How to Make Secure Email Easier To Use

Simson L. Garfinkel
Erik Nordlander
Robert C. Miller
MIT CSAIL
Cambridge, MA
{simsong,erikn,rcm}@mit.edu

David Margrave
Amazon.com
Seattle, WA
DavidMA@amazon.com

Jeffrey I. Schiller
MIT Network Services
Cambridge, MA
jis@mit.edu

ABSTRACT
Cryptographically protected email has a justly deserved reputation of being difficult to use. Based on an analysis of the PEM, PGP and S/MIME standards and a survey of 470 merchants who sell products on Amazon.com, we argue that the vast majority of Internet users can start enjoying digitally signed email today. We present suggestions for the use of digitally signed mail in e-commerce and simple modifications to webmail systems that would significantly increase integrity, privacy and authorship guarantees that those systems make. We then show how to use the S/MIME standard to extend such protections Internet-wide. Finally, we argue that software vendors must make minor charges to the way that mail clients store email before unsophisticated users can safely handle mail that is sealed with encryption.

Author Keywords
User Studies, E-Commerce, User Interaction Design

ACM Classification Keywords
D.4.6.c Security and Privacy Protection – Cryptographic Controls;
K.4.4.f Computers and Society – Electronic Commerce – Security;
H.5.2.e HCI User Interfaces – Evaluation/methodology

INTRODUCTION
Email messages are not protected as they move across the Internet. Messages can be misdelivered or intercepted and read by unauthorized or unintended individuals. Email can also be surreptitiously modified—even forged—creating the impression that a person made a statement that she did not. Ordinary Internet email simply does not provide techniques for assuring *integrity*, *privacy* or establishing *authorship*.

Email can be protected by restricting its movement to trusted computers and secure communications links, but such controls are not possible in a large-scale environment with distributed management. As a result, the only way to protect Internet mail is through the use of cryptography. Yet even though cryptographic technology is now built into the email programs being used by most Internet users, few messages that travel over the Internet are actually secured. [12]

We surveyed 470 Amazon.com merchants regarding their experience, knowledge and perceptions of digitally signed email. Some of these merchants (93) had been receiving digitally signed VAT invoices from Amazon for more than a year. The messages were signed with standard S/MIME (Secure Multipurpose Internet Mail Extensions) signatures—signatures that can be transparently verified by programs such as Microsoft Outlook and Netscape Communicator.

We were unable to find a previous survey that explored the attitudes of users towards digitally signed email. We therefore designed our survey to test the respondents knowledge of digital signatures and their interest in the potential benefits that promoters of the technology frequently mention.

Based on our analysis of this survey and our knowledge of current S/MIME implementations, we argue that S/MIME is ready to be used for digitally-signing many routine e-business communications. We suggest a scheme in which the security provided by AOL and various webmail systems could be extended through the use of S/MIME signatures. We also propose modifications to email clients and services in the handling of self-signed certificates, which we believe would further increase the acceptance and use of S/MIME signatures, and might have the benefit of assisting in the fight against so-called spam email.

We argue that there remain interface and implementation problems regarding the use of S/MIME for cryptographically sealing messages to assure privacy. We argue from our survey data that these implementations are not consistent with user expectations or wants, and suggest improvements.

PRIOR WORK
The past 20 years has seen numerous efforts to make secure Internet email possible, if not ubiquitous.

Privacy Enhanced Mail (PEM)
The Internet Activities Board's Privacy Task Force started work in the mid-1980s to develop standards designed to provide end-to-end encryption for email. These standards became known as Privacy Enhanced Mail (PEM), embodied in RFC (Request For Comment) 989 [15] issued in 1987.

The PEM standards were revised twice, with the final set of RFCs [2, 14, 16] published in 1993. These documents defined a signature and encryption standard for ASCII email messages based on public key cryptography using the RSA (Rivest Shamir and Adelman) algorithm.

PEM defined two main protection features: (1) Signed Messages and (2) Signed and Encrypted Messages. Users published their RSA public keys in digital certificates as defined by the X.509 CCITT Standard. These certificates were signed using the private RSA key of a Certifying Authority (CA). The public key of the Certifying Authority was itself placed in another certificate, which itself could be signed by another CA, and so on, composing a Certificate Chain that led back to a single trusted Root.

Because there was no centralized online public key directory in 1989, PEM was designed to operate without one. Instead, each signed message included all of the certificates in the Chain needed to verify the message signature. Once received, PEM implementations would store those accompanying certificates on the recipient's computer. The recipient could then reply to messages with a response that was both signed with the sender's own key and encrypted with the public key of the intended recipient.

Secure Multipurpose Internet Mail Extensions (S/MIME)

When work on PEM stalled shortly after the publication of the PEM standards, RSA Data Security began a new project to re-implement the PEM concept on top of the new MIME mail standards. Called S/MIME, this work was eventually migrated to the Internet Engineering Task Force (IETF) and standardized through RFC2311 and follow-ons. [6, 18]

Because management of a single Root with a single certification policy proved to be problematical, S/MIME implementations do not implement a strict hierarchy of certificates, but instead accommodates any number of trusted Certificate Authorities.

Today support for S/MIME is integrated into many email clients, including Microsoft Outlook and Outlook Express, Netscape Communicator, Lotus Notes, and others. But support for S/MIME is notably missing from AOL's client software as well as from many web-based mail systems (e.g. Yahoo, Google's GMail, Hotmail). On these systems, digitally signed S/MIME messages appear as ordinary messages with an additional attachment named `smime.p7s`, while S/MIME messages that are sealed with encryption are indecipherable.

