Anonymous Communications and the Macintosh

Marshall Clow <mclow@mailhost2.csusm.edu>

With the explosive rise in popularity of the internet, communications via email have taken off. However, the lack of privacy of email has raised concerns with people that their correspondence is, in fact, not private. Encryption programs, such as PGP, offer a solution to this problem. Some people, however, remain concerned about the collection and analysis of their email, even if the contents are unreadable. They feel that it is no one's business who they are communicating with, let alone what they are saying. In this paper, I will explore various methods of electronic communications that do not reveal the sender's or the receiver's identity, with an emphasis on implementations for the Macintosh.

Electronic mail, privacy and anonymity

Electronic mail is here to stay. Fast, cheap, and convenient, it is used every day by millions of people. However, it is neither private nor anonymous. Any system administrator can search all the mail passing through his machine with minimal effort, or simply record headers to see who is sending mail to whom.

How can we bring privacy and anonymity to electronic mail? The first problem, privacy, has a simple solution. Email can be made private by encrypting the contents of the mail using a program like PGP (See <http://www.pgp.com>) This has been compared to placing the letter in an envelope so that the postal carriers cannot read it. Many of the solutions in this paper will use encryption to hide the contents of the letter.

Anonymity is rather more difficult.

The United States has a long history of anonymous communications and discourse. A check of the shelves of a library or bookstore will turn up several books written by ‘Anonymous’. There is no requirement that a “user of the US Mail” identify themselves in any way. Anyone can drop a letter into a mailbox, and, if it contains sufficient postage, it will be delivered.

‘Publius’ and ‘Deep Throat’ are two anonymous figures who have had significant effects on the government of the United States. The Supreme Court recently re-affirmed the right of people to speak anonymously (the case involved a state law that required all political materials to have the name and address of the person responsible)

Solution #0: anon.penet.fi

In 1993, a man named Julf Helsingius set up a machine at anon.penet.fi and told people it was an anonymous remailer. The first time that you sent mail through the machine it would make up a user name for you (an01234@anon.penet.fi, for example). It would then rewrite the headers of the mail message so that they message appeared to come from that address. It also kept a database mapping the anon address to your real address, so that when someone replied to an01234@anon.penet.fi, it would get forwarded to you. It was simple, it worked well. Thousands of people used it on a daily basis.

It had one problem. Julf's machine contained a database mapping all of the anonymous addresses to real addresses. If Julf was unscrupulous, he could "out" any or all of his users. This list was a target for anyone who had been wronged via anonymous email. Hackers would try to break into Julf's system and retrieve this list. In 1995, Julf was served with papers from the police in Finland (being prodded by a complaint from the Church of Scientology) that he turn over the list of users. He declined. After a month, or so, he was forced to reveal the email address of two of his users, and in 1996 decided to take down his mail server and to destroy the list of users.

For more details about anon.penet.fi, see <http://www.penet.fi/> and <http://duplox.wz-berlin.de/people/s/aneng.html> and <http://www.well.com/user/abacard/remail.html>

Solution #1: Anonymous remailers.

Imagine that you took a letter, put it in an envelope, and addressed it to a friend (call him Zach). Then you took that letter, and put it in another envelope, addressed to someone else, along with a note asking him to drop the inner envelope into a mailbox. Obviously, you could then put that envelope into another one, and repeat the process as many times as you liked.

That's how anonymous remailers work. The envelopes are layers of encryption, and the people that you mail them to are computers connected to the internet, but those are just details. The process is exactly the same. Each remailer, when it receives a message, removes the envelope (decrypts the message), which reveals some instructions and a new (encrypted) message. The instructions are usually of the form "Remail this to foo@bar.com" which may or may not be another remailer.

Let's think about the implications for this. No one except the first remailer knows who the message came from (all the other ones see that it came from another remailer). No one except the last remailer knows where the message is going (all the other ones send it to another remailer). None of the remailers (or anyone watching) can read the message. No one remailer can piece together all the information needed to identify both the sender and the receiver. In fact, it takes all the remailers that handle a message, working in collusion, to identify a message as it passes through the remailer network. Every single remailer that handled a message has to cooperate in order to trace a message. If even one remailer doesn't log its' traffic, then no one can trace it.

