. The default configuration assumes PGP.

I don't see any point in posting TAPPKE anywhere besides on CompuServe, since the only people who would have any use for it are TAPCIS users, and they by definition have access to the CompuServe TAPCIS forum libraries. However, it's free (I released it to the public domain, along with source code), so anyone who wants to propagate it is welcome to do so.

Some mailers apparently munge my address; you might have to use bsmart@bsmart.tti.com -- or if that fails, fall back to 72027.3210@compuserve.com. Ain't UNIX grand? "

> PWF12 A Windows front end for PGP
For all those MS Windows users who want a point and click PGP front end, PGP WinFront 1.2 (PWF12) is for you. This program is an easy to use Windows front end for PGP. You can access main PGP features more easily than from DOS. This program features:

> A simple file management system
> The ability to create plaintext files to encrypt very easily using the editor of your choice
> A quick way to shell to DOS to access esoteric PGP features
> Allows you to edit the command line to access the more specialised features of PGP
> Plus more

Check it out; IT'S FREE and available by email.

TO GET THIS PROGRAM (PWF12.ZIP):

1) Send an email message to rbarclay@trentu.ca
2) The subject MUST READ: GET PWF

3) The body can be left blank.

You will be sent a two part signed Radix-64 ASCII Armoured zip file. Use PGP to de-armour it. Read the document file fully. This program has a number of features not mentioned here and you wouldn't want to miss them.

--

ross barclay

MAC

Unix

> Emacs Auto-PGP 1.02

==========

This is a bunch of Elisp, Perl and C to allow you to integrate PGP2 (version 2.2 or later) into your Emacs mailreader (and perhaps also your newsreader).

Features:

- Scans the header of a message to be encrypted to determine the recipients and thus the keys to use to encrypt.
- Incoming encrypted messages can be decrypted once and then stored in plaintext, but ...
- Information about the recipient keys of an incoming encrypted message is preserved.
- Incoming signed and encrypted messages are turned into clearsigned messages (modulo some bugs/misfeatures in PGP).
- Signatures on incoming messages can be verified in place.
- You only have to type your passphrase once, but ...
- Your passphrase is not stored in your Emacs but in a separate small program which can easily be killed, or replaced (e.g. by an X client which pops up a window to confirm whether to supply the passphrase - though no such program exists yet (-:).
- The stored passphrase can easily be used when using pgp from the Unix command line by using the small wrapper program (which works just like normal pgp) which the scripts themselves use.
- No modification to the PGP sources necessary.

WARNING: You should probably not use this software if it is likely that an attacker could gain access to your account, for example because you are not the sysadmin or the security on your system is dubious (this is true of most networked Unix systems).

To install it:

Edit the file EDITME to reflect your situation, ie where you want stuff installed, whether you want to pick up a version from your PATH or run it via the explicit pathname, etc.

Type `make install'.

This should compile ringsearch and install the programs (using the scripts included) as you specified in EDITME.

Edit the `dir' file in the Emacs Info directory - add a menu item for Auto-PGP potining to the file `auto-pgp.info'.

Now read auto-pgp.info if you haven't done so already.

If you find a bug please READ THE SECTION ON REPORTING BUGS!

Ian Jackson <ijackson@nyx.cs.du.edu>

31st August 1993

> mailcrypt.el
From: jsc@mit.edu (Jin S Choi)
Current Version: 1.3
Where Available: gnu.emacs.sources
Info Updated: 21-Dec-93

This is an elisp package for encrypting and decrypting mail. I wrote this to provide a single interface to the two most common mail encryption programs, PGP and RIPEM. You can use either or both in any combination.

Includes:

VM mailreader support.
Support for addresses with spaces and <>'s in them.
Support for using an explicit path for the encryption executables.
Key management functions.
The ability to avoid some of the prompts when encrypting.
Assumes mc-default-scheme unless prefixed.
Includes menubar support under emacs 19 and gnus support.

> PGPPAGER ver. 1.1
Newsgroups: alt.security.pgp
From: abottone@minerva1.bull.it (Alessandro Bottonelli)
Subject: pgppager 1.1 sources
Date: Tue, 6 Jul 1993 11:37:06 GMT

pgppager, designed to be possibly integrated with elm mail reader. This programs reads from a specified file or from stdin if no file is specified and creates three temporary files i(header, encrypted, and trailer) as needed, in order to store the header portion in clear text, the encrypted portion still in cipher text, and the trailer portion of the clear text. Then, if applicable, the clear text header is outputted, the encrypted portion is piped through pgp as needed, then the trailer (if any) is outputted. THIS PROCESS IS TRANSPARENT TO NON PGP ENCRYPTED TEXTS

> rat-pgp.el
rat-pgp.el is a GNU Emacs interface to the PGP public key system. It lets you easily encrypt and decrypt message, sign messages with your secret key (to prove that it really came from you). It does signature verification, and it provides a number of other functions. The package is growing steadily as more is added. It is my intention that it will eventually allow as much functionality as accessing PGP directly. The most recent version of rat-pgp.el is always available via anonymous FTP at ftp.ccs.neu.edu, directory /pub/ratinox/emacs-lisp/rat-pgp.el.

VAX/VMS

> ENCRYPT.COM is a VMS mail script that works fine for joleary@esterh.wm.estec.esa.nl (John O'Leary)

==========

Appendix II - Glossary of Cryptographic Terms

==========

Chosen Plain Text Attack
This is the next step up from the Known Plain Text Attack. In this version, the cryptoanalysit can choose what

plain text message he wishes to encrypt and view the results, as opposed to simply taking any old plain text that he might happen to lay his hands on. If he can recover the key, he can use it to decode all data encrypted under this key. This is a much stronger form of attack than known plain text. The better encryption systems will resist this form of attack.

Clipper

A chip developed by the United States Government that was to be used as the standard chip in all encrypted communications. Aside from the fact that all details of how the Clipper chip work remain classified, the biggest concern was the fact that it has an acknowledged trap door in it to allow the government to eavesdrop on anyone using Clipper provided they first obtained a wiretap warrant. This fact, along with the fact that it can't be exported from the United States, has lead a number of large corporations to oppose the idea. Clipper uses an 80 bit key to perform a series of nonlinear transformation on a 64 bit data block.

DES (Data Encryption Standard)

A data encryption standard developed by the United States Government. It was criticized because the research that went into the development of the standard remained classified. Concerns were raised that there might be hidden trap doors in the logic that would allow the government to break anyone's code if they wanted to listen in. DES uses a 56 bit key to perform a series of nonlinear transformation on a 64 bit data block. Even when it was first introduced a number of years ago, it was criticized for not having a long enough key. 56 bits just didn't put it far enough out of reach of a brute force attack. Today, with the increasing speed of hardware and its falling cost, it would be feasible, to build a machine that could crack a 56 bit key in under a day's time. It is not known if such a machine has really been built, but the fact that it is feasible tends to weaken the security of DES substantially.

I would like to thank Paul Leyland <pcl@ox.ac.uk> for the following information relating to the cost of building such a DES cracking machine:

> _Efficient DES Key Search_
>
> At Crypto 93, Michael Wiener gave a paper with the above title. He showed how a DES key search engine could be built for $1 million which can do exhaustive search in 7 hours. Expected time to find a key from a matching pair of 64-bit plaintext and 64-bit ciphertext is 3.5 hours.
>
> So far as I can tell, the machine is scalable, which implies that a $100M machine could find keys every couple of minutes or so.
>
> The machine is fairly reliable: an error analysis implies that the mean time between failure is about 270 keys.
>
> The final sentence in the abstract is telling: In the light of this work, it would be prudent in many applications to use DES in triple-encryption mode.
>
> I only have portions of a virtually illegible FAX copy, so please don't ask me for much more detail. A complete copy of the paper is being snailed to me.
>
> Paul C. Leyland <pcl@ox.ac.uk>

Laszlo Baranyi <laszlo@instrlab.kth.se> says that the full paper is available in PostScript via ftp from:

> ftp.eff.org:/pub/crypto/des_key_search.ps
> cpsr.org:/cpsr/crypto/des/des_key_search.ps

cpsr.org also makes it available via their Gopher service.

EFF (Electronic Frontier Foundation)

The Electronic Frontier Foundation (EFF) was founded in July, 1990, to assure freedom of expression in digital media, with a particular emphasis on applying the principles embodied in the Constitution and the Bill of Rights to computer-based communication. For further information, contact:

Electronic Frontier Foundation
1001 G St., NW
Suite 950 East
Washington, DC 20001
+1 202 347 5400
+1 202 393 5509 FAX
Internet: eff@eff.org

IDEA (International Data Encryption Algorithm)

Developed in Switzerland and licensed for non commercial use in PGP. IDEA uses a 128 bit user supplied key to perform a series of nonlinear mathematical transformations on a 64 bit data block. Compare the length of this key with the 56 bits in DES or the 80 bits in Clipper.

ITAR (International Traffic in Arms Regulations)

ITAR are the regulations covering the exporting of weapons and weapons related technology from the United States. For some strange reason, the government claims that data encryption is a weapon and comes under the ITAR regulations. There is presently a move in congress to relax the section of ITAR dealing with cryptographic technology.

Known Plain Text Attack

A method of attack on a crypto system where the cryptoanalysit has matching copies of plain text, and its encrypted version. With weaker encryption systems, this can improve the chances of cracking the code and getting at the plain text of other messages where the plain text is not known.

MD5 (Message Digest Algorithm #5)

The message digest algorithm used in PGP is the MD5 Message Digest Algorithm, placed in the public domain by RSA Data Security, Inc. MD5's designer, Ronald Rivest, writes this about MD5:

> "It is conjectured that the difficulty of coming up with two messages having the same message digest is on the order of 2^64 operations, and that the difficulty of coming up with any message having a given message digest is on the order of 2^128 operations. The MD5 algorithm has been carefully scrutinized for weaknesses. It is, however, a relatively new algorithm and further security analysis is of course justified, as is the case with any new proposal of this sort. The level of security provided by MD5 should be sufficient for implementing very high security hybrid digital signature schemes based on MD5 and the RSA public-key cryptosystem."