Pretty Good Privacy (PGP)

In 1991 a programmer in Colorado named Phil Zimmermann released PGP, an email encryption system that performed rudimentary message signing, sealing, and key management. [9] The primary difference between PGP and PEM was the system's approach to certification: whereas PEM specified a centralized Public Key Infrastructure (PKI) with a single root, PGP users can both independently certify keys as belonging to other users, and decide to trust certification statements made by other users. Zimmermann envisioned these capabilities would be used to create a reputation-based "web of trust."

PGP was quite popular in some technical communities, but greater adoption was hampered because PGP was difficult to centrally-manage, PGP did not come with licenses for the patented public key technology, and PGP was a separate program that did not transparently interoperate with existing email systems. These objections were overcome with the introduction of commercial PGP version in 1997 that included all necessary patent licenses and plug-ins that let PGP interoperate with popular email systems such as Microsoft Outlook and Eudora. PGP message formats were eventually standardized by RFCs 1991, 2015 and 2440. [1, 4, 7]

Usability Problems Plague Secure Messaging

Despite the widespread availability of software that implements cryptographically-secured email, secure messaging is not widely practiced. Gutmann suggests that messages employing any kind of end-to-end cryptographic protection comprise only a tiny percentage of the non-spam messages that traverse the Internet each day. [12]

Nevertheless, there is clearly both desire and need for mail security. The 10th GVU WWW User Survey [13] found that a majority of respondents described themselves "very" (52.8%) or "somewhat" (26.7%) concerned about security. A plurality of users (19.1%) identified privacy as "the most important issue facing the Internet."

In recent years Internet users have been beset by a deluge of both unwanted "spam" mail and more importantly by so-called "phishing" messages—messages that purport to be from a respected bank or other financial institution, but which direct the recipients to bandit websites that exist for the purpose of stealing usernames and passwords. [8]

Many observers blame encryption's lack of success on the difficulty of using the software and the lack of a perceived need on the part of users. We have found very few published studies substantiating this belief. Best known is Whitten and Tygar's study of PGP 5.0, [22] in which test subjects found it difficult or impossible to exchange keys and send encrypted mail. Whitten argues in her dissertation [21] that software can successfully employ a technique she calls *safe staging* to teach secure messaging practices to users.

Technology to Solve Usability Problems

The easiest way to solve the problem of secure email is to build an integrated system in which keys are automatically created and distributed whenever new accounts are added by the system's manager, an is the case with Lotus Notes and Groove. [19]

Existing standards-based email systems can be adopted with a proxy between the user's mail client and the mail server that automatically and transparently encrypts mail as it is sent and decrypt mail as it is received. There are many examples of such proxies. [3, 10, 17] Some of these systems

use existing keys and certificates, while others generate and distribute keys and certificates as needed. But despite the technical appeal of such solutions, their existence has not made secure email commonplace. (Notes can use a similar approach to send and receive S/MIME mail over the Internet.)

Another approach is to integrate cryptographic technology directly into the user interface of conventional mail clients. Programs such as Outlook, Outlook Express, and Mozilla Thunderbird provide simple "encrypt" and "sign" toolbar buttons. Our experience has shown that most users either ignore these buttons or are confused by them, as discussed in the next section.

The Certificate Barrier

In order to send mail that is digitally signed, the sender must first create a public/private key pair and obtain a certificate certifying that pair. In order to send mail that is encrypted, it is necessary to obtain the public key of one's intended recipient. Thus, even when there are easy-to-use "encrypt" and "sign" buttons in a program's toolbar, there can still be significant barriers to using that functionality.

This state of affairs seems odd to the initiated. After all, creating keypairs is trivial: Finding hundred-digit prime numbers is a process that can be automated and run at the click of a button. The problem is what happens next: there is nothing to stop a user from placing any name that they wish on the public key after it is created. This creates the opportunity for deception and skulduggery.

The S/MIME system addresses this potential for deceit by requiring users to obtain a certificate from a well-known and presumably reliable CA, assuring that the name on each certificate really belongs to the entity that control's the certificate's matching private key. This is a complex process that frequently involves payment. For example:

- **VeriSign Inc.**, one of the best known CAs, sells a simple certificate called a "Class 1 Digital ID" for $14.95; these certificates expire one year after issuance. [20]
- **Thawte Consulting Ltd.**, a VeriSign subsidary, gives away free "personal email certificates" from its website, but requires that individuals provide a "national identification number" such as a passport number, drivers license number, or social security number—something that many users may not wish to do. Users must then click through more than 20 web pages (some with very difficult-to-find links) and answer complex questions such as "Charset Preference" which many users may not understand.

Because of these difficulties and possible expense, few users go to the trouble of obtaining S/MIME certificates from well-known CAs.

Self-signed certificates can be used with today's S/MIME implementations, but their use is discouraged by most S/MIME clients. These programs display a frightening or confusing message when they receive email that is digitally signed using a certificate issued from an unknown CA—as is the case with mail signed with a self-signed certificate—as shown in Figure 1. Presented with these warnings, recipients generally "push back" on the message sender, asking them to stop using the security technology.

