MENVPRIV: A User-Controllable Privacy-Enhanced E-Mail Network[*]

Kenji Rikitake[†]

September 18, 1997

Abstract

We propose an enhanced e-mail message format to provide a privacy-enhanced transport for e-mail delivery. The format consists of two parts. The first one is *MENV* (Mail with ENVelope). MENV is a message format of encapsulating the envelope information of a message into the body of a new message. MENV adds mail transfer functionality to a mailbox retrieval protocol such as POP3. The second one is *MENVPRIV* (MENV with PRIVacy). MENVPRIV provides protection against spoofing and wiretapping of the entire parts of a message by combining MENV and a cryptographic mail transport such as PGP. By using MENVPRIV a group of user can operate a virtual privacy-enhanced network of e-mail with an adequate privacy protection against the wiretapping and the traffic analysis.

Introduction

Protecting privacy is a vital issue on the Internet. The Internet is a set of interconnected computer networks of various organizations and individuals. The contents transferred over the Internet can be freely monitored because they are not encrypted as default. This openness of the Internet can cause serious privacy invasion by unwanted traffic monitoring.

IPsec is insufficient to protect e-mail privacy

The Internet Engineering Task Force (IETF) are working hard to deploy the new Internet Protocol, IP Version Six (IPv6). The IPv6 specification mandates all the hosts (computers) to be able to validate each packet, and to be able to encrypt the packets unless encryption is prohibited by law or other social reasons. The security enhancement of the IPv6 development has been ported back into IP Version Four (IPv4), the current Internet Protocol, as a feature called IP Security (IPsec).

[*] this paper has been reformatted by LaTeX 2_ε on May 2001.

[†] email: `kenji.rikitake@acm.org` /

$$\text{Host Inside} \xrightarrow{\text{IPsec}} \underbrace{\text{Relay Host}}_{\text{This host must decrypt/encrypt the traffic}} \xrightarrow{\text{IPsec}} \text{The Destination Host}$$

Figure 1: Monitoring by A Relay Host

The IPsec provides a cryptographically secure transport between arbitrary two hosts directly exchanging IP datagrams. The IPsec, however, does not necessarily protect the whole traffic stream between the two application programs. On the Internet, a traffic stream can be relayed among multiple hosts, using two or more exchanges of IP datagrams. An Internet firewall relaying traffics between a private network and the Internet is a good example. If a host inside the private network wants to communicate with a host outside on the Internet, the host inside must send the packet to a relay host of the firewall. Even if the host inside and the relay host securely communicates using IPsec, the relay host must decrypt the packet from the host inside because it has to establish another IPsec link between the destination host on the Internet. The relay host encrypt it again for the destination host. This means that the relay host can monitor the entire contents of the traffic stream (Figure 1).

IPsec is an important enhancement and a critical tool to provide secure communication foundation on the Internet. IPsec, however, does not cover the whole privacy protection requirements. IPsec only provides IP datagram transport security between two hosts. On the Internet the communication stream must be protected by end-to-end basis. Thoroughly encrypting the message before sending it to the communication stream is the only way to keep the privacy of the message contents.

E-mail message structure

Electronic mail (e-mail) is one of the most widely used application on the Internet. On the Internet, an e-mail message format is defined in RFC822, and the major delivery method SMTP is defined in RFC821 and various other enhancement RFCs. In the Internet world, an e-mail message has three parts:

- *Envelope*: the envelope contains information necessary to specify the actual sender and recipients of the message. The mail transfer agents (MTAs) primarily uses information contained in the envelope for actual delivery.
  - *Envelope sender*: this represents the mail address of the entity which invokes the MTA.
  - *Envelope recipients*: this represents one of more mail addresses to whom the MTA delivers the message.
- *Header*: the header contains information to tell the mail user agents (MUAs) about various attributes of the message, e.g., when the message is written, the subject of the message, who writes the message to whom, etc. MTAs may add some headers here to record the delivery process, but MTAs try to minimize the change of the header information.

- *Body*: the body contains information transparently delivered without being modified by either MTAs or MUAs.

Why separating the header and the envelope?

An important idea in separating the header and the envelope of a message is that you don't have to have the same sender or recipient addresses between the two parts. The separation of the envelope and the header is essential in the current practice of handling e-mail.

If you have multiple mailboxes of different addresses. You can identify which mailbox the message is intended to be delivered while the message for all the mailboxes are forwarded into one primary mailbox. MTAs only rewrite the envelope sender and recipients and preserve the sender and recipient information recorded in the header.