NSA (National Security Agency)

The following was lifted unedited except for formatting from the sci.crypt FAQ:

The NSA is the official communications security body of the U.S. government. It was given its charter by President Truman in the early 50's, and has continued research in cryptology till the present. The NSA is known to be the largest employer of mathematicians in the world, and is also the largest purchaser of computer hardware in the world. Governments in general have always been prime employers of cryptologists. The NSA probably possesses cryptographic expertise many years ahead of the public state of the art, and can undoubtedly break many of the systems used in practice; but for reasons of national security almost all information about the

NSA is classified.

One Time Pad

The one time pad is the ONLY encryption scheme that can be proven to be absolutely unbreakable! It is used extensively by spies because it doesn't require any hardware to implement and because of its absolute security. This algorithm requires the generation of many sets of matching encryption keys pads. Each pad consists of a number of random key characters. These key characters are chosen completely at random using some truly random process. They are NOT generated by any kind of cryptographic key generator. Each party involved receives matching sets of pads. Each key character in the pad is used to encrypt one and only one plain text character, then the key character is never used again. Any violation of these conditions negates the perfect security available in the one time pad.

So why don't we use the one time pad all the time? The answer is that the number of random key pads that need to be generated must be at least equal to the volume of plain text messages to be encrypted, and the fact that these key pads must somehow be exchanged ahead of time. This becomes totally impractical in modern high speed communications systems.

Among the more famous of the communications links using a one time pad scheme is the Washington to Moscow hot line.

PEM (Privacy Enhanced Mail)

The following was taken from the sci.crypt FAQ:

How do I send encrypted mail under UNIX? [PGP, RIPEM, PEM, ...]?

Here's one popular method, using the des command:

cat file | compress | des private_key | uuencode | mail

Meanwhile, there is a de jure Internet standard in the works called PEM (Privacy Enhanced Mail). It is described in RFCs 1421 through 1424. To join the PEM mailing list, contact pem-dev-request@tis.com. There is a beta version of PEM being tested at the time of this writing.

There are also two programs available in the public domain for encrypting mail: PGP and RIPEM. Both are available by FTP. Each has its own news group: alt.security.pgp and alt.security.ripem. Each has its own FAQ as well. PGP is most commonly used outside the USA since it uses the RSA algorithm without a license and RSA's patent is valid only (or at least primarily) in the USA.

RIPEM is most commonly used inside the USA since it uses the RSAREF which is freely available within the USA but not available for shipment outside the USA.

Since both programs use a secret key algorithm for encrypting the body of the message (PGP used IDEA; RIPEM uses DES) and RSA for encrypting the message key, they should be able to interoperate freely. Although there have been repeated calls for each to understand the other's formats and algorithm choices, no interoperation is available at this time (as far as we know).

PGP (Pretty Good Privacy)

PKP (Public Key Partners)

Claim to have a patent on RSA.

RIPEM

See PEM

RSA (Rivest-Shamir-Adleman)

RSA is the public key encryption method used in PGP. RSA are the initials of the developers of the algorithm which was done at tax payer expense. The basic security in RSA comes from the fact that, while it is relatively easy to multiply two huge prime numbers together to obtain their product, it is computationally difficult to go the reverse direction: to find the two prime factors of a given composite number. It is this one-way nature of RSA that allows an encryption key to be generated and disclosed to the world, and yet not allow a message to be decrypted.

Skipjack

See Clipper

TEMPEST

TEMPEST is a standard for electromagnetic shielding for computer equipment. It was created in response to the fact that information can be read from computer radiation (e.g., from a CRT) at quite a distance and with little effort. Needless to say, encryption doesn't do much good if the cleartext is available this way. The typical home computer WOULD fail ALL of the TEMPEST standards by a long shot. So, if you are doing anything illegal, don't expect PGP or any other encryption program to save you. The government could just set up a monitoring van outside your home and read everything that you are doing on your computer.

Short of shelling out the ten thousand dollars or so that it would take to properly shield your computer, a good second choice might be a laptop computer running on batteries. No emissions would be fed back into the power lines, and the amount of power being fed to the display and being consumed by the computer is much less than the typical home computer and CRT. This provides a much weaker RF field for snoopers to monitor. It still isn't safe, just safer. In addition, a laptop computer has the advantage of not being anchored to one location. Anyone trying to monitor your emissions would have to follow you around, maybe making themselves a little more obvious. I must emphasize again that a laptop still is NOT safe from a tempest standpoint, just safer than the standard personal computer.

==========

Appendix III - Cypherpunks

==========

> What are Cypherpunks?

> What is the cypherpunks mailing list?

Eric Hughes <hughes@toad.com> runs the "cypherpunk" mailing list dedicated to "discussion about technological defenses for privacy in the digital domain." Frequent topics include voice and data encryption, anonymous remailers, and the Clipper chip. Send e-mail to cypherpunks-request@toad.com to be added or subtracted from the list. The mailing list itself is cypherpunks@toad.com. You don't need to be a member of the list in order to send messages to it, thus allowing the use of anonymous remailers to post your more sensitive messages that you just as soon would not be credited to you. (Traffic is sometimes up to 30-40 messages per day.)

> What is the purpose of the Cypherpunk remailers?

The purpose of these remailers is to take privacy one level further. While a third party who is snooping on the net may not be able to read the encrypted mail that you are sending, he is still able to know who you are sending mail to. This could possibly give him some useful information. This is called traffic flow analysis. To counter this type of attack, you can use a third party whose function is simply to remail your message with his return address on it instead of yours.

Two types of remailers exist. The first type only accepts plain text remailing headers. This type would only

be used if your goal was only to prevent the person to whom your are sending mail from learning your identity. It would do nothing for the problem of net eavesdroppers from learning to whom you are sending mail.

The second type of remailer accepts encrypted remailing headers. With this type of remailer, you encrypt your message twice. First, you encrypt it to the person ultimately receiving the message. You then add the remailing header and encrypt it again using the key for the remailer that you are using. When the remailer receives your message, the system will recognize that the header is encrypted and will use its secret decryption key to decrypt the message. He can now read the forwarding information, but because the body of the message is still encrypted in the key of another party, he is unable to read your mail. He simply remails the message to the proper destination. At its ultimate destination, the recipient uses his secret to decrypt this nested encryption and reads the message.

Since this process of multiple encryptions and remailing headers can get quite involved, there are several programs available to simplify the process. FTP to soda.berkeley.edu and examine the directory /pub/cypherpunks/remailers for the programs that are available.

> Where are the currently active Cypherpunk remailers?

Any additions, deletions, or corrections to the following list should be posted on alt.security.pgp and forwarded to me for inclusion in a future release of the FAQ. The number appearing in the first column has the following meaning:

1: Remailer accepts only plain text headers.
2: Remailer accepts both plain text and encrypted headers.
3: Remailer accepts only encrypted headers.

Only remailers whose operational status has been verified by me appear on this list. Remember, however, that this list is subject to change quite often. Always send yourself a test message through the Remailer before starting to use it for real.

1 hh@pmantis.berkeley.edu
1 hh@cicada.berkeley.edu
1 hh@soda.berkeley.edu
hh@soda.berkeley.edu also supports these header commands:
Post-To: <USENET GROUP(S)> (Regular posting to USENET)
Anon-Post-To: <USENET GROUP(S)> (Anonymous posting to USENET)
1 nowhere@bsu-cs.bsu.edu
1 remail@tamsun.tamu.edu
2 ebrandt@jarthur.claremont.edu
2 hal@alumni.caltech.edu [Fwd: hfinney@shell.portal.com]
2 elee7h5@rosebud.ee.uh.edu
2 hfinney@shell.portal.com
2 remailer@utter.dis.org
1 00x@uclink.berkeley.edu [Fwd: hh@soda.berkeley.edu]
2 remailer@rebma.mn.org
3 remail@extropia.wimsey.com

The following former Cypherpunk remailers are no longer in service. Either a message stating that the system had been shutdown was received, or the test message was returned due to an invalid address, or no test message was returned after three attempts.

phantom@mead.u.washington.edu [Shutdown message returned]
remail@tamaix.tamu.edu [Mail returned, invalid address]

> Are there other anonymous remailers besides the cypherpunk remailers?

Yes, the most commonly used remailer on the Internet is in Finland. It is known as anon.penet.fi. The syntax for sending mail through this remailer is different from the cypherpunk remailers. For example, if you wanted to send mail to me (gbe@netcom.com) through anon.penet.fi, you would send the mail to "gbe %netcom.com@anon.penet.fi". Notice that the "@" sign in my Internet address is changed to a "%". Unlike the cypherpunk remailers, anon.penet.fi directly supports anonymous return addresses. Anybody using the remailer is assigned an anonymous id of the form "an?????" where "?????" is filled in with a number representing that user. To send mail to someone when you only know their anonymous address, address your mail to "an????? @anon.penet.fi" replacing the question marks with the user id you are interested in. For additional information on anon.penet.fi, send a blank message to "help@anon.penet.fi". You will receive complete instructions on how to use the remailer, including how to obtain a pass phrase on the system.

> Where can I learn more about Cypherpunks?

FTP: soda.berkeley.edu Directory: /pub/cypherpunks

> What is the command syntax?

The first non blank line in the message must start with two colons (::). The next line must contain the user defined header "Request-Remailing-To: <destination>". This line must be followed by a blank line. Finally, your message can occupy the rest of the space. As an example, if you wanted to send a message to me via a remailer , you would compose the following message:

::
Request-Remailing-To: gbe@netcom.com

[body of message]

You would then send the above message to the desired remailer. Note the section labeled "body of message" may be either a plain text message, or an encrypted and armored PGP message addressed to the desired recipient. To send the above message with an encrypted header, use PGP to encrypt the entire message shown above to the desired remailer. Be sure to take the output in armored text form. In front of the BEGIN PGP MESSAGE portion of the file, insert two colons (::) as the first non-blank line of the file. The next line should say "Encrypted: PGP". Finally the third line should be blank. The message now looks as follows:

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version 2.3a

[body of pgp message]
-----END PGP MESSAGE-----

You would then send the above message to the desired remailer just as you did in the case of the non-encrypted header. Note that it is possible to chain remailers together so that the message passes through several levels of anonymity before it reaches its ultimate destination.

