Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express

Simson L. Garfinkel
MIT CSAIL
Cambridge, MA 02139
simsong@csail.mit.edu
http://www.simson.net/

Robert C. Miller
MIT CSAIL
Cambridge, MA 02139
rcm@mit.edu
http://www.mit.edu/˜rcm/

Draft of 2005/02/04 13:39

Abstract

Secure email has struggled with signifcant obstacles to adoption, among them the low usability of encryption software and the cost and overhead of obtaining public key certificates. Key continuity management (KCM) has been proposed as a way to lower these barriers to adoption, by making key generation, key management, and message signing essentially automatic. We present the first user study of KCM-secured email, conducted on naive users who had no previous experience with secure email. Our secure email prototype, CoPilot, color-codes messages depending on whether they were signed and whether the signer was previously known or unknown. We find that this interface makes users significantly less susceptible to social engineering attacks overall, but new-identity attacks (from email addresses never seen before) are still effective. Also, naive users *do* use the Sign and Encrypt button on the Outlook Express toolbar when the situation seems to warrant it, even without explicit instruction, although some falsely hoped that Encrypt would protect a secret message even when sent directly to an attacker. We conclude that KCM is a workable model for improving email security today, but more work is needed to alert users to certain attacks.

1 Introduction

After more than 20 years of research, cryptographically protected email is still a rarity on the Internet today. Usability failings are commonly blamed for the current state of affairs: programs like PGP and GPG must be specially obtained, installed, and are generally considered hard to use. And while support for the S/MIME mail encryption standard is widely available, procedures for obtaining S/MIME certificates are onerous because of the necessity of verifying one's identity to a Certification Authority.

Key Continuity Management (KCM) [5] has been proposed as a way around this conundrum. Under this model, individuals would create their own, uncertified S/MIME certificates, use these certificates to sign their outgoing mail, and attach those certificates to outgoing messages. Correspondents who wish to send mail that is sealed with encryption are able to do so because they posses the sender's certificate. Mail clients (e.g. Outlook Express, Eudora) alert users when a correspondent's certificate changes.

We conducted a user test of KCM with 43 email users who had no previous experience or knowledge of cryptography and email security. Using a scenario similar to that of Whitten and Tygar's *Why Johnny Can't Encrypt* [13] study, we show that naïve subjects generally understand the gist of digitally signed mail, and further understand that a changed key represents a potential attack. However, such subjects are less equipped to handle the circumstances when a new email address is simultaneously presented with a new digital certificate.

We conclude that KCM is a workable model that can be used today to improve email security for naïve users, but that work is needed to develop effective interfaces to alert those users to a particular subset of attacks.

2 Background

In their seminal 1976 paper disclosing the invention of public key cryptography [1], Diffie and Hellman wrote somewhat optimistically that their invention "enables any user of the system to send a message to any other user enciphered in such a way that only the intended receiver is able to decipher it." Diffie and Hellman proposed that public keys would be placed in "a public directory." The following year (1977), Rivest, Shamir and Adelman introduced what has come to be known as the *RSA Cryptosystem*, an algorithm that provided a practical realization of the kind of public key system that Diffie and Hellman foresaw. In 1978 Loren Kohnfelder proposed in his MIT undergraduate thesis [6] that certificates could be used as a more efficient and scalable system for distributing public keys.

With these three inventions—public key cryptography, the RSA algorithm, and certificates—the basic building blocks for a global secure messaging system were in place. Yet more than 20 years later, after the deployment of a global Internet and the creation of low-cost computers that can perform hundreds of RSA operations in the blink of an eye, the vast majority of the legitimate (ie: non-spam) mail sent over the Internet lacks any form of cryptographic protection.

2.1 Secure Messaging: What's Needed

Despite the apparent scarcity of cryptographically protected email on the Internet today, literally dozens of systems have emerged to provide some kind of secure messaging functionality. Nearly all of these systems implement a common set of functions that allow users to:

- *Sign* an outgoing email message using the sender's private key.
- *Seal* an email message with the intended recipient's public key, so that the message can only be opened by someone in possession of the recipient's corresponding private key.
- *Verify* the signature on a digitally signed message that has been received, a process that involves use of the sender's public key.
- *Unseal* a message that has been sealed for the recipient using the recipient's public key.