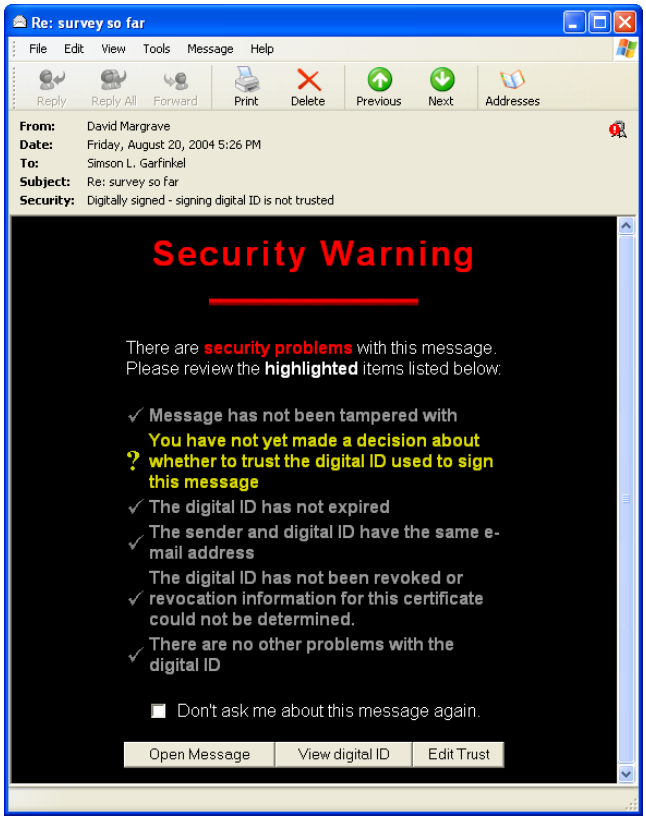

Figure 1. Outlook Express displays a frightening "Security Warning" when it receives email that is digitally signed using a certificate from an unknown CA.

The PGP encryption system does not have this problem: with PGP people create their own self-signed certificates (called *keys* by PGP) which can be used immediately; these certificates can be later signed by others, if desired. Yet it is precisely for this reason that many organizations are hesitant to use PGP in an official capacity: with PGP *anybody* can create and use a key with any name on that key. For example, there are numerous keys on the public PGP key servers that have the name "Bill Clinton" on them, even though it is widely believed that none of the keys were put on the key server by the former US President.

A SECURE MESSAGING USER SURVEY

Our survey consisted of 40 questions on 5 web pages. Respondents were recruited through a set of notices placed by Amazon's employees in the Amazon Seller's Forum. Participation was voluntary and all respondents were anonymous. Respondents from Europe and the United States were distinguished through the use of different URLs. A cookie deposited on the respondent's web browser prevented the same respondent from easily filling out the survey multiple times.

A total of 1083 respondents clicked on the link that was posted in the Amazon forums in August and September 2004. Of these, 470 respondents submitted the first web page, with 417 of those respondents completing all five pages. We attribute this high follow-through rate to the brevity of the survey: each page took on average just two minutes to complete. Nevertheless, because the questions on some pages were viewed by more individuals than questions on other pages, the numbers **Total Respondents** and **No Response** many of our presented tables do not sum to 470. Because the questions later in the survey were not critical for evaluating responses to questions at the start of the survey, we believe that the decision to include the data from partially completed surveys did not introduce systematic bias.

Respondents

Average age of our respondents was 41.5. Respondents were highly educated, with more than half claiming an advanced or college degree. Most described themselves as "very sophisticated" (18.0%) or "comfortable" (63.7%) using computers and the Internet. Roughly half of the respondents had obtained their first email account in the 1990s.

The majority of respondents (94.4%) used computers running Microsoft Windows for email. The two other leading platforms were Apple Macintosh (8.5%) and some kind of mobile computing device such as a cell phone (5.8%).

Initial Results

We presented an analysis of the respondents' attitudes towards digitally signed email to the Financial Cryptography 2005 conference. [11] In that paper we found that a majority (54%) of respondents understood the difference between digital signatures and sealing with encryption; that prior receipt of digitally signed mail significantly increased understanding of that difference; and that having previously received digitally signed email from Amazon increased respondents' overall trust in email. A majority (59%) believed that receipts from online merchants should be digitally signed, while a larger majority (62.7%) felt that bank or credit card statements should be digitally signed *and* sealed. We concluded that companies such as Amazon involved in e-commerce should obtain certificates from well-known CAs and commence signing with S/MIME signatures all email sent to their customers.

This paper discusses aspects of the survey not presented in the FC2005 paper—findings that specifically address issues of Computer Human Interaction (CHI). It then uses those findings to discuss changes that can be made to existing applications and services to improve usability.

The complete survey text with simple tabulations of every question and all respondent comments for which permission was given to reproduce is at `http://www.simson.net/smime-survey.html`.

Few Barriers for Receiving Signed Mail

We found few if any barriers to receiving digitally signed S/MIME mail when the signer's certificate was validated

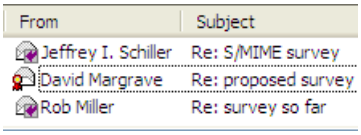

Figure 2. Outlook Express places on a small certificate icon on top of the envelope icon of the middle message to indicate that it is digitally signed.

Figure 3. Apple's OS X Mail application displays a special "Security:" header to indicate if messages are digitally signed.

by a well-known CA. This is because S/MIME-aware mail clients automatically detect the presence of the signature and display an indication that the message is signed, as shown in Figures 2 and 3.

Awareness of Cryptographic Capabilities

We asked respondents which programs they use to read their mail; answers are shown in Table 1. We then asked "Does your email client handle encryption?" (Table 2). The majority (59%) didn't know, while another 9% chose the answer "What's encryption?"

Based on our knowledge of popularly-used email programs,[1] we then split our respondents into two groups: *+S/MIME*, those whose mail readers really do support S/MIME, and *-S/MIME*, those whose mail readers did not. We then compared the responses from these two groups using a logistic regression based on a Chi-Square test to determine if there were statistically-significant ($p < 0.05$) differences in their responses to survey questions. Such differences, when found, are printed **in bold** and necessarily appear in pairs. (Note: The lack of bold type does not indicate that findings are not statistically-significant; it merely indicates that there is no statistically-significant difference in the response between the two groups.)