==========

Appendix IV - How to obtain articles from Wired Magazine

==========

Greetings from the WIRED INFOBOT!

This file provides both an index to some general Wired information files and instructions for getting specific listings of the articles from back issues of Wired via email.

* * *

Wired General Information Files

* * *

To retrieve the following files, send an email message to infobot@wired.com containing the word "get" or "send," followed by the name of the file, in the body of the message. For instance, to retrieve the submission guide for Wired writers, you would send a message to the InfoBot containing the following line:

```
send writers.guidelines
```

The files will be returned to you via email.

For more information, see the Help file, which can be obtained by sending a message to the InfoBot containing the following line:

```
help
```

General information files currently available from the Wired InfoBot include the following:

File	Description
index	This file
writers.guidelines	Submissions guide for writers
ad.rates	Advertising rates and other details
visions	New Voices, New Visions 1994
wired.wonders	Seven Wired Wonders article (Wired 1.6) plus some additional Wired Wonders not listed in print.

* * *

Retrieving Files from Previous Issues of Wired

* * *

To retrieve files from back issues of Wired, you first need to retrieve

the index of the files contained in those issues. In order to make file size more manageable, there are two index files per issue, one for regular _Wired_ departments (such as Street Cred, Electric Word, and Electrosphere), and one for feature articles specific to that issue.

To order an index, send a message to the Wired InfoBot containing the "get" or "send" command, followed by the issue number, a "slash" character ("/"), either the keyword "departments" or "features", another "slash" character ("/"), and the word "index".

For those of you who like reading DOS or UNIX manuals, the general case command looks a little something like this:

send <issue number>/[departments][features]/index

For those of you who prefer real examples, if, for instance, you wanted to order the index to all the feature articles in Wired 1.2, you would send the command

send 1.2/features/index

and to get the index to the regular Wired departments in issue 1.3, you would send the command

send 1.3/departments/index

Once you have received the index, you can order specific articles by simply substituting the keyword for that article for the word "index" in the above commands. Thus, to order the Street Cred section of issue 1.3, you would send the command

send 1.3/departments/street-cred

and to get Bruce Sterling's Virtual War article from issue 1.1, you would use the command

send 1.1/features/virtwar

Got it? Great! Happy reading...

* * *

Getting help from a Real Human Being

* * *

We at Wired understand that using any new technology can be frustrating.

If you have any problems using the Wired InfoBot, please send mail to the Wired InfoBeing (infoman@wired.com), the real human assigned the task of maintaining this service. Please be patient with the InfoBeing, as it is also responsible for other important tasks here at Wired. For instance, if you send a message to the InfoBeing but do not receive a follow-up, please wait *at least* 24 hours (and hopefully longer) before sending any additional messages.

We here at Wired Online look forward to expanding our services. If you have questions or comments regarding this service or others we should offer, please address them to online@wired.com.

Thanks for your support!!!

--all us folks at Wired Online--

==========

Appendix V - Testimony of Philip Zimmermann to Congress.
Reproduced by permission.

==========

From netcom.com!netcomsv!decwrl!sdd.hp.com!col.hp.com!csn!yuma!ld231782 Sun Oct 10 07:55:51 1993
Xref: netcom.com talk.politics.crypto:650 comp.org.eff.talk:20832 alt.politics.org.nsa:89
Newsgroups: talk.politics.crypto,comp.org.eff.talk,alt.politics.org.nsa
Path: netcom.com!netcomsv!decwrl!sdd.hp.com!col.hp.com!csn!yuma!ld231782
From: ld231782@LANCE.ColoState.Edu (L. Detweiler)
Subject: ZIMMERMANN SPEAKS TO HOUSE SUBCOMMITTEE
Sender: news@yuma.ACNS.ColoState.EDU (News Account)
Message-ID: <Oct10.044212.45343@yuma.ACNS.ColoState.EDU>
Date: Sun, 10 Oct 1993 04:42:12 GMT
Nntp-Posting-Host: turner.lance.colostate.edu
Organization: Colorado State University, Fort Collins, CO 80523
Lines: 281

Date: Sat, 9 Oct 93 11:57:54 MDT
From: Philip Zimmermann <prz@acm.org>
Subject: Zimmerman testimony to House subcommittee

Testimony of Philip Zimmermann to
Subcommittee for Economic Policy, Trade, and the Environment
US House of Representatives
12 Oct 1993

Mr. Chairman and members of the committee, my name is Philip Zimmermann, and I am a software engineer who specializes in cryptography and data security. I'm here to talk to you today about the need to change US export control policy for cryptographic software. I want to thank you for the opportunity to be here and commend you for your attention to this important issue.

I am the author of PGP (Pretty Good Privacy), a public-key encryption software package for the protection of electronic mail. Since PGP was published domestically as freeware in June of 1991, it has spread organically all over the world and has since become the de facto worldwide standard for encryption of E-mail. The US Customs Service is investigating how PGP spread outside the US. Because I am a target of this ongoing criminal investigation, my lawyer has advised me not to answer any questions related to the investigation.

I. The information age is here.

Computers were developed in secret back in World War II mainly to break codes. Ordinary people did not have access to computers, because they were few in number and too expensive. Some people postulated that there would never be a need for more than half a dozen computers in the country. Governments formed their attitudes toward cryptographic technology during this period. And these attitudes persist today. Why would ordinary people need to have access to good cryptography?

Another problem with cryptography in those days was that cryptographic keys had to be distributed over secure channels so that both parties could send encrypted traffic over insecure channels. Governments solved that problem by dispatching key couriers with satchels handcuffed to their wrists. Governments could afford to send guys like these to their embassies overseas. But the great masses of ordinary people would never have access to practical cryptography if keys had to be distributed this way. No matter how cheap and powerful personal computers might someday become, you just can't send the keys electronically without the risk of interception.

This widened the feasibility gap between Government and personal access to cryptography.

Today, we live in a new world that has had two major breakthroughs that have an impact on this state of affairs. The first is the coming of the personal computer and the information age. The second breakthrough is public-key cryptography.

With the first breakthrough comes cheap ubiquitous personal computers, modems, FAX machines, the Internet, E-mail, digital cellular phones, personal digital assistants (PDAs), wireless digital networks, ISDN, cable TV, and the data superhighway. This information revolution is catalyzing the emergence of a global economy.

But this renaissance in electronic digital communication brings with it a disturbing erosion of our privacy. In the past, if the Government wanted to violate the privacy of ordinary citizens, it had to expend a certain amount of effort to intercept and steam open and read paper mail, and listen to and possibly transcribe spoken telephone conversation. This is analogous to catching fish with a hook and a line, one fish at a time. Fortunately for freedom and democracy, this kind of labor-intensive monitoring is not practical on a large scale.

Today, electronic mail is gradually replacing conventional paper mail, and is soon to be the norm for everyone, not the novelty is is today. Unlike paper mail, E-mail messages are just too easy to intercept and scan for interesting keywords. This can be done easily, routinely, automatically, and undetectably on a grand scale. This is analogous to driftnet fishing-- making a quantitative and qualitative Orwellian difference to the health of democracy.

The second breakthrough came in the late 1970s, with the mathematics of public key cryptography. This allows people to communicate securely and conveniently with people they've never met, with no prior exchange of keys over secure channels. No more special key couriers with black bags. This, coupled with the trappings of the information age, means the great masses of people can at last use cryptography. This new technology also provides digital signatures to authenticate transactions and messages, and allows for digital money, with all the implications that has for an electronic digital economy. (See appendix)

This convergence of technology-- cheap ubiquitous PCs, modems, FAX, digital phones, information superhighways, et cetera-- is all part of

the information revolution. Encryption is just simple arithmetic to all this digital hardware. All these devices will be using encryption. The rest of the world uses it, and they laugh at the US because we are railing against nature, trying to stop it. Trying to stop this is like trying to legislate the tides and the weather. It's like the buggy whip manufacturers trying to stop the cars-- even with the NSA on their side, it's still impossible. The information revolution is good for democracy-- good for a free market and trade. It contributed to the fall of the Soviet empire. They couldn't stop it either.

Soon, every off-the-shelf multimedia PC will become a secure voice telephone, through the use of freely available software. What does this mean for the Government's Clipper chip and key escrow systems?

Like every new technology, this comes at some cost. Cars pollute the air. Cryptography can help criminals hide their activities. People in the law enforcement and intelligence communities are going to look at this only in their own terms. But even with these costs, we still can't stop this from happening in a free market global economy. Most people I talk to outside of Government feel that the net result of providing privacy will be positive.

President Clinton is fond of saying that we should "make change our friend". These sweeping technological changes have big implications, but are unstoppable. Are we going to make change our friend? Or are we going to criminalize cryptography? Are we going to incarcerate our honest, well-intentioned software engineers?

Law enforcement and intelligence interests in the Government have attempted many times to suppress the availability of strong domestic encryption technology. The most recent examples are Senate Bill 266 which mandated back doors in crypto systems, the FBI Digital Telephony bill, and the Clipper chip key escrow initiative. All of these have met with strong opposition from industry and civil liberties groups. It is impossible to obtain real privacy in the information age without good cryptography.

The Clinton Administration has made it a major policy priority to help build the National Information Infrastructure (NII). Yet, some elements of the Government seems intent on deploying and entrenching a communications infrastructure that would deny the citizenry the ability to protect its privacy. This is unsettling because in a democracy, it is possible for bad people to occasionally get elected-- sometimes very bad people. Normally, a well-functioning democracy has ways to remove these people from power. But the wrong

technology infrastructure could allow such a future government to watch every move anyone makes to oppose it. It could very well be the last government we ever elect.

When making public policy decisions about new technologies for the Government, I think one should ask oneself which technologies would best strengthen the hand of a police state. Then, do not allow the Government to deploy those technologies. This is simply a matter of good civic hygiene.

II. Export controls are outdated and are a threat to privacy and economic competitivness.

The current export control regime makes no sense anymore, given advances in technology.