Traditionally the operation of *sealing* has been called *encrypting*.

Although these four functions represent the key functions required for any secure messaging system based on public key cryptography, various systems implement additional functions. PGP, for example, gives users the ability to create public/private key pairs, to "sign" the public keys belonging to other users, and to create "revocation certificates" for repudiating their public key pairs at some later point in time. Lotus Notes, on the other hand, does not give users the ability to create their own keys at all: Notes keys are created by the Notes administrator and distributed to users in a special "identity file."

Most S/MIME users create their public/private key pairs using a web browser pointed to the web page of a trusted third parties called Certification Authorities (CAs). The browser makes the public/private key, sends the public key to the CA, the CA signs the key and returns to the user (either through email or through a download) a so-called "Digital ID" that contains both the user's public key, a signature from the CA, and an identity to which that public key it bound.

2.2 Why Johnny Can't Encrypt

In 1999 Whitten and Tygar published "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0." [13] The paper included a definition of usable security software, an enumeration of five properties that make it difficult for software to combine security and usability, and a cognitive walkthrough of the commercial email security program, PGP 5.0 for the Macintosh. But the paper was best known for its usability study of 12 naïve users who were given the task of creating PGP keys and using those keys with PGP and Eudora to send digitally signed and sealed email to a specified recipient whose key was downloaded from a PGP key server.

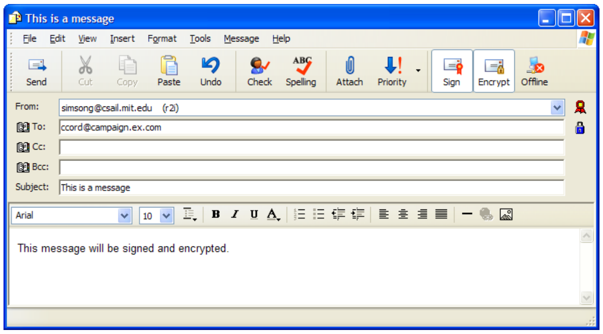
Although 11 of the 12 *Johnny* participants were able to create public/private keypairs and 10 were able to make their keys available to other members of their "team," only four were able to successfully send properly sealed email, and only one was able to complete all of the tasks that the authors considered necessary for proper use of the secure messaging software. The authors concluded that effective security tools require different usability standards than non-security software and "that PGP 5.0 is not usable enough to provide effective security for most computer users, despite its attractive graphical user interface, supporting our hypothesis that user interface design for effective security remains an open problem."

2.3 S/MIME

Many of the usability failings identified by Whitten and Tygar in PGP 5.0 and Eudora simply do not exist in programs like Microsoft Outlook, Outlook Express (OE) and Netscape Communicator, all of which have integrated support for the S/MIME email security standard [8]. For example, OE has buttons labeled "Encrypt" and "Sign" in the toolbar of the Compose Message Window (Figure 1). To digitally sign a message, the user only needs to click the button labeled "Sign." Likewise, to seal a message for a recipient, only the "Encrypt" button need be clicked.

Of course, programs like Outlook Express can only sign outgoing mail if the user has previously obtained and installed a Digital ID. If the user clicks the "Sign" button and tries to send a message without having a Digital ID on the sending computer, a warning message appears (Figure 2). Trying to send an sealed message to a recipient for whom there is no Digital ID on file in the sender's OE6 Address Book generates a similar warning, this time giving the user a choice between aborting the send or sending the message without encryption (Figure 3).

