Enabling Email Confidentiality through the use of Opportunistic Encryption

Simson L. Garfinkel

MIT Laboratory for Computer Science
Security Warning

C:|Documents and Settings\simsong\Desktop\presentation.doc contains macros by Simson L Garfinkel

This publisher has not been authenticated and therefore could be imitated. Do not trust these credentials.

Macros may contain viruses. It is always safe to disable macros, but if the macros are legitimate, you might lose some functionality.

Always trust macros from this source.

Disable Macros  Enable Macros  More Info
1992
Pretty
Good
Privacy
How can PGP’s privacy be “pretty good,” if the software is so hard to use?
Security & Usability: The Big Idea

<table>
<thead>
<tr>
<th>Security</th>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

- early computers
- military computers

- Windows
- human-centered
Another View

Increasing Security → Increasing Usability →
Email Encryption:
17 years and counting
1986: MailSafe
RSA Data Security

MailSafe
Public Key Encryption Software

The easy-to-use RSA security overlay for DOS-based electronic mail systems. MailSafe allows you to generate your own RSA public/private keypairs, tamper-proof documents with RSA Digital Signatures, and encrypt documents in RSA Digital Envelopes. MailSafe is compatible with any file transfer method or protocol.
Privacy Enhancement for Internet Electronic Mail

(Obsoleted by 1040, 1113)
S/MIME Version 2 Message Specification

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.
Wide Support for OpenPGP and S/MIME

<table>
<thead>
<tr>
<th>Product</th>
<th>OpenPGP</th>
<th>S/MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eudora</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Microsoft Outlook</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Microsoft Outlook Express</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Netscape Messenger</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Apple Mail</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Why don’t we use encryption?

First we thought the barrier was that people just didn’t have the software.

Now everybody has the software…

.. What’s wrong?
Do we Need Email Encryption?

Email Signing
   Assures integrity
   Assures authorship

Email Sealing
   Prevents unauthorized interception
   Minimizes damage from mail bounces
Typical Applications

Patient-doctor communications
Depositor-bank communications
Student records
Business communications

All email … ?
Opportunistic Encryption

Encrypt if you can.
Don’t encrypt if you can’t.
Don’t bother the user.
SSL
ssh
Just like “telnet” and “rsh,” but with crypto...
Simple Mail Transport Protocol
Simple Mail Transport Protocol (SMTP)

r2# tcpdump -s 4096 -w 25.dmp port 25
r2# strings 25.dmp
EHLO alias.acm.org
250-nitroba.com Hello alias.acm.org [199.222.69.90], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 15000000
250-DSN
250-ETRN
250-DELEVERBY
250 HELP
MAIL From:<simsong@lcs.mit.edu> SIZE=804
250 2.1.0 <simsong@lcs.mit.edu>... Sender ok
RCPT To:<simsong@nitroba.com>
250 2.1.5 <simsong@nitroba.com>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Received: from nitroba.com ([64.7.15.234])
    by alias.acm.org (ACM Email Forwarding Service) with ESMTP
    id COB73880
    for <simsong@acm.org>; Thu, 15 May 2003 16:56:30 -0400
Received: from lcs.mit.edu (localhost [127.0.0.1])
    by nitroba.com (8.12.9/8.12.6) with ESMTP id h4FKuTiI022673
    for <simsong@acm.org>; Thu, 15 May 2003 16:56:29 -0400
SMTP with Opportunistic Encryption

r2# strings out.dmp
T=EHLO nitroba.com
250-mintaka.lcs.mit.edu Hello ip-64-7-15-234.dsl.bos.megapath.net [64.7.15.234], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-STARTTLS
250-DELIVERBY
250 HELP
T=<STARTTLS
220 2.0.0 Ready to start TLS
Massachusetts
   Cambridge
#MIT Laboratory for Computer Science1
SSL Servers1|
SSL Server CA1/0-
   bug-lcs-certificates@LCS.MIT.EDU0
020725210000Z
030725210000Z0
Massachusetts1
Simple Mail Transport Protocol
Key Lessons:

Usable security interfaces have little or no user burden.