The syntax for using anonymous remailers is simple. You create an encrypted message (to Zach), and prepend remailer commands to it. Remailer commands start with a line containing only "::", and end with a blank line. Here's a simple example:

```
::
Request-Remailing-To: mclow@mailhost2.csusm.edu

-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.0
MessageID: CZJ+MAkjvH7NUHmClZ9uYPE2j+SNW4YZ

hQCMA6nepJc42RbVAQQArmuzkG5e5stnDy8T67L2vF/FdLks+j0L2RNalwOOsKri
Ypp9NuGQIHafwgFZjFxgYCZijUxdmCbyrH7erPLzHOFI/vyNiYb0UA2WS/CqHxsW
iH2GrtJTRUAcMB55bNlajjb7iHuZQKdTqxl5ZXsfKuXn1of0FIpGoWnJAdlFks2k
SwCYFEicv4B7Ej4/wcYhKIN2muUy9Ih4Bcs1AblILAUn6/XCwanNqhjzc0dPCUxi
nRXpbimiYwihhZxM6gF52qBhxYCNP1iLhNpakQ==
=ven4
-----END PGP MESSAGE-----
```

If you want to send your message through more than one remailer (this is known as chaining remailers), you repeat the process, using the remailer's PGP key for the next encryption.

I have written a program to automate this process. It is available at <http://www.idio.com/anon/remailer.html>.

The program is very simple to use. You choose a routing for the message (or have the program choose one at random), give the destination for the message, and the encrypted message is created. You can then paste the message into your email program and mail it to the first remailer in your chain.

There exist web-based interfaces to the remailer network. However, these are not as secure as doing your own processing and mailing, since the owner of the web site can snoop your message and the path through the remailer network that you are using.

Web-based remailer interfaces are availiable at: <http://www.anonymizer.com/email/remailer-simple.cgi> and <https://www.replay.com/remailer/anon.html>

Replying to Anonymous email.

Anonymous remailers are great for sending mail, but if your mail comes into Zach's mailbox as from "remailer@anonymous.com", how can he answer? If Zach knows who you are (your true name), then he can reply using the remailer network. However, there are several common cases where Zach doesn't know your name (Whistle blowers being just one case) How can he possibly reply to a person whose name and email address he doesn't know?

Solution #1: Usenet News

If you don't know where Zach is, you can't send a message to him. Instead, you might choose to send a message everywhere, and hope that he sees it. Fortunately, the Internet provides a mechanism for copying a message around the world. It's called Usenet News. You can post an encrypted message to Usenet, and it will get replicated to all the news servers in the world (many thousands of machines). If Zach reads news, then he can decrypt it. In fact, there exists a newsgroup exactly for this purpose. It is called 'alt.anonymous.messages'.

What are the possible pitfalls of this approach? There are two that I can see:

1) You might be identified as the originator of the message.
2) Zach might be identified while reading the message.

The first problem is easily solved. Posting news anonymously is fairly easy if you have access to a UNIX machine running a NNTP server. If you do not, there are several "mail to news" gateways . All they do is take mail messages and post them to a newsgroup (according to the instructions in the message). We have already seen that it is possible to send mail without identifying yourself, using the anonymous remailer network.

A list of mail to news gateways is available at: <http://www.sabotage.org/~don/mail2news.html>

The second problem is also easily solved (if a bit more expensive). Zach could run his own news server. Since the news server downloads **all** the articles, there would be no way to tell which ones Zach was interested in. However, this is an expensive proposition in terms of both bandwidth and machine resources. An easier way is for Zach to run a program that downloads all the messages in the "alt.anonymous.messages" newsgroup, and then throws away the ones that are not encrypted to him.

I have written such a program.

The program makes use of Internet Config to get the name of your new server. It connects to the server, downloads all the messages in the newsgroup "alt.anonymous.messages", and checks each message to see if it was encrypted to a particular key (set in the preferences). If it is, then the message is saved (still encrypted) to the disk (in the folder set in the preferences). The user can then decrypt the messages at their leisure. This separation of functionality removes several security holes, and allows this program to be distributed freely, since it does not contain any encryption code.