Respondents with S/MIME-capable mail readers were more than twice as likely to know that their programs were capable of encryption, and half as likely to select the answer "What's encryption?" Nevertheless, the majority of *+S/MIME* respondents (54%) did not know the cryptographic capabilities of the software that they were using.

Almost half of our respondents (44.9%) indicated that they would be willing to upgrade their client in order to "get more protection" for their email (Table 3)—implying that

[1]For example, Microsoft Outlook or Outlook Express (OE), used by 30.6% and 41.8% of our respondents, have supported S/MIME encryption since 1997.

AOL	17.9%
Apple Mail	2.5%
Eudora	6.9%
Evolution	0.9%
Lotus Notes	2.1%
Mozilla Mail	3.2%
Netscape	10.1%
Outlook	30.6%
Outlook Express	41.8%
Total Respondents	435
No Response	(19)

Table 1. "Which computer programs do you use to read your email? Check all that apply."

	ALL	+S/MIME[a]	-S/MIME
Yes	27%	**34%*****	**14%*****
No	5%	5%	5%
I don't know	59%	**54%***	**66%***
What's encryption?	9%	**7%****	**14%****
Total Respondents	446	291	155
No Response	(8)	(1)	(7)

$^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$;

[a]+S/MIME indicates that respondents used an email client that supports S/MIME; -S/MIME indicates respondents used an email client which did not.

Table 2. "Does your email client handle encryption?"

there is a potential market opportunity for makers of new email clients. No statistically-significant differences were observed when we compared the responses to this question for users of different email clients.

Why Don't People Use Email Security?

We asked our respondents on the first page of the survey whether or not they send email that is digitally signed or sealed with encryption. These results are presented in Tables 4 and 5, respectively. Very few (33 out of 470) of our respondents indicated that they digitally signed or sealed their mail "sometimes" or "always."[2]

Although roughly half of our respondents indicated that they didn't use cryptography because they didn't know how, the free-response answers from the more knowledgeable respondents indicated that they either didn't think that encryption was necessary or else that the effort, if made, would be wasted.

I don't because I don't care.

I doubt any of my usual recipients would understand the significance of the signature.

Never had the need to send these kinds of emails.

I don't think it's necessary to encrypt my email & frankly it's just another step & something else I don't have time for!

[2]We believe that the single user who answered "I always send email that is sealed for the recipient" chose that answer in error.

Survey Response (choose one)	
Yes	44.9%
No	21.0%
My email program already gives me adequate protection.	21.0%
Not my decision — my email program is decided by others.	6.1%
I do not use an email program — I only use web mail.	7.0%
Total Respondents	443
No Response	(11)

Table 3. "Would you change or upgrade your email program to get more protection for your email."

Survey Response (multiple selections allowed)	
I always send email that is sealed for the recipient.	0.9%
I sometimes send email that is sealed.	3.5%
I rarely send email that is sealed because it is not necessary for the kind of mail that I send.	16.7%
I rarely send email that is sealed because I just don't care.	7.9%
I don't send email that is sealed because it is too hard to do.	5.7%
I don't send email that is sealed because I don't know how.	41.0%
I don't send email that is sealed because I am worried that the recipient won't be able to read it.	14.3%
I'm sorry, but I don't understand what you mean by "sealed" or "encrypted".	22.0%
Other	3.3%
Total Respondents	454
No Response	(16)

Table 4. "Do you send email that is sealed with encryption so that it can only be read by the recipient? Please check all that apply."

These statistics and free-form comments are particularly significant in light of the fact that 25.2% of our respondents thought that receipts sent by online merchants should be digitally signed, while 33.6% thought that they should both be signed and sealed! Remember, all respondents are themselves Amazon.com online merchants!

Signature Interfaces and Metaphors

As the S/MIME RFCs are silent as to how the presence of a valid digital signature should be displayed, different programs employ different metaphors, as shown in Figures 2 and 3.

We asked our respondents how they would like their email programs to indicate that a message has a valid digital signature. Roughly equal numbers (44% vs. 41%) said that they would like the one-line of text added to the header interface (as shown in Figure 3) as a ribbon or certificate that is shown when the message is displayed in a list (as shown in Figure 2). Roughly a quarter (24%) agreed with the statement that they "would like to see a signature at the bottom

Survey Response (choose one)	
I always send my email digitally signed.	2.2%
I sometimes send email that is digitally signed.	4.2%
I rarely send email that is signed because it is not necessary for the kind of mail that I send.	19.2%
I usually don't because I don't care enough to sign my email.	9.9%
I don't ever send email that is digitally signed because I don't know how.	44.8%
I'm sorry, but I don't understand what you mean by "digitally signed."	24.1%
Other	3.8%
Total Respondents	453
No Response	(17)

Table 5. "Do you send digitally signed mail? Please check all that apply."

of the message, as if it was signed in ink." Users of encryption favored the ink metaphor to non-users, 31% to 22%, a statistically-significant difference.($p < .05$)

We also asked what respondents thought to be "good description," of a digitally signed message would be. Respondents could chose one of five choices or provide their own answer; a plurality of respondents (37.3%) agreed that a digital signature is "like signing your name at the bottom of a message." Next were the 27.5% who agreed that a signature was "like having the message notarized," followed by 30.7% who believed that a signature is "like putting your fingerprint at the bottom of a message." No statistically-significant differences were seen between users and non-users, although we did see statistically-significant differences the *Europe* and *US* samples, with more Europeans (43% vs. 28%) preferring the fingerprint metaphor, and more Americans (30% vs. 15%) prefering the notarized metaphor.