There has been considerable debate about allowing the export of implementations of the full 56-bit Data Encryption Standard (DES). At a recent academic cryptography conference, Michael Wiener of Bell Northern Research in Ottawa presented a paper on how to crack the DES with a special machine. He has fully designed and tested a chip that guesses DES keys at high speed until it finds the right one. Although he has refrained from building the real chips so far, he can get these chips manufactured for $10.50 each, and can build 57000 of them into a special machine for $1 million that can try every DES key in 7 hours, averaging a solution in 3.5 hours. $1 million can be hidden in the budget of many companies. For $10 million, it takes 21 minutes to crack, and for $100 million, just two minutes. That's full 56-bit DES, cracked in just two minutes. I'm sure the NSA can do it in seconds, with their budget. This means that DES is now effectively dead for purposes of serious data security applications. If Congress acts now to enable the export of full DES products, it will be a day late and a dollar short.

If a Boeing executive who carries his notebook computer to the Paris airshow wants to use PGP to send email to his home office in Seattle, are we helping American competitivness by arguing that he has even potentially committed a federal crime?

Knowledge of cryptography is becoming so widespread, that export controls are no longer effective at controlling the spread of this technology. People everywhere can and do write good cryptographic software, and we import it here but cannot export it, to the detriment of our indigenous software industry.

I wrote PGP from information in the open literature, putting it into a convenient package that everyone can use in a desktop or palmtop computer. Then I gave it away for free, for the good of our democracy. This could have popped up anywhere, and spread. Other people could have and would have done it. And are doing it. Again and again. All over the planet. This technology belongs to everybody.

III. People want their privacy very badly.

PGP has spread like a prairie fire, fanned by countless people who fervently want their privacy restored in the information age.

Today, human rights organizations are using PGP to protect their people overseas. Amnesty International uses it. The human rights group in the American Association for the Advancement of Science uses it.

Some Americans don't understand why I should be this concerned about the power of Government. But talking to people in Eastern Europe, you don't have to explain it to them. They already get it-- and they don't understand why we don't.

I want to read you a quote from some E-mail I got last week from someone in Latvia, on the day that Boris Yeltsin was going to war with his Parliament:

> "Phil I wish you to know: let it never be, but if dictatorship takes over Russia your PGP is widespread from Baltic to Far East now and will help democratic people if necessary. Thanks."

Appendix -- How Public-Key Cryptography Works

In conventional cryptosystems, such as the US Federal Data Encryption Standard (DES), a single key is used for both encryption and decryption. This means that a key must be initially transmitted via secure channels so that both parties have it before encrypted messages can be sent over insecure channels. This may be inconvenient. If you have a secure channel for exchanging keys, then why do you need cryptography in the first place?

In public key cryptosystems, everyone has two related complementary

keys, a publicly revealed key and a secret key. Each key unlocks the code that the other key makes. Knowing the public key does not help you deduce the corresponding secret key. The public key can be published and widely disseminated across a communications network. This protocol provides privacy without the need for the same kind of secure channels that a conventional cryptosystem requires.

Anyone can use a recipient's public key to encrypt a message to that person, and that recipient uses her own corresponding secret key to decrypt that message. No one but the recipient can decrypt it, because no one else has access to that secret key. Not even the person who encrypted the message can decrypt it.

Message authentication is also provided. The sender's own secret key can be used to encrypt a message, thereby "signing" it. This creates a digital signature of a message, which the recipient (or anyone else) can check by using the sender's public key to decrypt it. This proves that the sender was the true originator of the message, and that the message has not been subsequently altered by anyone else, because the sender alone possesses the secret key that made that signature. Forgery of a signed message is infeasible, and the sender cannot later disavow his signature.

These two processes can be combined to provide both privacy and authentication by first signing a message with your own secret key, then encrypting the signed message with the recipient's public key. The recipient reverses these steps by first decrypting the message with her own secret key, then checking the enclosed signature with your public key. These steps are done automatically by the recipient's software.

--

Philip Zimmermann
3021 11th Street
Boulder, Colorado 80304
303 541-0140
E-mail: prz@acm.org

--

ld231782@longs.LANCE.ColoState.EDU

==========
Appendix VI - Anouncement of Philip Zimmermann Defense Fund.
Reproduced by permission.
==========

From prz@columbine.cgd.ucar.EDU Thu Oct 14 23:16:32 1993
Return-Path: <prz@columbine.cgd.ucar.EDU>
Received: from ncar.ucar.edu by mail.netcom.com (5.65/SMI-4.1/Netcom)
id AA05680; Thu, 14 Oct 93 23:16:29 -0700
Received: from sage.cgd.ucar.edu by ncar.ucar.EDU (5.65/ NCAR Central Post Office 03/11/93)
id AA01642; Fri, 15 Oct 93 00:15:34 MDT
Received: from columbine.cgd.ucar.edu by sage.cgd.ucar.EDU (5.65/ NCAR Mail Server 04/10/90)
id AA22977; Fri, 15 Oct 93 00:14:08 MDT
Message-Id: <9310150616.AA09815@columbine.cgd.ucar.EDU>
Received: by columbine.cgd.ucar.EDU (4.1/ NCAR Mail Server 04/10/90)
id AA09815; Fri, 15 Oct 93 00:16:57 MDT
Subject: PGP legal defense fund
To: gbe@netcom.com (Gary Edstrom)
Date: Fri, 15 Oct 93 0:16:56 MDT
From: Philip Zimmermann <prz@columbine.cgd.ucar.EDU>
In-Reply-To: <9310112013.AA07737@netcom5.netcom.com>; from "Gary Edstrom" at Oct 11, 93 1:13 pm
From: Philip Zimmermann <prz@acm.org>
Reply-To: Philip Zimmermann <prz@acm.org>
X-Mailer: ELM [version 2.3 PL0]
Status: OR

Date: Fri, 24 Sep 1993 02:41:31 -0600 (CDT)
From: hmiller@orion.it.luc.edu (Hugh Miller)
Subject: PGP defense fund

As you may already know, on September 14 LEMCOM Systems (ViaCrypt) in Phoenix, Arizona was served with a subpoena issued by the US District Court of Northern California to testify before a grand jury and produce documents related to "ViaCrypt, PGP, Philip Zimmermann, and anyone or any entity acting on behalf of Philip Zimmermann for the time period June 1, 1991 to the present."

Phil Zimmermann has been explicitly told that he is the primary target of the investigation being mounted from the San Jose office of U.S. Customs. It is not known if there are other targets. Whether or not an indictment is returned in this case, the legal bills will be astronomical.

If this case comes to trial, it will be one of the most important cases in recent times dealing with cryptography, effective communications privacy, and the free flow of information and ideas in cyberspace in the post-Cold War political order. The stakes are high, both for those of us who support the idea of effective personal

communications privacy and for Phil, who risks jail for his selfless and successful effort to bring to birth "cryptography for the masses," a.k.a. PGP. Export controls are being used as a means to curtail domestic access to effective cryptographic tools: Customs is taking the position that posting cryptographic code to the Internet is equivalent to exporting it. Phil has assumed the burden and risk of being the first to develop truly effective tools with which we all might secure our communications against prying eyes, in a political environment increasingly hostile to such an idea -- an environment in which Clipper chips and Digital Telephony bills are our own government's answer to our concerns. Now is the time for us all to step forward and help shoulder that burden with him.

Phil is assembling a legal defense team to prepare for the possibility of a trial, and he needs your help. This will be an expensive affair, and the meter is already ticking. I call on all of us, both here in the U.S. and abroad, to help defend Phil and perhaps establish a groundbreaking legal precedent. A legal trust fund has been established with Phil's attorney in Boulder. Donations will be accepted in any reliable form, check, money order, or wire transfer, and in any currency. Here are the details:

To send a check or money order by mail, make it payable, NOT to Phil Zimmermann, but to Phil's attorney, Philip Dubois. Mail the check or money order to the following address:

Philip Dubois
2305 Broadway
Boulder, CO USA 80304
(Phone #: 303-444-3885)

To send a wire transfer, your bank will need the following information:

Bank: VectraBank
Routing #: 107004365
Account #: 0113830
Account Name: "Philip L. Dubois, Attorney Trust Account"

Any funds remaining after the end of legal action will be returned to named donors in proportion to the size of their donations.

You may give anonymously or not, but PLEASE - give generously. If you admire PGP, what it was intended to do and the ideals which animated its creation, express your support with a contribution to this fund.

Posted to: alt.security.pgp; sci.crypt; talk.politics.crypto;
comp.org.eff.talk; comp.society.cu-digest; comp.society; alt.sci.sociology;
alt.security.index; alt.security.keydist; alt.security;
alt.society.civil-liberty; alt.society.civil-disob; alt.society.futures

--

Hugh Miller | Asst. Prof. of Philosophy | Loyola University Chicago
FAX: 312-508-2292 | Voice: 312-508-2727 | hmiller@lucpul.it.luc.edu
PGP 2.3A Key fingerprint: FF 67 57 CC 0C 91 12 7D 89 21 C7 12 F7 CF C5 7E

==========

Appendix VII - A Statement from ViaCrypt Concerning ITAR
Reproduced by Permission

==========

-----BEGIN PGP SIGNED MESSAGE-----The ITAR (International Traffic in Arms Regulations) includesa regulation that requires a manufacturer of cryptographicproducts to register with the U.S. State Department even if themanufacturer has no intentions of exporting products. It appears that this particular regulation is either not widely known, or is widely ignored.While no pressure was placed upon ViaCrypt to register, it is the Company's position to comply with all applicable laws and regulations.In keeping with this philosophy, ViaCrypt has registered with theU.S. Department of State as a munitions manufacturer.-----BEGIN PGP SIGNATURE-----
Version:
2.4iQCVAgUBLQ+DfmhHpCDLdoUBAQGa+AP/YzLpHBGOgsU4b7DjLYj8KFC4FFACryRJCKaBzeDI30p6y
6PZitsMRBv7y2dzDILjYogIP0L3FTRyN36OebgVCXPiUAc3VaeeaIdLJ6emnDjt+tVS/dbgx0F+gB/
KooMoY3SJiGPE+hUH8p3pNkYmhzeR3xXi9OEuGAZdK+E+RRA==o13M-----END PGP SIGNATURE-----

==========

Appendix VIII - United States Congress Phone and FAX List

==========

Since PGP is such a political piece of software, I felt that it would be appropriate to include a phone and fax list for the executive and legislative branches of the United States government. If you care at all about the issue of personal privacy, please write to your local representitives and the President expressing your feelings.