Sometimes you have to use multiple sender addresses for different roles. Suppose if you send a message with the header address of your business use from another e-mail account and the message is not properly delivered; where the error message should be directed to? The warning should be directed to the host which you send the e-mail, ¡em¿not¡/em¿ to the sender address written in the header, since the error does not occur on the host represents the business address.

MUAs usually don't care about the envelope information of a message because it is discarded when the message is delivered into a mailbox. So does the human users. So in the MTA-MUA link protocols such as Post Office Protocol version 3 (POP3, described in RFC1939 and RFC1957), the envelope information is not transferred; the protocols handle only the header and body parts.

On the other hand, information contained in the envelope is critical for MTAs to perform delivering messages to the correct recipients and to send back error messages to the proper sender. So in the MTA-MTA link protocols such as Simple Mail Transfer Protocol (SMTP, described in RFC821), the envelope information is explicitly transferred and confirmed in the protocol, while the header and body are transferred together as the message data.

MENV: Mail with ENVelope

MENV is a message format extension to put all the three parts of an e-mail message together into the body of another message. By using MENV, you can use an MTA-MUA link as an MTA-MTA link. This means you can forward messages between multiple mail sites using POP3 (an MTA-MUA protocol) and MENV.

MAILFEED: using POP3 between MTAs

I proposed a POP3-based mail transfer scheme called MAILFEED in 1996 [1]. MAILFEED is a primitive method of using an MTA-MUA link as an MTA-MTA link. The implementation model is as follows:

The MAILFEED delivery agent which puts messages into a POP3 mailbox adds the envelope information for each message on delivery by adding the agent's own header fields on top of the original header information. Each message is retrieved to the MAILFEED recipient agent by a POP3 client. The MAILFEED

```
sender MTA ---> (Envelope + Header + Body) = original message
                |------------+-----------|
                             |
      combined into a MENV-formatted body by MENV Delivery Agent
                             |
                             V
      (Envelope + Header + Body) -> another message (sent to a mailbox)
                                         |
                                         V
                          delivered through a mail transport
                             (MTA-MUA link: e.g., POP3)
                                         |
                                         V
      (Envelope + Header + Body) <- received message (via a mailbox)
                             |
      extracted from a MENV-formatted body by MENV Recipient Agent
                             |
                             V
                    |--------+---------------|
receipient MTA <--- (Envelope + Header + Body) = original message
```

Figure 2: Delivery Process through MENV

recipient agent parses the header fields added by the delivery agent, passes the original envelope information to the MTA, deletes the MAILFEED headers to recover the original message, and passes the original message header and body to the MTA.

The structure of MENV

The structure of MENV is almost the same as MAILFEED. However, MENV does not manipulate the original message header so it can be used over any mail transports which guarantees transparent transfer of message bodies (Figure 2).

MENV application example: on dial-up IP link

MENV is an effective workaround for batch mail transfer on dial-up IP link. Using MENV and POP3 together is an effective candidate to replace batch mail transfer over dial-up link using UUCP.

In most of the cases access from dial-up IP hosts to the Internet Service Providers (ISPs) is restricted as follows:

- The dial-up host can initiate a call to the ISP at any time, but the ISP does not.
- The IP address which ISP assigns to the dial-up host changes on each established link.

So the mail delivery to the dialup host has the following requirements:

- The ISP has to provide a mail storage area for the host.
- The host must initiate the action of message retrieval since only the host can control when the dial-up link is established.
- The IP address of the host is not fixed so an IP-address-based authentication method cannot be applied.

SMTP does not meet the above requirements since an SMTP session is always initiated by the sender side. The ETRN extension of SMTP described in RFC1985 does not precisely meet the above requirements either since the recipient can only send a string to the sender, which is not sufficient for authenticate the link itself.

On dial-up link, UUCP (Unix-to-Unix CoPy) protocol has been widely used since it has the basic functions of remote execution. On UUCP mail delivery you can transfer the envelope information of a mail message. UUCP, however, is hardly supported by the major ISPs for many reasons, mainly because of the technical difficulties and redundancies as follows:

- Running UUCP over a TCP stream is redundant.
- UUCP is a complex protocol which contains serial port conversation, queue management, remote execution, and error correction. Learning UUCP is as difficult as learning the whole mail delivery system.
- UUCP on dial-up link requires its own investment to ISPs since the serial link protocols are its own.

POP3 is an MTA-MUA protocol for delivery to a mailbox. A mailbox is usually only for one mail address. Quite a few ISPs now provide *virtual domain* service which maps between a virtual e-mail address and the actual mailbox of the ISP. Operating a virtual domain is, however, very complex and difficult to manage. It doesn't scall well either. Adding or deleting a mailbox to the virtual domain requires the modification of settings by the ISP, since virtual domain is a mere remapping of mailbox addresses. What if the number of mailbox gets larger?