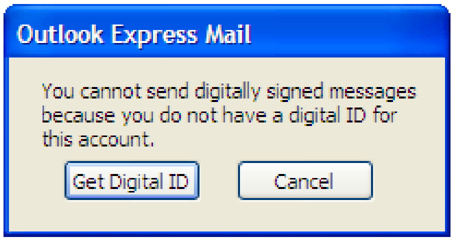
S/MIME systems address other usability errors that Whitten and Tygar identified as well. Whereas PGP 5.0 had support for two incompatible key types, S/MIME supports but one. Whereas message unsealing with PGP 5.0 was manual, unsealing with OE and similar programs is automatic: if the mail client receives a sealed


This is a message
File Edit View Insert Format Tools Message Help
Send Cut Copy Paste Undo Check Spelling Attach Priority Sign Encrypt Offline
Arial
10

From: simsong@csail.mit.edu (r2i)
To: ccord@campaign.ex.com
Cc:
Bcc:
Subject: This is a message

This message will be signed and encrypted.

Figure 1: The toolbar of Outlook Express 6 allows messages to be signed or sealed simply by clicking a button. The little certificate icon to the right of the “From:” field indicates that the message will be signed, while the little padlock icon next to the “To:” field indicates that the message will be sealed for the recipient.



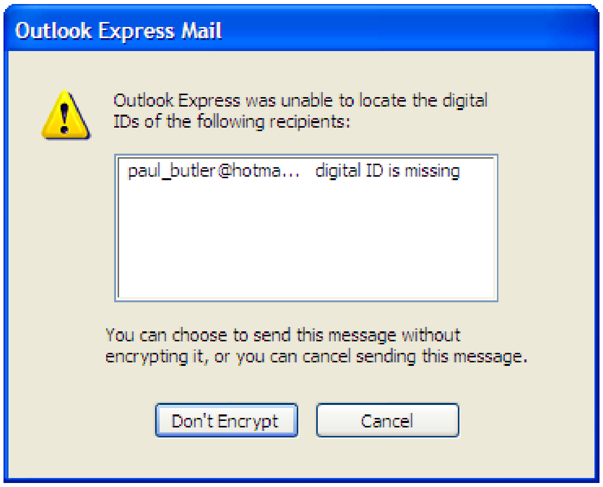
You cannot send digitally signed messages because you do not have a digital ID for this account.

Figure 2: This warning appears if an OE6 user attempts to send a signed message and there is no Digital ID on file for the sender.

message and the client possesses the matching private key, the message is unsealed. The S/MIME standard even automates a rudimentary form of key distribution: when a digitally signed message is sent, that message comes with a copy of public key certificate that can be used to verify the message. This certificate is automatically copied from the message into the user’s address book, allowing a the recipient of a signed message to respond with a sealed one simply by hitting the “reply” key, should the recipient wish to do so.

What S/MIME did not simplify, however, was the process of creating a public/private key pair. Indeed, the standard made this process more complicated.

Whereas the PGP key certification model allows individuals to create their own public/private key pairs and then optionally have those public keys certified by one or more individuals, S/MIME implementations today all but require that individuals receive certificates, called *Digital IDs*, from one of several well-known Certification Authorities (CAs). Although programs like Outlook Express can be configured to trust individual Digital IDs or, indeed, entire new CAs, the default behavior of these programs when receiving messages that are signed by Digital IDs issued by otherwise unknown CAs is to display frightening and confusing warning messages, such as those shown in Figure 4.



Outlook Express was unable to locate the digital IDs of the following recipients:

paul_butler@hotma... digital ID is missing

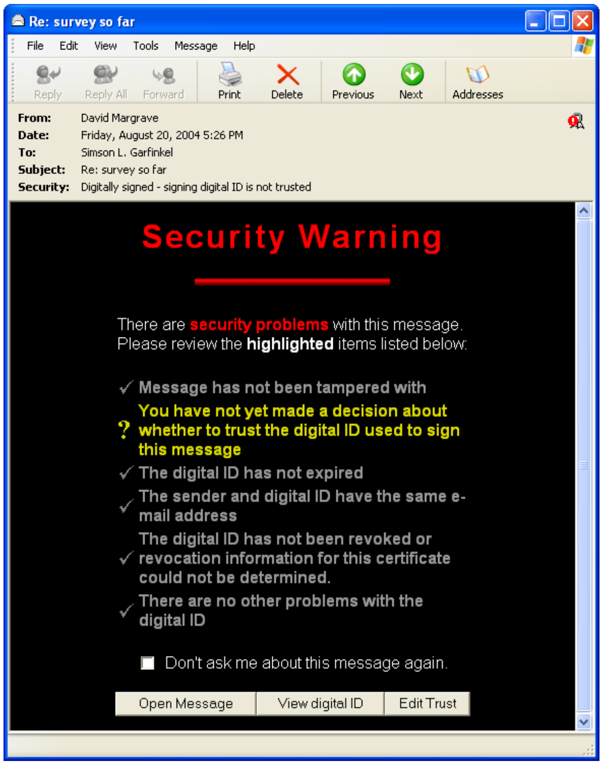
You can choose to send this message without encrypting it, or you can cancel sending this message.

Figure 3: This warning appears if an OE6 user attempts to send a sealed message to a recipient and there is no Digital ID for the recipient in the sender’s Address Book

As noted above, these frightening messages do not appear when users obtain a Digital ID from a well-known CA such as VeriSign or Thawte. But obtaining such an ID is a complex and time consuming process. Even to obtain a free certificate issued by Thawte requires that users click through legal agreements and forms on approximately 20 web pages, enter a “National Identification Code (NIC),” and prove that they can receive email at a given address. Surprisingly, Thawte requires a national identification number even for its “Freemail” certificates which do not include the subject’s legal name in the certificate and only claim to identify the user’s email address, according to the Thawte Certificate Practices Statement.[10] Obtaining a certificate from VeriSign requires a similar procedure and, additionally, the payment of a $19.95 annual fee. Many users are unwilling to make this investment in time, money, and the potential loss of privacy.

Widely available tools such as OpenSSL allow sophisticated users to create and manage their own certificates. Although these tools are cumbersome to use today, they could be automated. For example, programs like Outlook Express could be designed so that they automatically create self-signed Digital IDs whenever they are first used with a new email address, similar to the way that SSH installations on FreeBSD and Linux automatically create new host keys the first time that the host boots.

Another alternative would be for programs like Outlook Express to tightly integrate with CAs that issue free S/MIME certificates. Thawte, for example, issues so-



From: David Margrave
Date: Friday, August 20, 2004 5:26 PM
To: Simson L. Garfinkel
Subject: Re: survey so far
Security: Digitally signed - signing digital ID is not trusted

Security Warning

There are security problems with this message.
Please review the **highlighted** items listed below:

✓ Message has not been tampered with
? You have not yet made a decision about whether to trust the digital ID used to sign this message
✓ The digital ID has not expired
✓ The sender and digital ID have the same e-mail address
✓ The digital ID has not been revoked or revocation information for this certificate could not be determined.
✓ There are no other problems with the digital ID

□ Don't ask me about this message again.

Figure 4: Outlook Express displays a frightening "Security Warning" when it receives email that is digitally signed using a certificate from an unknown CA.

called "Freemail" certificates based on the ability to receive mail at a given address. Instead of taking the user to a web page, the Outlook Express "Get Digital ID" button *could* create an X.509 Certificate Signing Request and send it to Thawte; Thawte could sign the CSR and email it back to the From: address. The act of being able to receive the certificate would confirm the requester's ability to receive mail at the given address. OE6 could then automatically install the certificate, making the entire process automatic. But this is just a hypothetical alternative, for no email client and no CA have created such a system.

Thus, it seems that the widespread deployment of S/MIME has simply replaced one usability problem with another. Although most Internet users now have very clean support for the fundamental requirements of a secure email system built into their email clients, using this system requires obtaining a Digital ID from an established CA — an obstacle that pushes away the majority of users.

2.4 Key Continuity Management

Gutmann and others have suggested relaxing the stringent identity certification rules under which S/MIME and other X.509-based systems operate in favor of a less ambitious certification regime that he calls Key Continuity Management (KCM). [5].