==========

US GOVERNMENT ADDRESSES

1 February 1993

The White House
=== ===== =====

President Bill Clinton
1600 Pennsylvania Avenue, NW
Washington, DC 20500

(202) 456-1414 Switchboard
(202) 456-1111 Comment line
(202) 456-2883 FAX 1
(202) 456-2461 FAX 2

75300.3115@compuserve.com EMail
president@whitehouse.gov EMail

First Lady Hillary Rodham Clinton
1600 Pennsylvania Avenue, NW
Washington, DC 20500

(202) 456-6266

Vice President Albert Gore
Old Executive Office Building
Washington, DC 20500

(202) 456-2326

vice-president@whitehouse.gov EMail

The Cabinet
=== =======

Commerce

Ronald H. Brown
Department of Commerce
14th Street and Constitution Avenue, NW
Washington, DC 20230

(202) 482-4901

Defense

Les Aspin
Department of Defense
The Pentagon
Washington, DC 20301

(703) 697-5737

State

Warren Christopher
Department of State
2201 C Street, NW
Washington, DC 20520

(202) 647-6575
(202) 647-7120 FAX

Justice

Janet Reno
Attorney General
Department of Justice
10th Street and Constitution Avenue, NW
Washington, DC 20530

(202) 514-2007
(202) 514-5331 FAX

Treasury

Lloyd Bentsen
Department of the Treasury
1500 Pennsylvania Avenue, NW
Washington, DC 20220

(202) 622-2960
(202) 622-1999 FAX

Federal Information Center
--- --- ---

(800) 726-4995

US Senate, 103rd Congress phone and fax numbers
==========

Information from US Congress Yellow Book, January 1993

p	st	name	phone	fax
R	AK	Murkowski, Frank H.	1-202-224-6665	1-202-224-5301

R AK Stevens, Ted 1-202-224-3004 1-202-224-1044
D AL Heflin, Howell T. 1-202-224-4124 1-202-224-3149
D AL Shelby, Richard C. 1-202-224-5744 1-202-224-3416
D AR Bumpers, Dale 1-202-224-4843 1-202-224-6435
D AR Pryor, David 1-202-224-2353 na
D AZ DeConcini, Dennis 1-202-224-4521 1-202-224-2302
R AZ McCain, John 1-202-224-2235 na
D CA Boxer, Barbara 1-202-225-5161 na
D CA Feinstein, Diane 1-202-224-3841 na
D CO Campbell, Ben N. 1-202-225-4761 1-202-225-0228
R CO Brown, Henry 1-202-224-5941 na
D CT Dodd, Christopher J. 1-202-224-2823 na
D CT Lieberman, Joseph I. 1-202-224-4041 1-202-224-9750
D DE Biden Jr., Joseph R. 1-202-224-5042 na
R DE Roth Jr., William V. 1-202-224-2441 1-202-224-2805
D FL Graham, Robert 1-202-224-3041 na
R FL Mack, Connie 1-202-224-5274 1-202-224-8022
D GA Nunn, Samuel 1-202-224-3521 1-202-224-0072
R GA Coverdell, Paul 1-202-224-3643 na
D HI Akaka, Daniel K. 1-202-224-6361 1-202-224-2126
D HI Inouye, Daniel K. 1-202-224-3934 1-202-224-6747
D IA Harkin, Thomas 1-202-224-3254 1-202-224-7431
R IA Grassley, Charles E. 1-202-224-3744 na
R ID Craig, Larry E. 1-202-224-2752 1-202-224-2573
R ID Kempthorne, Dirk 1-202-224-6142 1-202-224-5893
D IL Moseley-Braun, Carol 1-202-224-2854 na
D IL Simon, Paul 1-202-224-2152 1-202-224-0868
R IN Coats, Daniel R. 1-202-224-5623 1-202-224-8964
R IN Lugar, Richard G. 1-202-224-4814 na
R KS Dole, Robert 1-202-224-6521 1-202-224-8952
R KS Kassebaum, Nancy L. 1-202-224-4774 1-202-224-3514
D KY Ford, Wendell H. 1-202-224-4343 na
R KY McConnell, Mitch 1-202-224-2541 1-202-224-2499
D LA Breaux, John B. 1-202-224-4623 na
D LA Johnston, J. Bennett 1-202-224-5824 na
D MA Kennedy, Edward M. 1-202-224-4543 1-202-224-2417
D MA Kerry, John F. 1-202-224-2742 na
D MD Mikulski, Barbara A. 1-202-224-4654 1-202-224-8858
D MD Sarbanes, Paul S. 1-202-224-4524 1-202-224-1651
D ME Mitchell, George J. 1-202-224-5344 na
R ME Cohen, William S. 1-202-224-2523 1-202-224-2693
D MI Levin, Carl 1-202-224-6221 na
D MI Riegle Jr., Donald 1-202-224-4822 1-202-224-8834
D MN Wellstone, Paul 1-202-224-5641 1-202-224-8438
R MN Durenberger, David 1-202-224-3244 na
R MO Bond, Christopher S. 1-202-224-5721 1-202-224-8149

R MO Danforth, John C. 1-202-224-6154 na
R MS Cochran, Thad 1-202-224-5054 na
R MS Lott, Trent 1-202-224-6253 1-202-224-2262
D MT Baucus, Max 1-202-224-2651 na
R MT Burns, Conrad R. 1-202-224-2644 1-202-224-8594
R NC Faircloth, D. M. 1-202-224-3154 1-202-224-7406
R NC Helms, Jesse 1-202-224-6342 na
D ND Conrad, Kent 1-202-224-2043 na
D ND Dorgan, Byron L. 1-202-225-2611 1-202-225-9436
D NE Exon, J. J. 1-202-224-4224 na
D NE Kerrey, Joseph R. 1-202-224-6551 1-202-224-7645
R NH Gregg, Judd 1-202-224-3324 na
R NH Smith, Robert 1-202-224-2841 1-202-224-1353
D NJ Bradley, William 1-202-224-3224 1-202-224-8567
D NJ Lautenberg, Frank R. 1-202-224-4744 1-202-224-9707
D NM Bingaman, Jeff 1-202-224-5521 na
R NM Domenici, Pete V. 1-202-224-6621 1-202-224-7371
D NV Bryan, Richard H. 1-202-224-6244 na
D NV Reid, Harry 1-202-224-3542 1-202-224-7327
D NY Moynihan, Daniel P. 1-202-224-4451 1-202-224-9293
R NY D'Amato, Alfonse M. 1-202-224-6542 1-202-224-5871
D OH Glenn, John 1-202-224-3353 na
D OH Metzenbaum, Howard 1-202-224-2315 1-202-224-6519
D OK Boren, David L. 1-202-224-4721 na
R OK Nickles, Donald 1-202-224-5754 1-202-224-6008
R OR Hatfield, Mark O. 1-202-224-3753 na
R OR Packwood, Robert 1-202-224-5244 na
D PA Wofford, Harris 1-202-224-6324 1-202-224-4161
R PA Specter, Arlen 1-202-224-4254 na
D RI Pell, Claiborne 1-202-224-4642 1-202-224-4680
R RI Chafee, John H. 1-202-224-2921 na
D SC Hollings, Ernest F. 1-202-224-6121 na
R SC Thurmond, Strom 1-202-224-5972 1-202-224-1300
D SD Daschle, Thomas A. 1-202-224-2321 1-202-224-2047
R SD Pressler, Larry 1-202-224-5842 1-202-224-1630
D TN Mathews, Harlan 1-202-224-1036 1-202-228-3679
D TN Sasser, James 1-202-224-3344 na
D TX Krueger, Robert 1-202-224-5922 na
R TX Gramm, Phil 1-202-224-2934 na
R UT Bennett, Robert 1-202-224-5444 na
R UT Hatch, Orrin G. 1-202-224-5251 1-202-224-6331
D VA Robb, Charles S. 1-202-224-4024 1-202-224-8689
R VA Warner, John W. 1-202-224-2023 1-202-224-6295
D VT Leahy, Patrick J. 1-202-224-4242 na
R VT Jeffords, James M. 1-202-224-5141 na
D WA Murray, Patty 1-202-224-2621 1-202-224-0238

R WA Gorton, Slade 1-202-224-3441 1-202-224-9393
D WI Feingold, Russell 1-202-224-5323 na
D WI Kohl, Herbert H. 1-202-224-5653 na
D WV Byrd, Robert C. 1-202-224-3954 1-202-224-4025
D WV Rockefeller, John D. 1-202-224-6472 1-202-224-1689
R WY Simpson, Alan K. 1-202-224-3424 1-202-224-1315
R WY Wallop, Malcolm 1-202-224-6441 1-202-224-3230

103rd Congress phone and fax numbers
==========

The following information is from the US Congress "Yellow Book," Jan. 1993. Four seats were vacant at that time, in CA, MS, OH, and WI. The list below of 436 people includes 5 non-voting members, from Guam (GU), Puerto Rico (PR), Samoa (SA), Virgin Islands (VI), and DC. (some of those abbreviations may be wrong)

p	st	representative	phone	fax
R	AK	Young, Donald	1-202-225-5765	1-202-225-5765
D	AL	Bevill, Thomas	1-202-225-4876	1-202-225-0842
D	AL	Browder, Glen	1-202-225-3261	1-202-225-9020
D	AL	Cramer Jr, Robert E.	1-202-225-4801	na
D	AL	Hilliard, Earl F.	1-202-225-2665	na
R	AL	Bachus, Spencer	1-202-225-4921	na
R	AL	Callahan, H. L.	1-202-225-4931	1-202-225-0562
R	AL	Everett, Terry	1-202-225-2901	na
D	AR	Lambert, Blanche	1-202-225-4076	na
D	AR	Thornton, Raymond	1-202-225-2506	1-202-225-9273
R	AR	Dickey, Jay	1-202-225-3772	1-202-225-8646
R	AR	Hutchinson, Tim	1-202-225-4301	na
D	AZ	Coppersmith, Sam	1-202-225-2635	1-202-225-2607
D	AZ	English, Karan	1-202-225-2190	1-202-225-8819
D	AZ	Pastor, Ed	1-202-225-4065	1-202-225-1655
R	AZ	Kolbe, James T.	1-202-225-2542	1-202-225-0378
R	AZ	Kyl, Jon L.	1-202-225-3361	na
R	AZ	Stump, Robert	1-202-225-4576	1-202-225-6328
D	CA	Becerra, Xavier	1-202-225-6235	1-202-225-2202
D	CA	Beilenson, Anthony	1-202-225-5911	na
D	CA	Berman, Howard L.	1-202-225-4695	na
D	CA	Brown Jr., George E.	1-202-225-6161	1-202-225-8671
D	CA	Condit, Gary	1-202-225-6131	1-202-225-0819
D	CA	Dellums, Ronald V.	1-202-225-2661	1-202-225-9817
D	CA	Dixon, Julian C.	1-202-225-7084	1-202-225-4091
D	CA	Dooley, Calvin M.	1-202-225-3341	1-202-225-9308