By using MENV over POP3, ISPs can use an MTA-MUA protocol as an MTA-MTA protocol, so they don't have to care about the mailbox remapping required in the virtual domain operation. The ISP only has to deliver through MENV Delivery Agent. The recipient host only has to forward each message retrieved by the MTA-MUA protocol into MENV Recipient Agent. This simplifies the delivery process and the management burden can be largely reduced.

Implementation issues on MENV format

Various aspects should be considered on actual MENV implementation. Some requirements from the programming point of view are as follows:

- The lines which had to be parsed should be minimized to avoid parsing errors.
- The whole parsing process should be completed by just parsing from the top to the end of the message only once.

The easiest implementation example to satisfy the above requirements is to simply concatenate the envelope sender, the number of the envelope recipients, and the list of recipients before the header and the body. The following is an example:

```
kenji@reseau.toyonaka.osaka.jp
2
kenji.rikitake@acm.org
kenji@rcac.tdi.co.jp
>>>envelope-header separator here
X-Description-1: the header begins from this line
Message-ID: <19970917142125.16411.qmail@reseau.toyonaka.osaka.jp>
From: kenji@reseau.toyonaka.osaka.jp
Date: 17 Sep 1997 23:21:25 +0900
To: kenji.rikitake@acm.org, kenji@rcac.tdi.co.jp
Subject: test
X-Description-2: the header ends at this line

The body begins here.
The body ends here.
```

The above example describes the minimum requirement of the actual MENV implementation as follows:

- The envelope sender (is `kenji@reseau.toyonaka.osaka.jp`)
- The number of the envelope recipient (is two)
- The envelope recipients (are `kenji.rikitake@acm.org` and `kenji@rcac.tdi.co.jp`)
- The separator of the envelope part and the header part. This separator must be a string uniquely distinguishable from the envelope information; a string which begins with a non-mail-address character is a sufficient candidate. (e.g., the character > must not appear in a mail address)
- In RFC822 the header and the body parts of a message must be separated with one blank line, so this syntax can be left untouched.

Putting an identification line of MENV with the specification version number can be a good practice for an actual implementation.

MENVPRIV: MENV with PRIVacy

MENVPRIV is an enhancement of MENV using a cryptographic mail transport, to protect the whole three parts of an e-mail message from spoofing and wiretapping.

Security issues of current cryptographic mail tools

Various cryptographic mail tools such as Pretty Good Privacy (PGP) and Privacy-Enhanced Mail (PEM) are currently available to protect the body of an e-mail message from unwanted monitoring. However, these tools do not usually encrypt the header of the message.

A header field of an e-mail message can reveal a surprising amount of critical information. One of the good examples is the `Subject:` field. Many people put the summary of mail messages into the field, such as:

```
        Subject: supervise'ing qmail-start
        Subject: tcpcontrol on domain names
        Subject: tcpd and/or tcpserver-tcpcontrol on Alpha-Linux
        Subject: tcpserver
        Subject: tcpserver/control questions
```

The above examples are picked up from a public mailing list about the management of an MTA. Just reading the subjects can easily tell the contents of being discussed in the message. The header contains many other critical information such as the timestamp in `Date:` field, the sender in `From:` field, and the recipients in `To:` and `Cc:` fields. Protecting the header fields is critical to protect mail privacy.

Protecting the envelope is critical as well since the actual sender and recipient mail addresses are contained in there. You cannot really hide the envelope from being monitored because MTA must obtain the information to perform the delivery. You can make it harder to find out the envelope information, however, by virtualizing MTAs described in later section.

The structure of MENVPRIV

The structure of MENVPRIV is an enhancement of MENV. The different point is that the mail transport is cryptographically protected (Figure 3). You can use any cryptographic transport which provides protection over the body part of each mail message.