The idea of KCM is for applications ignore certification X.509 certification chains, and instead become directly aware of the public key that each certificate contains. Applications such as email clients or web browsers would remember a server's public key the first time that the key is presented. Subsequent uses of that same key would require no user intervention. Users *would* be notified, on the other hand, if a server's key suddenly changed. This is precisely the key certification model that is used by the popular SSH remote access program. [14] It is also quite similar to the "Resurrecting Duckling" model proposed by Stajano and Anderson for ad-hoc wireless networks. [9]

Implementing KCM for secure email is relatively straightforward. All that is required is a mail client that automatically creates a self-signed key pair whenever the user configures in a new From: address. The public key of this key pair is then attached to all outgoing mail messages. When mail is received that has an attached key, that key is automatically stored in the user's address book. When outgoing mail is sent to an address for which a public key exists, the outgoing mail is automatically sealed. Likewise, when mail is received, it is automatically unsealed and signatures are verified. Key changes are reported to the user. Given the preceding discussion of S/MIME, it should be clear how existing S/MIME clients could be retrofit to implement this model. We have previously demonstrated a transparent POP and SMTP proxy that implements a version of this model for PGP. [3]

KCM for secure email represents a radical departure from existing secure email methodologies. Despite the problem of key establishment, the real usability strength of S/MIME and PGP is that public keys, once certified, are trustworthy. In an S/MIME world, a certificate that claims to be from `marketplace-messages@amazon.co.uk` and is signed by the VeriSign Class 1 CA almost certainly was issued to an individual or organization that had the ability to receive email messages sent to the `marketplace-messages@amazon.co.uk` address and that the distinguished name on the certificate is unique within the owner's domain (this is what VeriSign promises in its Relying Party Agreement). We have previously shown that the majority of people who received such signed messages understood that digital signatures increased the security of the signed message, and, it turns out, also increased the recipient's trust in email. [4]

No such assurances can be made in a KCM system, as there is no trusted third party that implements a certification practices policy and that can be held responsible for violations of that policy. In a KCM world, users are on their own — just as SSH users are today. If your laptop tells you that the server's SSH public key has changed, the change might be because somebody has reinstalled the server's operating system. Or the change might be because you are trying to access your server from a wireless "hot spot" at the DEFCON hacker convention and somebody is trying to mount a sophisticated man-in-the-middle attack to steal your username and password. There's really no way to be sure. The hope of KCM is that this apparent decrease in security is made up for by the fact that KCM is more likely to be used than existing systems that require strong-certification, and is further more likely to scale in a world made up of multiple organizations that are unable or unwilling to cross-certify.

2.5 *Johnny 2*

The study described in this paper, *Johnny 2*, is based on a radical reinterpretation of Whitten and Tygar's *Johnny* results: that the usability problems uncovered in the *Johnny* user study were not driven by the PGP 5.0 program itself, nor by the lack of training offered within the program, but by the underlying key certification model used by PGP. *Johnny 2* seeks to determine whether or not the fundamental usability barriers that appear to be inherent in secure messaging can be solved by replacing third-party certification with Key Continuity Management.

We therefore set out to replicate Whitten and Tygar's original *Johnny* study as closely as possible, but using a system that implements KCM. We knew that the KCM technology could be made to work. Our question was whether or not making it work would discard many of the advantages of existing secure email systems.

Whitten and Tygar interpreted their *Johnny* results as an indication that security software has specific usability problems that make it different from non-security software. As such, the authors reasoned, security software must be developed with special care and using specific techniques. Whitten then developed two such techniques—*safe staging* and *metaphor tailoring*—which were shown to be effective for helping untrained users to use and even enjoy the key certification approaches to secure messaging that are exemplified by PGP. [11]

Although it may be possible to use safe staging and metaphor tailoring to teach untrained users the ins-and-outs of third-party key certification, it may be that these techniques are not needed for for the sending and receiving of secure email if the underlying trust model can be revisited.