Our analysis of the metaphor question indicates that users don't have strong metaphors or analogies for what it means to digitally-sign mail. This may be a reflection of the fact that the technology itself is somewhat ambiguous, providing both *integrity protection* and *sender identification*. What is frequently left unresolved, in both user interfaces and documentation, is whether or not sending digitally signed mail is meant to convey some form of intentionality as well. This confusion is mirrored in the physical world. For example, to have a document notarized in the United States merely means that the signature on the document was witnessed by a commissioned officer of the state; it is no guarantee of the veracity of the document's contents. Nevertheless, the idea that notarized documents are somehow more trustworthy is a misconception that is commonly presented in American media. In fact, notarized documents are not more likely to be truthful—and neither are messages that are digitally signed.

Storage of Encrypted Messages

Cryptography is not without associated costs and risks. The main cost—increased processing time and message size—has become less significant with faster computers and network connections. But the chief risk—that a lost key will render a sealed message indecipherable—remains.

This risk of encryption appears to be little appreciated. As evidenced in Table 4, only 14.3% of our respondents agreed with the statement "I don't send email that is sealed because I am worried that the recipient won't be able to read it." And despite the fact that we received literally hundreds of free-format comments, none of our respondents suggested that they were afraid to receive sealed mail because they might lose the ability to access it at a later point in time.

The majority of email systems store email on the computer in a format that is similar or identical to the form in which it was received over the network—a practice that dates to the first email systems.

For encrypted mail, storing a message as it is received means leaving that message encrypted with the recipient's public key. As such, it can only be decrypted with the recipient's corresponding private key. This presents users with a conundrum:

- If the recipient loses the private key, the stored email will become inaccessible. Thus, the user has a strong incentive to retain their private keys indefinitely.

- At the same time, it is considered good cryptographic practice to periodically destroy private keys that are used for data encryption. Destroying the key minimizes the chances that the key will be compromised at a later point in time. If a message is intercepted by an attacker and the key is later compromised, the message is compromised as well. The best way to protect against this attack is to routinely expire and destroy message-sealing keys.

We believe that storing email messages encrypted with the original key is a design flaw common to all current S/MIME client implementations.

We surveyed our respondents to see if they were aware that they would no longer be able to access their stored encrypted mail if they lost or destroyed their private key: Most of them (58%) told us that they were not. We then compared how people who reported that they sometimes or always made use of cryptographic protection with those who said that they did not. The results, shown in Table 6, indicates that significantly more *Users* than *Non-users* of cryptography were aware of the risk. Although this is to be expected, it is still troubling that nearly half (40%) were not aware that losing their key meant that they would be unable to read stored messages.

What's worse, a flaw in the design of Microsoft's Internet Explorer 6.0 all but actively encourages users to delete their old certificates when they obtain new ones. When Internet Explorer (versions 4.0 through 6.0) receives a request for client-side authentication, it forces the user to chose from *all certificates for which it has private keys*—even certificates that have expired, as shown in Figure 4. This creates clutter and visual confusion; the only way to eliminate the clutter is to delete the old, expired certificates. Unfortunately, delet-

With today's email systems like Microsoft Outlook and Outlook Express, mail that is sent with encryption can only be displayed if you have the appropriate digital key. Normally this key is stored on your computer, but it can also be stored on a credit card with a built-in computer chip.

If you lose your key or move your email to another computer where you do not have access to your key, the mail can no longer be read.

	ALL	*Users*	*Non-Users*
Yes	33%	**56%*****	**26%*****
No	58%	**40%*****	**63%*****
Don't Know	9%	**4%***	**11%***
Total Respondents	414	102	312
No Response	(6)	(0)	(6)

$^{*}p < .05$; $^{***}p < .001$;

Table 6. "Before you read the paragraph above, did you know that you might lose the ability to read mail sealed with encryption after you had received it?"

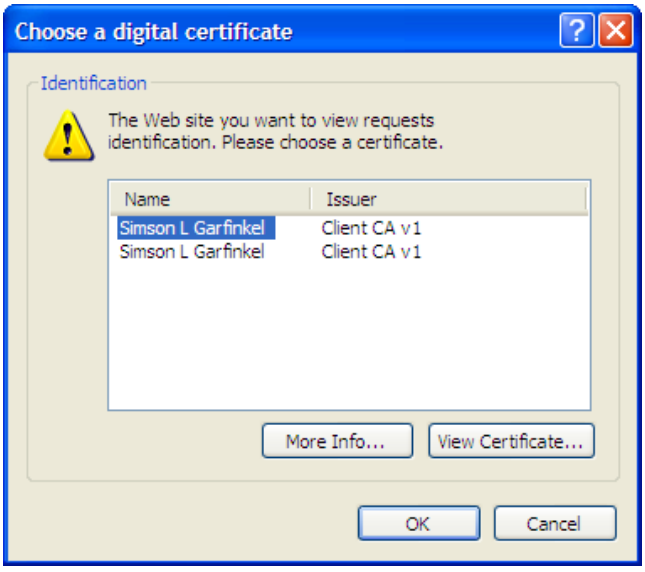

Figure 4. Microsoft Internet Explorer 6.0 asks users to "chose a certificate" before invoking SSL client-side authentication. The user interface displays both current (top) and expired (bottom) certificates without distinguishing them.

ing expired certificates will render indecipherable the stored email that was sealed with their corresponding public keys.

There are many ways to assure access to archived mail that do not require users to retain their keys. One approach is to decrypt the messages before they are stored. An equally secure variant is to unseal each message and then re-seal it with another key—or to store all of the mail on an encrypted file system. We explained five different possible scenarios to our respondents and asked them which they would prefer. We then compared how *Users* vs. *Non Users* answered the question. The results indicate that the majority of respondents want to be able to manually control how their email messages are saved—and that this result is largely consistent for both users and non-users. No email system that we are aware of gives users this kind of control.