D CA Edwards, Donald 1-202-225-3072 1-202-225-9460
D CA Eshoo, Anna G. 1-202-225-8104 na
D CA Fazio, Vic 1-202-225-5716 1-202-225-0354
D CA Filner, Bob 1-202-225-8045 na
D CA Hamburg, Dan 1-202-225-3311 na
D CA Harman, Jane 1-202-225-8220 na
D CA Lantos, Thomas 1-202-225-3531 na
D CA Lehman, Richard H. 1-202-225-4540 na
D CA Martinez, Matthew G. 1-202-225-5464 1-202-225-4467
D CA Matsui, Robert T. 1-202-225-7163 1-202-225-0566
D CA McCandless, Alfred 1-202-225-5330 1-202-226-1040
D CA Miller, George 1-202-225-2095 1-202-225-5609
D CA Mineta, Norman Y. 1-202-225-2631 na
D CA Pelosi, Nancy 1-202-225-4965 1-202-225-8259
D CA Roybal-Allard, Lucille 1-202-225-1766 1-202-226-0350
D CA Schenk, Lynn 1-202-225-2040 1-202-225-2042
D CA Stark, Fortney H. 1-202-225-5065 na
D CA Torres, Esteban E. 1-202-225-5256 na
D CA Tucker III, Walter R. 1-202-225-7924 1-202-225-7926
D CA Waters, Maxine 1-202-225-2201 na
D CA Waxman, Henry A. 1-202-225-3976 1-202-225-4099
D CA Woolsey, Lynn 1-202-225-5161 na
R CA Baker, Bill 1-202-225-1880 1-202-225-2150
R CA Calvert, Ken 1-202-225-1986 na
R CA Cox, Christopher 1-202-225-5611 1-202-225-9177
R CA Cunningham, Randy 1-202-225-5452 1-202-225-2558
R CA Doolittle, John T. 1-202-225-2511 1-202-225-5444
R CA Dornan, Robert K. 1-202-225-2965 1-202-225-3694
R CA Dreier, David 1-202-225-2305 1-202-225-4745
R CA Gallegly, Elton 1-202-225-5811 na
R CA Herger, Walter W. 1-202-225-3076 1-202-225-1609
R CA Horn, Steve 1-202-225-6676 na
R CA Huffington, Michael 1-202-225-3601 na
R CA Hunter, Duncan L. 1-202-225-5672 1-202-225-0235
R CA Kim, Jay C. 1-202-225-3201 1-202-226-1485
R CA Lewis, Jerry 1-202-225-5861 1-202-225-6498
R CA McKeon, Howard P. 1-202-225-1956 1-202-226-0683
R CA Moorhead, Carlos J. 1-202-225-4176 1-202-226-1279
R CA Packard, Ronald 1-202-225-3906 1-202-225-0134
R CA Pombo, Richard 1-202-225-1947 1-202-226-0861
R CA Rohrabacher, Dana 1-202-225-2415 1-202-225-7067
R CA Royce, Ed 1-202-225-4111 na
R CA Thomas, Bill 1-202-225-2915 na
D CO Schroeder, Patricia 1-202-225-4431 1-202-225-5842
D CO Skaggs, David E. 1-202-225-2161 na
R CO Allard, Wayne 1-202-225-4676 1-202-225-8630

R CO Hefley, Joel 1-202-225-4422 1-202-225-1942
R CO McInnis, Scott 1-202-225-4761 1-202-226-0622
R CO Schaefer, Daniel 1-202-225-7882 1-202-225-7885
D CT DeLauro, Rosa 1-202-225-3661 1-202-225-4890
D CT Gejdenson, Samuel 1-202-225-2076 1-202-225-4977
D CT Kennelly, Barbara B. 1-202-225-2265 1-202-225-1031
R CT Franks, Gary 1-202-225-3822 1-202-225-5085
R CT Johnson, Nancy L. 1-202-225-4476 1-202-225-4488
R CT Shays, Christopher 1-202-225-5541 1-202-225-9629
D DC Norton, Eleanor Holmes 1-202-225-8050 1-202-225-3002
R DE Castle, Michael N. 1-202-225-4165 1-202-225-2291
D FL Bacchus, James 1-202-225-3671 1-202-225-9039
D FL Brown, Corrine 1-202-225-0123 1-202-225-2256
D FL Deutsch, Peter 1-202-225-7931 1-202-225-8456
D FL Gibbons, Samuel M. 1-202-225-3376 na
D FL Hastings, Alcee L. 1-202-225-1313 1-202-225-0690
D FL Hutto, Earl 1-202-225-4136 1-202-225-5785
D FL Johnston II, Harry 1-202-225-3001 1-202-225-8791
D FL Meek, Carrie 1-202-225-4506 1-202-226-0777
D FL Peterson, Peter 1-202-225-5235 1-202-225-1586
R FL Bilirakis, Michael 1-202-225-5755 1-202-225-4085
R FL Canady, Charles T. 1-202-225-1252 na
R FL Diaz-Balart, Lincoln 1-202-225-4211 1-202-225-8576
R FL Fowler, Tillie 1-202-225-2501 na
R FL Goss, Porter J. 1-202-225-2536 1-202-225-6820
R FL Lewis, Thomas 1-202-225-5792 1-202-225-1860
R FL McCollum, William 1-202-225-2176 na
R FL Mica, John L. 1-202-225-4035 1-202-226-0821
R FL Miller, Dan 1-202-225-5015 1-202-226-0828
R FL Ros-Lehtinen, Ileana 1-202-225-3931 1-202-225-5620
R FL Shaw Jr., E. C. 1-202-225-3026 1-202-225-8398
R FL Stearns, Clifford B. 1-202-225-5744 1-202-225-3973
R FL Thurman, Carol L. 1-202-225-1002 1-202-226-0329
R FL Young, C. W. 1-202-225-5961 1-202-225-9764
D GA Bishop, Sanford 1-202-225-3631 1-202-225-2203
D GA Darden III, George 1-202-225-2931 na
D GA Deal, Nathan 1-202-225-5211 1-202-225-8272
D GA Johnson, Don 1-202-225-4101 1-202-226-1466
D GA Lewis, John 1-202-225-3801 1-202-225-0351
D GA McKinney, Cynthia 1-202-225-1605 1-202-226-0691
D GA Rowland, J. R. 1-202-225-6531 na
R GA Collins, Mac 1-202-225-5901 1-202-225-2515
R GA Gingrich, Newt 1-202-225-4501 1-202-225-4656
R GA Kingston, Jack 1-202-225-5831 1-202-226-2269
R GA Linder, John 1-202-225-4272 na
D GU Underwood, Robert A. 1-202-225-1188 1-202-226-0341

D HI Abercrombie, Neil 1-202-225-2726 na
D HI Mink, Patsy T. 1-202-225-4906 1-202-225-4987
D IA Smith, Neal 1-202-225-4426 na
R IA Grandy, Fred 1-202-225-5476 na
R IA Leach, James 1-202-225-6576 1-202-226-1278
R IA Lightfoot, James R. 1-202-225-3806 1-202-225-6973
R IA Nussle, James Allen 1-202-225-2911 1-202-225-9129
D ID LaRocco, Larry 1-202-225-6611 na
R ID Crapo, Michael D. 1-202-225-5531 na
D IL Collins, Cardiss 1-202-225-5006 1-202-225-8396
D IL Costello, Jerry F. 1-202-225-5661 1-202-225-0285
D IL Durbin, Richard J. 1-202-225-5271 1-202-225-0170
D IL Evans, Lane 1-202-225-5905 1-202-225-5396
D IL Lipinski, William O. 1-202-225-5701 1-202-225-1012
D IL Poshard, Glendal W. 1-202-225-5201 1-202-225-1541
D IL Reynolds, Mel 1-202-225-0773 na
D IL Rostenkowski, Daniel 1-202-225-4061 na
D IL Rush, Bobby L. 1-202-225-4372 1-202-226-0333
D IL Sangmeister, George 1-202-225-3635 1-202-225-4447
D IL Yates, Sidney R. 1-202-225-2111 1-202-225-3493
R IL Crane, Philip M. 1-202-225-3711 na
R IL Ewing, Thomas 1-202-225-2371 1-202-225-8071
R IL Fawell, Harris W. 1-202-225-3515 1-202-225-9420
R IL Gutierrez, Luis V. 1-202-225-8203 1-202-225-7810
R IL Hastert, J. D. 1-202-225-2976 1-202-225-0697
R IL Hyde, Henry J. 1-202-225-4561 1-202-226-1240
R IL Manzullo, Donald 1-202-225-5676 1-202-225-5284
R IL Michel, Robert H. 1-202-225-6201 1-202-225-9461
R IL Porter, John E. 1-202-225-4835 1-202-225-0157
D IN Buyer, Steve 1-202-225-5037 na
D IN Hamilton, Lee H. 1-202-225-5315 1-202-225-1101
D IN Jacobs Jr., Andrew 1-202-225-4011 na
D IN Long, Jill 1-202-225-4436 na
D IN McCloskey, Frank 1-202-225-4636 1-202-225-4688
D IN Roemer, Timothy 1-202-225-3915 1-202-225-6798
D IN Sharp, Philip R. 1-202-225-3021 na
D IN Visclosky, Peter J. 1-202-225-2461 1-202-225-2493
R IN Burton, Daniel 1-202-225-2276 1-202-225-0016
R IN Myers, John T. 1-202-225-5805 na
D KS Glickman, Daniel 1-202-225-6216 na
D KS Slattery, James 1-202-225-6601 1-202-225-1445
R KS Meyers, Jan 1-202-225-2865 1-202-225-0554
R KS Roberts, Pat 1-202-225-2715 1-202-225-5375
D KY Baesler, Scotty 1-202-225-4706 na
D KY Barlow, Tom 1-202-225-3115 1-202-225-2169
D KY Mazzoli, Romano L. 1-202-225-5401 na