The program and the source are available at <http://www.idio.com/anon/news.html>

Solution #2: Encrypted reply blocks

If you wanted Zach to be able to reply to you, and you didn't have access to Usenet news (or your news service was not reliable), you could include in your messages an "encrypted reply block". This is a series of instructions to remailers to deliver the attached message to you. However, since the reply block is encrypted with the keys of the remailers, Zach cannot determine which remailers you are using, or what your email address actually is.

Here's how you create a reply block:
Create a text file containing:

```
::
Request-Remailing-To: yourname@yourhost.com
```

Encrypt this file using a remailer's PGP key. Prepend the following lines to the PGP message:

```
::
Encrypted: PGP
```

This tells the remailer that the PGP message is encrypted to it.
This is what the reply block looks like:

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.0
MessageID: GthRwUjhkaZ3cDQZUISQNvPbQ9Rmrked

hQCJAz0VsrLnrsHlAQPott1ZTHRzIbNnJQAP4krIn8YUySWbX/H97ysQ8ty8aLzA
BQMqPUuPPfYUo6i5mUm5HJWiK9/BEOHz+RzgaryEt6DeYIZ/Kf4wUXm9mklbBEi5
S7uQny7O+EYffHaX7bLZjNSw/TYIzq4SZr4mC9wQE05Co3S5LiZ5cgrciG6kTo3n
bXfis+VDxAal+vCD2tLWt94o1IVPLiIDj6EPEvXfTDZ+fO1m3+vbAWgxH4RQB90O
TY5kK+BzjcSo9RlMMM5kj6Dejicg0QEaWycd2Q==
=3AD2
-----END PGP MESSAGE-----
```

You then put this bit at the end of your message to Zach. He puts it at the start of his reply, and follows it with his actual reply. He mails the message to the remailer, which decrypts the reply block, reads the instructions, and sends the message to you.

Of course, you can chain remailers together in reply blocks for added security.

I have written a program to automate this process. It is available at <http://www.idio.com/anon/replyblocks.html>.

The program is very similar to the one that sends messages through the remailer network. You choose a routing for the message (or have the program choose one at random), give the destination for the message, and the reply block is created. You can then paste the message into your email program and send it to the recipient.

Solution #3: Get an anonymous email account.

There now exist services that offer free, web-based email accounts. It is easy to get one of these, and there is very little someone can do to find out who you are. If someone could get the logs from HotMail, Yahoo! Mail or NetAddress, they could find out what IP # you were using. They could then go to your ISP and get their logs and (possibly) determine what user was using that IP # at that time, but that would be difficult. To thwart even that possibility, use an anonymizing web proxy like <http://www.anonymizer.com/>

Three of them are available at:
<http://www.hotmail.com>
<http://www.netaddress.com>
<http://mail.yahoo.com>

This has the advantage that you can receive email anonymously as well. However, in accepting the "terms and conditions" for HotMail and Yahoo! Mail, you agree to provide accurate personal identification to the service.

Conclusion

There exist today several methods for communicating anonymously over the internet. All of them work, but they require extra work to use. It is up to each individual user to decide how much effort he or she is willing to expend to ensure their privacy. However, with the programs that are available (and the ones that I have written for this paper), the amount of work required to send private, anonymous email is minimal.

Bibliography

[1] <http://msn.yahoo.com/msn/Computers_and_Internet/Security_and_Encryption/Anonymous_Mailers> A good starting off point for remailer info.

[2] <http://www.sabotage.org/~don/mail2news.html> A mail list of mail to news gateways

[3] <http://www.well.com/user/abacard/remail.html> The anonymous remailer FAQ.

[4] A list of remailers is available at <http://www.publius.net/rlist.html>

[5] Geoffrey Keating has written a Java applet for creating messages to be sent via the remailer network. It is available at <http://www.ozemail.com.au/~geoffk/anon/anon.html>