Views on Key Escrow

Another way to assure continued access to private keys is to escrow them—that is, to store a secured copy of the private key with a third party. Key escrow has a controversial past: in the 1990s the US Government attempted to get American businesses to adopt mandatory key escrow policy. Originally keys would be escrowed by the government so that law enforcement and national intelligence agencies could have access to data encrypted with them. Eventually the proposals were softened so that the keys could be escrowed with any certified escrow authority. The public's reaction was mixed. The GVU survey found that 18.0% of the respondents "agree strongly" with the statement "I support the establishment of key escrow encryption," while 25.4% stated that they "disagree strongly." [13] Placing these values on a scale of 1 to 5, we found an average response of $\overline{x} = 3.013; n = 1286; \sigma = 1.53$.

We described key escrow to our respondents and asked for their opinion on the practice on a scale of 1 to 5, where 1 was "I like it" and 5 was "I don't like it." The average score for our users was $\overline{x} = 3.2; n = 406; \sigma = 1.22$.[3] We conclude that attitudes towards key escrow have mellowed somewhat in recent years, but that the practice is still controversial.

Key Availability and Migration

One practical problem that arises with encrypted communications is that decryption keys must be available on all computers where the communications are received. One way to solve this problem is to use a single computer for email and store your key on it. A majority of our respondents (88.1%) said that they read email on "A desktop computer reserved for my own personal use," while 32.5% said that they read their email on "My own laptop."

A deeper analysis shows that key migration is likely to be a problem. First, a significant number of respondents indicated that they read email using a shared machine at either a library (21.1%) or at work (21.3%). Many respondents (29%) indicated that they used *both* a desktop computer and a laptop computer; there needs to be some way for these users to easily get the private key from one system to another.

A commonly advocated way to move private keys is external tokens, such as a smart card or a USB dongle. Another solution to store the keys on proxy servers or on shared machines accessed through remote control programs such as Microsoft Terminal Server.

RECOMMENDATIONS

[3]Although we found a statistically-significant bias that *Users* were slightly more likely to be in favor of escrow than *Non Users*, the bias was minimal—just two-tenths of a point.

In our paper on digitally signed mail and recipient perceptions, [11] we recommend that online merchants and other corporations send digitally signed mail using certificates issued by well-known CAs whenever possible. In this paper we explore the ways that changes to the interface of both mail applications and webmail systems could further improve the security of email and help Internet users to realize the benefits of secure email.

Promote Incremental Deployment

Deploying email encryption systems is frequently seen as a chicken-and-egg problem. Senders can't encrypt messages for a recipient unless the recipient first creates a public/private keypair and obtains the necessary certificate. But there is no incentive for a recipient to go to this work unless there is first a sender who wants to send encrypted mail.

No such chicken-and-egg problem exists for senders who wish to sign outgoing mail. Our survey shows that many if not most Internet users have software that will automatically verify S/MIME signatures in a manner that is exactly analogous to accepting a CA-issued certificate during the SSL handshake. Companies sending email can begin adopting S/MIME now and incrementally deploy it.

Although in the 1990's digitally signatures might have been seen as extravagant or expensive technology that required special-purpose cryptographic accelerators to implement on a large scale, those days have long passed. A 2GHz Pentium-based desktop computer can create an more than 700 S/MIME signatures every minute using the freely-available OpenSSL package. S/MIME certificates are also cheap: a single VeriSign Digital ID purchased for $19.95 per year can be used to sign literally billions of outgoing messages, since VeriSign and other CAs charge by the year, not by the message.

Extending Security from the Walled Garden

End-to-end encryption on the Internet was developed because the Internet computers and their links were not a secure infrastructure operated by a single management team. But many of encryption's benefits—identification of sender, integrity of messages, and privacy of message contents—can be accomplished for email sent within closed systems such as AOL and Hotmail. These so-called *walled gardens* can provide security assurances for their content because they use passwords to authenticate message senders and provide reasonable security for message contents.

We believe that several online services have taken initial steps to providing S/MIME-like functionality—at least so far as showing the recipients of some messages that the message senders were properly authenticated.

For example, both AOL's webmail and client interfaces identify email that originated within AOL with a little icon of a human being in the `From:` field, as shown in Figure 5. Mail that comes from the Internet is displayed with a complete Internet email address, as shown in Figure 6, and with the notation "Sent from the Internet" (not shown). This is true even when the email that arrives from the Internet has an `@aol.com` in *From:* field. The AOL network also has the ability to carry "Official AOL Mail," indicated by a blue envelope icon in the user's mailbox, an "Official AOL Mail" seal on the email message, and a dark blue frame around the message, as shown in Figure 7. All of these visual indications provide the user with cues that mail sent from within AOL is somehow different—and presumably more trustworthy—than mail from outside of AOL.

The security of the Official AOL Mail system depends upon the security of the AOL network and the AOL client software. Although the implementation might use S/MIME or a similar digital signature system, it could be implemented with a variety of simpler means as well. Proponents of cryptography might be tempted to argue that the S/MIME-based system would be more secure. But such a system probably would not offer AOL users any more *security*, since AOL's users would still have been placing their trust in the AOL client software to verify the S/MIME signatures.

Other webmail providers do not follow AOL's practice. For example, Google's "GMail" service displays messages with `@gmail.com` addresses that originated *outside* GMail in exactly the same manner as messages that originated from *within* GMail, as shown in Figures 8 and 9. These two cases should be distinguished: mail originating within GMail was sent by a sender who provided a valid username and password, while no such verification was performed for the sender of mail sent from outside GMail. Inside mail is more trustworthy and should be distinguished from outside mail.