D KY Natcher, William H. 1-202-225-3501 na
R KY Bunning, James 1-202-225-3465 1-202-225-0003
R KY Rogers, Harold 1-202-225-4601 1-202-225-0940
D LA Fields, Cleo 1-202-225-8490 1-202-225-8959
D LA Hayes, James A. 1-202-225-2031 1-202-225-1175
D LA Jefferson, William 1-202-225-6636 1-202-225-1988
D LA Tauzin, W. J. 1-202-225-4031 1-202-225-0563
R LA Baker, Richard H. 1-202-225-3901 1-202-225-7313
R LA Livingston, Robert 1-202-225-3015 1-202-225-0739
R LA McCrery, James 1-202-225-2777 1-202-225-8039
D MA Frank, Barney 1-202-225-5931 1-202-225-0182
D MA Kennedy II, Joseph P. 1-202-225-5111 1-202-225-9322
D MA Markey, Edward J. 1-202-225-2836 1-202-225-8689
D MA Meehan, Martin T. 1-202-225-3411 1-202-226-0771
D MA Moakley, John Joseph 1-202-225-8273 1-202-225-7304
D MA Neal, Richard E. 1-202-225-5601 1-202-225-8112
D MA Olver, John W. 1-202-225-5335 1-202-226-1224
D MA Studds, Gerry E. 1-202-225-3111 1-202-225-2212
R MA Blute, Peter I. 1-202-225-6101 1-202-225-2217
R MA Torkildsen, Peter G. 1-202-225-8020 1-202-225-8037
D MD Cardin, Benjamin L. 1-202-225-4016 na
D MD Hoyer, Steny H. 1-202-225-4131 1-202-225-4300
D MD Mfume, Kweisi 1-202-225-4741 1-202-225-3178
D MD Wynn, Albert R. 1-202-225-8699 1-202-225-8714
R MD Bartlett, Roscoe G. 1-202-225-2721 na
R MD Bentley, Helen D. 1-202-225-3061 1-202-225-4251
R MD Gilchrest, Wayne T. 1-202-225-5311 1-202-225-0254
R MD Morella, Constance 1-202-225-5341 1-202-225-1389
D ME Andrews, Thomas H. 1-202-225-6116 1-202-225-9065
R ME Snowe, Olympia J. 1-202-225-6306 na
D MI Barcia, James A. 1-202-225-8171 1-202-225-2168
D MI Bonior, David E. 1-202-225-2106 1-202-226-1169
D MI Carr, Robert 1-202-225-4872 1-202-225-1260
D MI Collins Jr., Barbara 1-202-225-2261 1-202-225-6645
D MI Conyers Jr., John 1-202-225-5126 1-202-225-0072
D MI Dingell, John D. 1-202-225-4071 1-202-225-7426
D MI Ford, William D. 1-202-225-6261 na
D MI Kildee, Dale E. 1-202-225-3611 na
D MI Levin, Sander M. 1-202-225-4961 1-202-226-1033
D MI Stupak, Bart 1-202-225-4735 1-202-225-4744
R MI Camp, David Lee 1-202-225-3561 1-202-225-9679
R MI Henry, Paul B. 1-202-225-3831 na
R MI Hoekstra, Peter 1-202-225-4401 na
R MI Knollenberg, Joe 1-202-225-5802 1-202-226-2356
R MI Smith, Nick 1-202-225-6276 na
R MI Upton, Frederick S. 1-202-225-3761 1-202-225-4986

D MN Minge, David 1-202-225-2331 na
D MN Oberstar, James L. 1-202-225-6211 1-202-225-0699
D MN Penny, Timothy J. 1-202-225-2472 1-202-225-0051
D MN Peterson, Collin C. 1-202-225-2165 1-202-225-1593
D MN Sabo, Martin O. 1-202-225-4755 na
D MN Vento, Bruce F. 1-202-225-6631 na
R MN Grams, Rod 1-202-225-2271 1-202-225-9802
R MN Ramstad, James M. 1-202-225-2871 1-202-225-6351
D MO Clay, William L. 1-202-225-2406 1-202-225-1725
D MO Danner, Pat 1-202-225-7041 na
D MO Gephardt, Richard A. 1-202-225-2671 1-202-225-7452
D MO Skelton, Ike 1-202-225-2876 1-202-225-2695
D MO Volkmer, Harold L. 1-202-225-2956 1-202-225-7834
D MO Wheat, Alan 1-202-225-4535 1-202-225-5990
R MO Emerson, Bill 1-202-225-4404 1-202-225-9621
R MO Hancock, Melton D. 1-202-225-6536 1-202-225-7700
R MO Talent, James M. 1-202-225-2561 1-202-225-2563
D MS Montgomery, G. V. 1-202-225-5031 1-202-225-3375
D MS Parker, Paul M. 1-202-225-5865 1-202-225-5886
D MS Taylor, Gene 1-202-225-5772 1-202-225-7074
D MS Whitten, Jamie L. 1-202-225-4306 1-202-225-4328
D MT Williams, Pat 1-202-225-3211 na
D NC Clayton, Eva 1-202-225-3101 na
D NC Hefner, W. G. 1-202-225-3715 1-202-225-4036
D NC Lancaster, H. M. 1-202-225-3415 1-202-225-0666
D NC Neal, Stephen L. 1-202-225-2071 1-202-225-4060
D NC Price, David E. 1-202-225-1784 1-202-225-6314
D NC Rose, Charles 1-202-225-2731 1-202-225-2470
D NC Valentine, Tim 1-202-225-4531 1-202-225-1539
D NC Watt, Melvin 1-202-225-1510 1-202-225-1512
R NC Ballenger, Thomas C. 1-202-225-2576 1-202-225-0316
R NC Coble, Howard 1-202-225-3065 1-202-225-8611
R NC McMillan, J. A. 1-202-225-1976 na
R NC Taylor, Charles Hart 1-202-225-6401 1-202-251-0794
D ND Pomeroy, Earl 1-202-225-2611 1-202-226-0893
D NE Hoagland, Peter 1-202-225-4155 na
R NE Barrett, William E. 1-202-225-6435 na
R NE Bereuter, Douglas 1-202-225-4806 1-202-226-1148
D NH Swett, Richard N. 1-202-225-5206 na
R NH Zeliff Jr., William 1-202-225-5456 1-202-225-4370
D NJ Andrews, Robert E. 1-202-225-6501 na
D NJ Hughes, William J. 1-202-225-6572 1-202-226-1108
D NJ Klein, Herbert C. 1-202-225-5751 na
D NJ Menendez, Robert 1-202-225-7919 1-202-226-0792
D NJ Pallone Jr., Frank 1-202-225-4671 1-202-225-9665
D NJ Payne, Donald M. 1-202-225-3436 1-202-225-4160

D NJ Torricelli, Robert 1-202-224-5061 1-202-225-0843
R NJ Franks, Bob 1-202-225-5361 1-202-225-9460
R NJ Gallo, Dean A. 1-202-225-5034 1-202-225-0658
R NJ Roukema, Marge 1-202-225-4465 1-202-225-9048
R NJ Saxton, H. J. 1-202-225-4765 1-202-225-0778
R NJ Smith, Christopher 1-202-225-3765 1-202-225-7768
R NJ Zimmer, Richard A. 1-202-225-5801 1-202-225-9181
D NM Richardson, William 1-202-225-6190 na
R NM Schiff, Steven H. 1-202-225-6316 1-202-225-4975
R NM Skeen, Joseph 1-202-225-2365 1-202-225-9599
D NV Bilbray, James H. 1-202-225-5965 1-202-225-8808
R NV Vucanovich, Barbara 1-202-225-6155 1-202-225-2319
D NY Ackerman, Gary L. 1-202-225-2601 na
D NY Engel, Eliot L. 1-202-225-2464 na
D NY Flake, Floyd H. 1-202-225-3461 1-202-226-4169
D NY Hinchey, Maurice D. 1-202-225-6335 na
D NY Hochbrueckner, G. 1-202-225-3826 1-202-225-0776
D NY LaFalce, John J. 1-202-225-3231 na
D NY Lowey, Nita M. 1-202-225-6506 1-202-225-0546
D NY Maloney, Carolyn B. 1-202-225-7944 na
D NY Manton, Thomas J. 1-202-225-3965 na
D NY McNulty, Michael R. 1-202-225-5076 1-202-225-5077
D NY Nadler, Jerrold 1-202-225-5635 1-202-225-6923
D NY Owens, Major R. 1-202-225-6231 1-202-226-0112
D NY Rangel, Charles B. 1-202-225-4365 1-202-225-0816
D NY Schumer, Charles E. 1-202-225-6616 1-202-225-4183
D NY Serrano, Jose E. 1-202-225-4361 1-202-225-6001
D NY Slaughter, Louise M. 1-202-225-3615 1-202-225-7822
D NY Towns, Edolphus 1-202-225-5936 1-202-225-1018
D NY Velazquez, Nydia M. 1-202-225-2361 1-202-226-0327
R NY Boehlert, Sherwood 1-202-225-3665 1-202-225-1891
R NY Fish Jr., Hamilton 1-202-225-5441 1-202-225-0962
R NY Gilman, Benjamin A. 1-202-225-3776 na
R NY Houghton, Amory 1-202-225-3161 1-202-225-5574
R NY King, Peter T. 1-202-225-7896 1-202-226-2279
R NY Lazio, Rick A. 1-202-225-3335 na
R NY Levy, David A. 1-202-225-5516 1-202-225-4672
R NY McHugh, John M. 1-202-225-4611 na
R NY Molinari, Susan 1-202-225-3371 1-202-226-1272
R NY Paxon, L. W. 1-202-225-5265 1-202-225-5910
R NY Quinn, Jack 1-202-225-3306 1-202-226-0347
R NY Solomon, Gerald B. 1-202-225-5614 1-202-225-1168
R NY Walsh, James T. 1-202-225-3701 1-202-225-4042
D OH Applegate, Douglas 1-202-225-6265 na
D OH Brown, Sherrod 1-202-225-3401 na
D OH Fingerhut, Eric D. 1-202-225-5731 na