Security needs to be default.

Zero (or near-zero) user interface.
Lipner’s Law

“The user base for strong cryptography declines by half with every additional keystroke or mouse click required to make it work.”

- Steve Lipner, (formerly with Trusted Information Systems)

As told by Carl Ellison
(note: this is not Ellison’s law!)
Alice and Bob are security pros!

What happens if I want to send an encrypted message to Jesse?
Sending Sealed Mail to Jesse

• Does Jesse want encrypted mail?
• Does Jesse use S/MIME or PGP?
• How do I get Jesse’s public key?
• Does Jesse still have the matching private key?
• How do I add the key to my email client?
• How do I tell my client to encrypt the message?
• How can I be sure that the message was really encrypted?
Jesse Receiving Sealed Mail

If Jesse’s email client supports encryption, it should be a single click...

... but most mail clients only decrypt the mail when they show it!

Email encryption creates problems for Jesse too!
- Jesse can’t ever throw out his key
- Creates a need for key escrow.
- Prevents searching inboxes.
Sending Signed Mail

Jesse might want to send out signed mail....

... but this may create problems for people who receive his mail:
Hey dude, where’s my message?
PGP/MIME doesn’t improve usability
People ask Jesse to stop sending digitally signed mail, because they can’t read it!

... they don’t know how to read it!
Usability Barriers with Email Encryption

Burden on senders:
  Many decisions and choices

Burden on recipients:
  Encrypted mail is harder to use!
STREAM

A Zero-Click Interface for Encrypted Email
“Zero-Click”

Do the “right thing.”
Do what a typical knowledgeable user would do.

*not* Zero-Visibility
Tell the user what the program is doing.

*not* Zero-Recourse
Give the user an opportunity to correct mistakes.
Remember Macros!

Security Warning

C:\Documents and Settings\simsong\Desktop\presentation.doc contains macros by Simson L Garfinkel

This publisher has not been authenticated and therefore could be imitated. Do not trust these credentials.

Macros may contain viruses. It is always safe to disable macros, but if the macros are legitimate, you might lose some functionality.

Always trust macros from this source.

Disable Macros  Enable Macros  More Info
Remember Macros!

Given a choice, most users will choose to get their work done --- even if it compromises security.
Stream: Bringing Zero-Click to Encrypted Email

Steam sits between the user’s computer and the mail server…

**STMP Proxy** – Encrypting outgoing mail
**POP Proxy** – Decrypting downloaded mail
Stream: Simple Rules for Sending Encrypted Mail

If there is no public/private key pair for the user’s email address, create one!

Put the public key on all outgoing mail.

Encrypt the message if you know the recipient’s key.
This message has a public key

Good news! I figured out how to hide the key from Eudora.

-Simson
The key!
It’s not so big!

Message Source

Received: from localhost.localdomain (192.016.65.157) by nitroba.com (8.12.6/8.12.6) with ESMTP id h3f045fe5033571 for <alg@com.com>; Tue, 15 Apr 2003 09:04:05 -0400 (EDT)

Date: Tue, 15 Apr 2003 09:04:05 -0400 (EDT)

Message-Id: <20030415304.3964345@ntoro.com>

To: alg@com.com

Content-Type: text/plain; charset=US-ASCII

Subject: This message has a hidden key.

From: Simon E. Gerttula! smsgroup@smsgroup.mtc.edu

Return-Path: <privacyactivistcongress-adminlists.csp.com>

Subject: Privacy Activist Congress Discussion Group server down

Message-Id: <20030415285.3550256@ntoro.com>

Date: Tue, 15 Apr 2003 09:04:28 -0400 (EDT)

From: privacyactivistcongress-adminlists.csp.com

Message-Id: <20030415285.3550256@ntoro.com>

Content-Type: text/plain; charset=US-ASCII

Content-Transfer-Encoding: quoted-printable
Stream: Sending Mail 2

As an SMTP Proxy, Stream is well-positioned to make other changes to the message being sent.

Subject: Encapsulation  --- no more confidential subject lines!