Security requirement on the mail transport of MENVPRIV

The cryptographic mail transport on MENVPRIV is the critical tool to protect privacy between the two MTAs linked by MENV. Since the whole encryption and decryption processes are supposed to work without human intervention, the following points should be emphasized:

```
sender MTA ---> (Envelope + Header + Body) = original message
                |------------+-----------|
                             |
      combined into a MENV-formatted body by MENV Delivery Agent
                             |
                             V
      (Envelope + Header + Body) -> another message (sent to a mailbox)
                                         |
                                         V
                          encrypted with a tool such as PGP
                                         |
                                         V
                          delivered through a mail transport
                             (MTA-MUA link: e.g., POP3)
                                         |
                                         V
                          decrypted with a tool such as PGP
                                         |
                                         V
      (Envelope + Header + Body) <- received message (via a mailbox)
                             |
      extracted from a MENV-formatted body by MENV Recipient Agent
                             |
                             V
                    |--------+---------------|
receipient MTA <--- (Envelope + Header + Body) = original message
```

Figure 3: Delivery Process through MENVPRIV

- Unique mail addresses must be assigned to the sender and the recipient MTAs. Note that the username part of the address should *not* be a trivial one, such as *postmaster* or any existing aliases or mailbox names.
- The private keys for the encryption/decryption process should be protected as secure as possible.
- The users using this MENVPRIV link *must not* rely on the cryptographic mail transport between the two MTAs to protect privacy of their messages. Each message encrypted by the sender MTA is *always decrypted* at the receipient MTA.
- MENVPRIV does not ensure message delivery. This means MENVPRIV itself cannot detect whether a message has been delivered or not. Another protocol to ensure the delivery should be used for the assurance purpose.

MENVPRIV application example: remailer and anonymizer

Not all e-mail users want to disclose their real mail addresses; many people have been using the tools called *anonymizer* or *remailer*, which manipulates the sender information in the header so that the recipients have difficulties on figuring out who the sender is, while keeping the mail reachability to the sender.

From the privacy point of view, the envelope part of an e-mail message contains critical information, as well as the body and the header. The envelope identifies the flow of the message since it contains the actual sender and recipient mail addresses. A third party can perform a common method of attack called traffic analysis, by monitoring and analyzing the envelope information of e-mail messages to discover the privacy of the parties involved in the message transaction.

By using MENV over a mail transport, you can build a virtual MTA-MTA link over a real mail traffic flow, which protects the whole message through the link. This is an effective way to protect the privacy of messages relayed through arbitrary two MTAs. You can no longer monitor the header and the envelope parts of relayed messages, so you cannot collect the information of traffic patterns between the two MTAs. A group of MENVPRIV-protected MTAs can function as a very good remailer, since it is really hard to identify the actual traffic flow.

MENVPRIV can be used for submitting a message, as well as for relaying. In this model, each user encapsulates and encrypts the whole message into another message by using MENVPRIV, and send it via an unencrypted mail transport to the recipient user, and the recipient user decrypt the message. This is useful because it provides a protection scheme of the header and envelope information between two users.

Conclusions and Further Directions

In this paper, we have designed an implementation of MENV, a message format which helps using an MTA-MUA link as an MTA-MTA link. We have also designed MENVPRIV, a privacy-enhanced MTA-MTA link using the format and a cryptographic mail tool. The application of MENV to provide a batch mail transfer method on a dial-up link, and the application of remailing using MENVPRIV are also discussed.

The efficiency of MENV Delivery/Recipient agents should be carefully evaluated, though we expect the agents are at least as efficient as the UUCP delivery agents being used in the ISPs.

The actual implementation of MENV/MENVPRIV should contain not only UNIX variants but also for other operating systems such as Microsoft Windows95 and Windows NT, since this format should be handled by the end-user client programs as well as the servers.

Acknowledgements

I would like to thank Motonori Nakamura from Kyoto University, and all the staff members of TDI RCAC Project for helpful comments on this paper, and TDI Research Center of Advanced Computer Networking Technology for supporting this work.

Comments Welcome

Please feel free to send your comments by e-mail to
<`kenji-menvpriv@k2r.org`>.

Reference

[1] Kenji Rikitake, *Development of MAILFEED: A POP3-based Inter-domain Mail Delivery System*, Proceedings of Internet Conference '96, Japan Society for Software Science and Technology et al., pp. 3-10, July 1996.

Bryan Costales and Eric Allman, *Sendmail*, 2nd Edition, O'Reilly and Associates, 1997.

Marshall T. Rose, *The Internet Message*, Prentice-Hall, 1993.

List of Internet RFCs referred in this paper:

- RFC821: J. Postel, *Simple Mail Transfer Protocol*, August 1982.
- RFC822: D. Crocker, *Standard for the format of ARPA Internet text messages*, August 1982.
- RFC1939: J. Myers and M. Rose, *Post Office Protocol - Version 3*, May 1996.
- RFC1957: R. Nelson, *Some Observations on Implementations of the Post Office Protocol (POP3)*, June 1996.
- RFC1985: J. De Winter, *SMTP Service Extension for Remote Message Queue Starting*, August 1996.