D OH Hall, Tony P. 1-202-225-6465 na
D OH Kaptur, Marcy 1-202-225-4146 1-202-225-7711
D OH Mann, Davis S. 1-202-225-2216 na
D OH Sawyer, Thomas C. 1-202-225-5231 1-202-225-5278
D OH Stokes, Louis 1-202-225-7032 1-202-225-1339
D OH Strickland, Ted 1-202-225-5705 1-202-226-0331
D OH Traficant Jr., James 1-202-225-5261 1-202-225-3719
R OH Boehner, John Andrew 1-202-225-6205 1-202-225-0704
R OH Gillmor, Paul E. 1-202-225-6405 na
R OH Hobson, David L. 1-202-225-4324 na
R OH Hoke, Martin R. 1-202-225-5871 1-202-226-0994
R OH Kasich, John R. 1-202-225-5355 na
R OH Oxley, Michael G. 1-202-225-2676 na
R OH Pryce, Deborah 1-202-225-2015 1-202-226-0986
R OH Regula, Ralph 1-202-225-3876 1-202-225-3059
D OK Brewster, Billy Kent 1-202-225-4565 na
D OK English, Glenn 1-202-225-5565 1-202-225-8698
D OK McCurdy, David 1-202-225-6165 1-202-225-9746
D OK Synar, Michael 1-202-225-2701 1-202-225-2796
R OK Inhofe, James M. 1-202-225-2211 1-202-225-9187
R OK Istook, Ernest Jim 1-202-225-2132 na
D OR DeFazio, Peter A. 1-202-225-6416 na
D OR Furse, Elizabeth 1-202-225-0855 na
D OR Kopetski, Michael J. 1-202-225-5711 1-202-225-9477
D OR Wyden, Ronald 1-202-225-4811 na
R OR Smith, Robert F. 1-202-225-6730 na
D PA Blackwell, Lucien E. 1-202-225-4001 1-202-225-7362
D PA Borski, Robert A. 1-202-225-8251 1-202-225-4628
D PA Coyne, William J. 1-202-225-2301 na
D PA Foglietta, Thomas M. 1-202-225-4731 1-202-225-0088
D PA Holden, Tim 1-202-225-5546 1-202-226-0996
D PA Kanjorski, Paul E. 1-202-225-6511 1-202-225-9024
D PA Klink, Ron 1-202-225-2565 na
D PA Margolies-Mezvinsky, Marjorie 1-202-225-6111 1-202-226-0798
D PA McHale, Paul 1-202-225-6411 1-202-225-5320
D PA Murphy, Austin J. 1-202-225-4665 1-202-225-4772
D PA Murtha, John P. 1-202-225-2065 1-202-225-5709
R PA Clinger Jr., William 1-202-225-5121 1-202-225-4681
R PA Gekas, George W. 1-202-225-4315 1-202-225-8440
R PA Goodling, William F. 1-202-225-5836 1-202-226-1000
R PA Greenwood, Jim 1-202-225-4276 1-202-225-9511
R PA McDade, Joseph M. 1-202-225-3731 1-202-225-9594
R PA Ridge, Thomas J. 1-202-225-5406 na
R PA Santorum, Richard J. 1-202-225-2135 1-202-225-7747
R PA Shuster, Bud 1-202-225-2431 na
R PA Walker, Robert S. 1-202-225-2411 na

R PA Weldon, Curt 1-202-225-2011 1-202-225-8137
D PR Romero-Barcelo, Carlos 1-202-225-2615 1-202-225-2154
D RI Reed, John F. 1-202-225-2735 1-202-225-9580
R RI Machtley, Ronald K. 1-202-225-4911 1-202-225-4417
D SA Faleomavaega, Eni F.H. 1-202-225-8577 na
D SC Clyburn, James E. 1-202-225-3315 1-202-225-2302
D SC Derrick, Butler 1-202-225-5301 na
D SC Spratt Jr., John M. 1-202-225-5501 1-202-225-0464
R SC Inglis, Bob 1-202-225-6030 na
R SC Ravenel Jr., Arthur 1-202-225-3176 na
R SC Spence, Floyd 1-202-225-2452 1-202-225-2455
D SD Johnson, Timothy P. 1-202-225-2801 1-202-225-2427
D TN Clement, Robert 1-202-225-4311 1-202-226-1035
D TN Cooper, James 1-202-225-6831 1-202-225-4520
D TN Ford, Harold E. 1-202-225-3265 na
D TN Lloyd, Marilyn 1-202-225-3271 1-202-225-6974
D TN Tanner, John S. 1-202-225-4714 1-202-225-1765
R TN Duncan Jr., John J. 1-202-225-5435 1-202-225-6440
R TN Gordon, Bart 1-202-225-4231 1-202-225-6887
R TN Quillen, James H. 1-202-225-6356 1-202-225-7812
R TN Sundquist, Donald 1-202-225-2811 1-202-225-2814
D TX Andrews, Michael A. 1-202-255-7508 na
D TX Brooks, Jack 1-202-225-6565 1-202-225-1584
D TX Bryant, John 1-202-225-2231 na
D TX Chapman, Jim 1-202-225-3035 1-202-225-7265
D TX Coleman, Ronald D. 1-202-225-4831 na
D TX Edwards, Chet 1-202-225-6105 1-202-225-0350
D TX Frost, Martin 1-202-225-3605 1-202-225-4951
D TX Geren, Peter 1-202-225-5071 1-202-225-2786
D TX Gonzalez, Henry B. 1-202-225-3236 1-202-225-1915
D TX Green, Gene 1-202-225-1688 1-202-225-9903
D TX Hall, Ralph M. 1-202-225-6673 1-202-225-3332
D TX Johnson, Eddie Bernice 1-202-225-8885 na
D TX Laughlin, Gregory H. 1-202-225-2831 1-202-225-1108
D TX Ortiz, Solomon P. 1-202-225-7742 1-202-226-1134
D TX Pickle, J. J. 1-202-225-4865 na
D TX Sarpalius, Bill 1-202-225-3706 1-202-225-6142
D TX Stenholm, Charles W. 1-202-225-6605 1-202-225-2234
D TX Tejeda, Frank 1-202-225-1640 na
D TX Washington, Craig A. 1-202-225-3816 na
D TX Wilson, Charles 1-202-225-2401 1-202-225-1764
D TX de la Garza, E 1-202-225-2531 1-202-225-2534
R TX Archer, William 1-202-225-2571 1-202-225-4381
R TX Armey, Richard K. 1-202-225-7772 1-202-225-7614
R TX Barton, Joseph 1-202-225-2002 1-202-225-3052
R TX Bonilla, Henry 1-202-225-4511 na

R TX Combest, Larry 1-202-225-4005 na
R TX DeLay, Thomas 1-202-225-5951 na
R TX Fields, Jack 1-202-225-4901 na
R TX Johnson, Sam 1-202-225-4201 na
R TX Smith, Lamar S. 1-202-225-4236 1-202-225-8628
D UT Orton, William H. 1-202-225-7751 1-202-226-1223
D UT Shepherd, Karen 1-202-225-3011 1-202-226-0354
R UT Hansen, James V. 1-202-225-0453 1-202-225-5857
D VA Boucher, Rick 1-202-225-3861 na
D VA Byrne, Leslie L. 1-202-225-1492 na
D VA Moran Jr., James P. 1-202-225-4376 1-202-225-0017
D VA Payne Jr., Lewis F. 1-202-225-4711 1-202-226-1147
D VA Pickett, Owen B. 1-202-225-4215 1-202-225-4218
D VA Scott, Robert C. 1-202-225-8351 1-202-225-3854
D VA Sisisky, Norman 1-202-225-6365 1-202-226-1170
R VA Bateman, Herbert H. 1-202-225-4261 1-202-225-4382
R VA Bliley Jr., Thomas J. 1-202-225-2815 na
R VA Goodlatte, Robert W. 1-202-225-5431 1-202-225-9681
R VA Wolf, Frank R. 1-202-225-5136 na
D VI de Lugo, Ron 1-202-225-1790 1-202-225-9392
I VT Sanders, Bernard 1-202-225-4115 1-202-225-6790
D WA Cantwell, Maria 1-202-225-6311 1-202-225-2286
D WA Dicks, Norman D. 1-202-225-5916 na
D WA Foley, Thomas S. 1-202-225-2006 na
D WA Inslee, Jay 1-202-225-5816 1-202-226-1137
D WA Kreidler, Mike 1-202-225-8901 1-202-226-2361
D WA McDermott, James A. 1-202-225-3106 1-202-225-9212
D WA Swift, Al 1-202-225-2605 1-202-225-2608
D WA Unsoeld, Jolene 1-202-225-3536 1-202-225-9095
R WA Dunn, Jennifer 1-202-225-7761 na
D WI Barrett, Thomas M. 1-202-225-3571 na
D WI Gunderson, Steve 1-202-225-5506 1-202-225-6195
D WI Kleczka, Gerald D. 1-202-225-4572 na
D WI Obey, David R. 1-202-225-3365 na
R WI Klug, Scott 1-202-225-2906 na
R WI Petri, Thomas E. 1-202-225-2476 1-202-225-2356
R WI Roth, Toby 1-202-225-5665 1-202-225-0087
R WI Sensenbrenner, F. J. 1-202-225-5101 1-202-225-3190
D WV Mollohan, Alan B. 1-202-225-4172 1-202-225-7564
D WV Rahall II, Nick Joe 1-202-225-3452 1-202-225-9061
D WV Wise Jr., Robert E. 1-202-225-2711 1-202-225-7856
R WY Thomas, Craig 1-202-225-2311 1-202-225-0726