3 Testing KCM

While it is relatively easy to understand how KCM could be implemented on top of existing email systems and secure messaging protocols, to our knowledge the KCM model has never been formally tested for usability.

3.1 Testing Goals

Just as SSH users today occasionally face trust decisions that must be resolved, we believe that KCM email users would likewise be faced with occasional trust decisions that would need to be resolved. These trust decisions would most likely include:

- What to do when a user that you are exchanging mail with changes their key? Do you trust the user's new key based on the content of their message, or distrust the message because the key has changed?
- What to do when a new key and a new email address are used for the very first time? Trust it? Be suspicious? In many cases a never-before-seen key with a never-before-seen email address will be legitimate. On the other hand, the new address might be an attacker trying to execute a phishing attack.
- What to do when a correspondent who normally sends you signed mail sends you a message that is not signed? Do you trust it, and assume that they made a mistake? Or do you not trust it, and assume the unsigned message is an attack?

When these circumstances arise in the context of an intentional attack, we will call these a *new key attack*, a *new identity attack*, and an *unsigned message attack*.

Our user test is designed to see how naïve users would answer these questions. In the course of the experiment, we discovered that we could further answer some questions that have been lingering among researchers and practitioners in the field of email security since the 1980s discussions regarding Privacy Enhanced Mail.

Those questions are:

- Do users readily understand the difference between signing and sealing?
- If computer users can trivially sign and/or seal their email correspondence by clicking a button, and the situation seems to warrant the extra security, will they click that button?
- If users can seal confidential information before they send it, will they be less careful about the actual recipients of the information?
- Do the current mechanisms for viewing certificates that are built into the Windows operating system provide users with information is a manner that is clear and understandable?

We were able to suggest qualitative answers to these questions.

3.2 The *Johnny 2* Scenario

Our original goal was to replicate the *Johnny* experiment as closely as possible using the record in the original Johnny paper [13], the longer CMU technical report [12], and Whitten's dissertation [11] so that our results could be directly compared to hers. For example, experimental subjects were recruited using Whitten's original posters and paid the same fee as were Whitten's participants. Subjects in our experiment played the role of a volunteer in a political campaign who is charged with sending out a schedule of a candidate's campaign appearances—exactly the same scenario that Whitten used for her participants. Our consent form and briefing materials contained identical language to Whitten's, so that our subjects would receive the same statements regarding the need for secrecy and security as did Whitten's participants. We even used the same fictional names for our campaign workers!

We were forced to modify the original *Johnny* scenario because Whitten's experiment did not feature any attacks. In the original scenario, the participant, playing the role of the Campaign Coordinator, was merely told to create a key, get a co-workers key, and send a message that was signed and sealed. Participants who completed this task were sent additional instructions by email—for example, to back up their keys to a floppy disk or create a revocation certificate. Such instructions would not be interesting in a KCM-world where public keys are automatically created, distributed and managed: there would be nothing for our users to do!

We devised a modified scenario using the same characters as Whitten did in *Johnny*. In our scenario, the experimental subject still plays the role of the Campaign Coordinator who is charged with sending out the candidate's schedule. The subject receives the schedule as an email message from Maria Page, the Campaign Manager, and is charged with sending out copies to the other campaign team members at specific times. The subject is further told that he or she should send out a copy of the schedule to any member of the campaign team who asks for the schedule. But there's a catch: Unknown to the subject, an attacker from the other campaign is trying to obtain a copy of the candidate's schedule.

Our opposing campaign attacker uses an attack very similar to an attack described by the famed hacker Kevin Mitnick in his book *The Art of Deception*. [7] The attacker poses as a campaign member and sends email to the Campaign Coordinator claiming that his email is not working. At the same time, the attacker mounts a denial-of-service attack on the Campaign's telephone lines, so that the volunteer cannot place an outgoing phone call to seek advice or help. The attacker uses a series of personas in an escalating attack. The messages are described in Table 2.