Multiple Recipients --- stream sends a separate email message to each recipient. Avoids traffic analysis while allowing mixed encrypted/plaintext recipients.
A Stream Message

To: simsong@acm.org
From: beth@ex.com
Subject: ENCRYPTED MESSAGE FROM beth@ex.com

-----BEGIN PGP MESSAGE-----
Version: PGP 7.1

qANQDFLzQhx9QwkMQD/9W1wCRQk33c4d5DkzdefCMtlni9zLQTGQnI9xLFy9nrLK1emZiJQ7oFit1dORD1vDDQzHZt8hc7pD0v25GL04z6+6C6DuYHiSE7eqf4VgPu/MFjDxZMsDmCEQ2m3710C7CcwTB9XqVYQs1+Uy3VrV1kgkK3cj8Af2zy0IV26dSh8qQS1otAFWWuwyLky5rTX1zhQE7p4FE+wvePOfgb+4dtxfStiTgvsVjZJ/My+BGy2597Rxt1BjBWy/NBDwlM4wyiUce92ErzbXW3IL978PKQYNCbOckRkXrqqsY5qmSn2obcW4IGVGicfhSgIx0A3xUafVQOyDn15BJbtACsxhCe3Hp0q/fIffj7TpojKbilWnExocikbufs+aMpmG0fkrfVJ
Stream: Simple Rules for Receiving Encrypted Mail

Is the message encrypted?
   If so, **decrypt** it (if possible).

Is there a key?
   Is it different from an existing key for this email?
      If so, **alert** the user.
   Is the key on file?
      If not, **add it to the keychain**.
Stream: Log Files

Two kinds of log files:
1. Log files written for debugging.
   (2. Log files for user auditing.)

Stream separates these logs.
Actions can* be “undone” or “explained.”
   Information disclosure is the one thing that cannot be undone.

* Represents area of current research
Stream: Mandatory Encryption

Problem: Sometimes opportunistic encryption isn’t good enough.

Solution: Put the “+” character at the beginning of the Subject line!
- If stream can’t send the message with encryption, it is returned to sender.
Stream: Private Key Migration

Problem: Key generated on laptop may need to be used on a server

Solution: Stream elements can [will] communicate by sending email to each other! Communications will be encrypted.
Stream: Trusted Introducers

Problem: Jesse doesn’t know Sam’s key

Solution:

- Why is Jesse sending mail to Sam in the first place? Because Simson is introducing them!
- When Simson sends email to Jesse and Sam, it [will] sends Jesse’s key to Sam and Sam’s key to Jesse.
- Leverages key distribution on top of existing social mechanisms!
Advantages of Stream

Automatic key management.
  – No explicit key management for the user.
  – easier than “annoying the pig [sic] …”

Works with existing email clients
Cross-platform
Bridges PGP and S/MIME
Stream and Spam

Instead of forcing all users to certify their email, stream provides for certification of the people with whom you are actually exchanging email.

Fights spam without fighting privacy!
Criticisms and Rebuttals

Stream decrypts the mail and leaves it decrypted!

– If your stored e-mail needs to be encrypted, probably lots of things need to be encrypted.
– Address book, other mail, word files … all need security.
– If you don’t provide security for the operating system and application, it doesn’t matter if you provide security for email messages.
– (previously identified) problems with stored encrypted mail:
  • Can’t search it
  • Can’t loose your key
Criticism #2 & #3

Stream doesn’t provide for certified keys, identification of keys, or PKI

Stream provides for continuity of identification.

Most PKI doesn’t work in practice.

Stream doesn’t protect against man-in-the-middle attacks

Use out-of-band verification.
(Call the person up and read the key’s fingerprint.)
Related Work

PGP Inc. has put improved support for zero-click in PGP 8.1 on the Mac.

Alma Whitten has developed a PGP interface for Windows that implements:

- Safe security staging
- metaphor tailoring (visual representations of security functions and data objects.)
Acknowledgements

Jean Camp
   – for her idea on telephone verification.

Rob Miller,
   – for critique on this presentation

and, of course, Jesse Burns
   – for those lovely photos