We believe most users would benefit from having those systems make explicit guarantees about message integrity, authorship and privacy. An easy way to start is for walled gardens to distinguish between email originating within their walls and email originating from the outside, as AOL does.

S/MIME for Webmail

Moving forwards, we believe that webmail providers such as Hotmail and AOL should work to support S/MIME directly in their systems. Today these services display S/MIME signatures as a small attachment that cannot be easily decoded and understood. Instead, we believe that they should validate the S/MIME signatures and display an icon indicating a signed message has a valid signature.

Once S/MIME messages are properly validated, we believe that the next step is for webmail providers to obtain S/MIME certificates on behalf of their customers and use those certificates to automatically sign all outgoing mail. This is ethically permissible because the webmail provider has verified the identity of the sender, at least to the point of knowing that the sender can receive email at the given email address. Major webmail providers could do this by establishing themselves as CAs and having Microsoft distribute their CA keys through the Windows Update mechanism; smaller webmail providers could work deals with existing CAs to obtain certificates that allow extension of the certification chain. This proposal is somewhat similar to Yahoo!'s DomainKey proposal, [5] except that the signatures would be created with

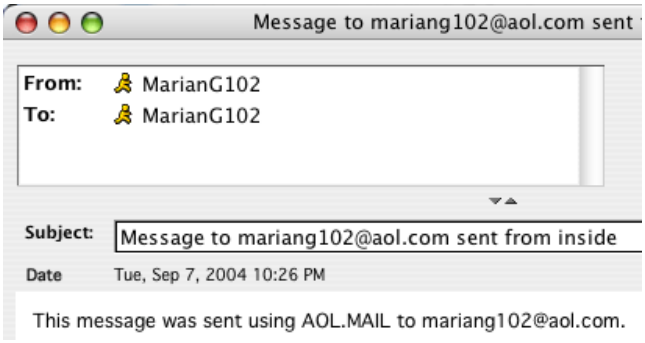

Figure 5. Addresses on messages that originate from within the AOL network, when viewed using AOL's webmail interface.

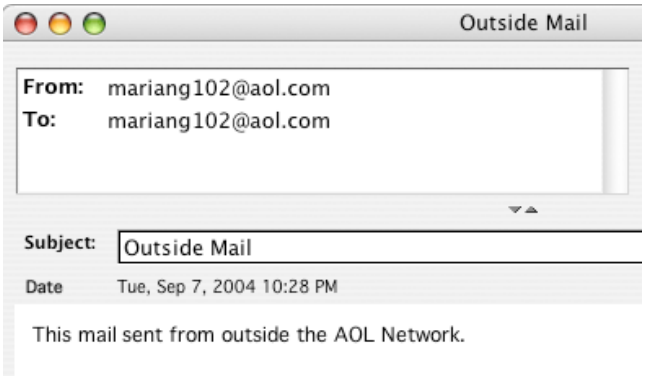

Figure 6. Addresses on messages received from outside the AOL network appear differently than messages originating from inside.

S/MIME and could be verified with software that is already deployed to hundreds of millions of desktops.

Continuity of Identity vs. Certified Identity

As discussed above, the difficulty of deploying certificates to Internet users has long been regarded by many as a kind of chicken-and-egg problem. Systems like S/MIME, which are based on a PKI, require that individuals obtain certificates from well-known Certificate Authorities (CAs). The advantage of obtaining one of these certificates is that the CA public keys are distributed as part of most consumer operating systems and web browsers: the identity of the sender of an S/MIME-signed message can be readily ascertained if the message is signed using the private key that matches a certificate that is issued by one of these CAs.[4] But obtaining a certificate from a CA can be difficult, because the CAs need to establish rigorous processes for verifying the identity of individuals to whom they gave certificates. After all, the service that the CA is selling is identity verification—the CA must be sure that the individual's identity is properly verified!

[4]To make S/MIME verification even easier, the S/MIME standard specifies that every S/MIME-signed message will include a copy of the certificate used to sign the message, eliminating the need for the recipient to explicitly retrieve the certificate from the CA.

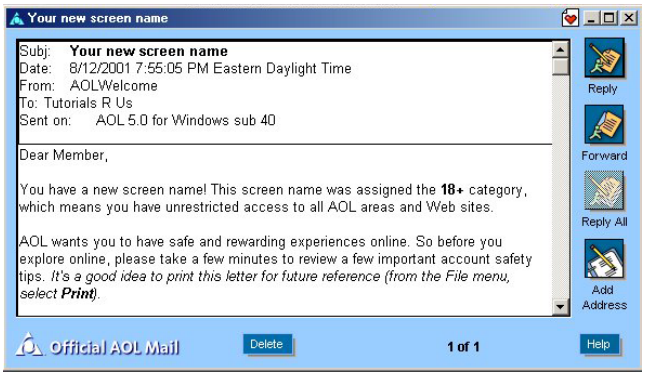

Figure 7. The AOL network has the ability to transport "Official AOL Mail." Such messages cannot be spoofed by outsiders or other AOL members.

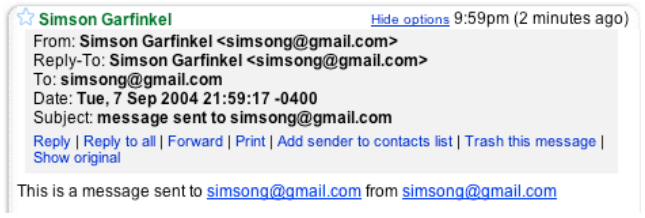

Figure 8. Addresses on messages that originate from within the GMAIL network, when viewed using GMail's webmail interface.

Figure 9. Addresses on messages received from outside the GMAIL network, when viewed using GMail's, appear the same as messages that originate inside.

While there is an obvious benefit for businesses to obtain S/MIME certificates from established CAs, there remains little incentive to for individuals to obtain these certificates.

Just as S/MIME gained acceptance over PEM because it expanded the number of CAs from one to several, we believe that further acceptance can be achieved by rethinking the role of CAs and adopting the "SSH model" [23] of public key certification. Instead of relying on a third party to certify that the name on a certificate is correct, users would accept *all* public keys and simply be notified when a key used to sign a message from a particular email address is *different* from the public key that was used on a prior occasion.

Such a change would make it possible for email clients to generate and use *self-signed certificates* for most low-value personal communications. Self-signed certificates would provide for protection against spammers and computer worms

that forge `From:` addresses, since a message with a forged address would necessarily have either no digital signature or else a signature that was signed by a different key. Even without a trusted CA, the certificates contain a public key that can be used to validate the email message.

We believe that steadily increasing the amount of signed mail in circulation on the Internet cannot help but improve the overall security of Internet mail, increase the confidence that users have in mail, and provide spam-fighters with powerful new tools that they can incorporate into their filters.

ACKNOWLEDGMENTS

The idea of for this survey was originally suggested by Jean Camp when she was at the Harvard's Kennedy School of Government. Sean Smith at Dartmouth College and John Linn at RSA Security provided useful comments on the survey's design and implementation. We are thankful to David C. Clark, Karen Solins and Min Wu for their initial comments on our survey questions, and to Steven Bauer, David C. Clark and David Krikorian for comments on the final paper. John-Paul Ferguson at MIT and Elaine Newton at CMU provided useful comments on our statistical methods. Beth Rosenberg graciously copyedited this paper; all errors that remain are ours. Amazon.com's computer security group and nearly 200 other Amazon employees graciously pretested an earlier version of this survey. Apart from allowing its employees to participate in the study, Amazon.com did not contribute monetarily to this study and does not necessarily endorse the conclusions and recommendations herein.

REFERENCES

1. D. Atkins, W. Stallings, and P. Zimmermann. RFC 1991: PGP message exchange formats, August 1996. Status: INFORMATIONAL.

2. D. Balenson. RFC 1423: Privacy enhancement for Internet electronic mail: Part III: Algorithms, modes, and identifiers, February 1993. Obsoletes RFC1115. Status: PROPOSED STANDARD.

3. Ian Brown and C. Richard Snow. A proxy approach to e-mail security. *Software Practice and Experience*, 29:1049–1060, October 1999.

4. J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. RFC 2440: OpenPGP message format, November 1998. Status: PROPOSED STANDARD.

5. Mark Delany. Domain-based email authentication using public-keys advertised in the dns (domainkeys), August 2004. INTERNET DRAFT.

6. S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, and L. Repka. RFC 2311: S/MIME version 2 message specification, March 1998. Status: INFORMATIONAL.

7. M. Elkins. RFC 2015: MIME security with pretty good privacy (PGP), October 1996. Status: PROPOSED STANDARD.

8. Federal Trade Comission. Identity thief goes "phishing" for consumers' credit information, July 2003. http://www.ftc.gov/opa/2003/07/phishing.htm.

9. Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly & Associates, 1994.

10. Simson L. Garfinkel. Enabling email confidentiality through the use of opportunistic encryption. In *National Conference on Digital Government Research*, 2003.

11. Simson L. Garfinkel, Jeffrey I. Schiller, Erik Nordlander, David Margrave, and Robert C. Miller. Views, reactions, and impact of digitally-signed mail in e-commerce. 2005.

12. Peter Gutmann. Why isn't the internet secure yet, dammit. In *AusCERT Asia Pacific Information Technology Security Conference 2004; Computer Security: Are we there yet?*, May 2004. http://conference.auscert.org.au/conf2004/.

13. GVU. GVU's tenth WWW user survey results, 1999. http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/.

14. S. Kent. RFC 1422: Privacy enhancement for Internet electronic mail: Part II: Certificate-based key management, February 1993. Obsoletes RFC1114. Status: PROPOSED STANDARD.

15. J. Linn. RFC 989: Privacy enhancement for Internet electronic mail: Part I: Message encipherment and authentication procedures, February 1987. Obsoleted by RFC1040, RFC1113. Status: UNKNOWN.

16. J. Linn. RFC 1421: Privacy enhancement for Internet electronic mail: Part I: Message encryption and authentication procedures, February 1993. Obsoletes RFC1113. Status: PROPOSED STANDARD.

17. Mindy Pereira. *Trusted S/MIME Gateways*. Dartmouth College, May 2003. Senior Honors Thesis: Winter/Spring 2003, Department of Computer Science, Dartmouth College.

18. B. Ramsdell. RFC 3851: Secure/multipurpose internet mail extensions (s/mime) version 3.1 message specification, July 2004.

19. Jon Udell. How ray ozzie got his groove back. *openp2p.com*, October 24 2000.

20. VeriSign. Digital ids for secure email, 2004.

21. Alma Whitten. *Making Security Usable*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2004.

22. Alma Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *8th USENIX Security Symposium*, pages 169 – 184, 1999.

23. T. Ylonen. SSH - secure login connections over the internet. Proceedings of the 6th Security Symposium) (USENIX Association: Berkeley, CA):37, 1996.

CONTRIBUTION AND BENEFITS STATEMENT

Presents findings of a study exploring experience and attitudes towards digitally signed mail. Examines interface and cryptographic techniques that could be used to increase secure messaging usabilty and acceptance.