3.3 Adding KCM to S/MIME with CoPilot

In the *Johnny 2* scenario the fictional Campaign has decided to equip its computers with CoPilot, an add-on program that manages S/MIME keys and address book. CoPilot automatically tracks the association between Digital IDs and email addresses, alerting the user with color-coded messages when there are changes. CoPilot also creates new Digital IDs for the user when it discovers that the user has configured a new email address, eliminating the need for the user to visit a CA website.

The name "CoPilot" comes from the idea that CoPilot functions as a kind of security expert who watches over the user's shoulder and understands how to perform a variety of security-relevant tasks. In this case, CoPilot takes not of the serial numbers that is on each Digital IDs and alerts the user when new IDs are seen, presenting this information in the context.

CoPilot was designed and implemented as a so-called "Wizard-of-Oz" prototype. The term "Wizard-of-Oz" is used to indicate that users were tested on a prototype that works with the assistance of the experimenter working "behind the curtain." This follows the example that Whitten set with Lime, a system that she designed and tested for her dissertation, without implementing in its entirety.

CoPilot is designed to be realized as a plug-in for programs such as Eudora, Outloook, and Outlook Express. Alternatively, it could be implemented as a combination POP and SMTP proxy, in a manner similar to Stream. [3] The specific technique of implementation doesn't matter, as long as CoPilot is able to act as a filter on all incoming and outgoing messages, and as long as CoPilot has a trusted channel through which it can communicate with the user. For the purpose of this study, CoPilot's message engine is implemented as an outgoing message filter that processed messages as they were sent by the experimenter. CoPilot's user interface was implemented as an HTML frame around the message.

CoPilot implements Key Continuity Management using a small set of rules:

- When any message containing a S/MIME certificate is received, that certificate is added to the certificate store. (S/MIME clients like Outlook Express already do this automatically.)
- The first time that CoPilot receives a digitally signed message from a particular email address, that message is flagged with a *yellow* border (Figure 5).
- If subsequent digitally signed messages are received from that address, those messages are flagged with a *green* border (Figure 6). CoPilot tracks how many email messages have been received that were signed with the same certificate

Experimental Subject:		
Campaign Coordinator	ccord@campaign.ex.com	Experimental subjects are told: "You are the Campaign Coordinator."
Campaign Personnel:		
Maria Page	mpage@campaign.ex.com	Campaign Manager and the Coordinator's boss.
Paul Butler	butler@campaign.ex.com	Campaign finance manager.
Ben Donnelly	bend@campaign.ex.com	IT coordinator. Officially Paul's assistant, but also a full-time student at the University of Pennsylvania.
Sarah Carson	carson@campaign.ex.com	"A full-time graphics designer."
Dana McIntyre	dmi@campaign.ex.com	Office manager, but away for the week because her husband is having surgery. (Don't worry, it's a routine procedure.)
Attacker:		
Attacker Paul	butler@campaign.ex.com	Claims to be Paul Butler, having computer problems.
Attacker Sarah	sara_carson_personal@hotmail.com	Claims to be Sarah Carson, sending email from home using her "personal Hotmail account" because she can't get to her campaign email from home.
Attacker Maria	mpage@campaign.ex.com	Attacker "Maria" sends an unsigned message to the Campaign Coordinator asking that the schedule be sent to both Ben and Sarah.

Table 1: Personas used in the Johnny2 experiment.

and displays this information as well.

- If a subsequent digitally signed message is received from that address that is signed with a different key, the message is flagged with a *red* border (Figure 7). The user can elect to trust such a key by clicking a button in the user interface, the effect of which is to change the CoPilot message from red to green and to indicate that the Digital ID is now trusted. The user can change his or her mind by clicking the button a second time, making the message red once more.
- If CoPilot receives an unsigned message from an email address for which it usually receives signed messages, the unsigned message is displayed with a *gray* border (Figure 8).
- If CoPilot receives an unsigned message from an email address that it has never prevously seen, the message is displayed with a *white* border. Once the majority of email that is received by a user is signed, this option could be eliminated and all unsigned mail could be displayed with a gray border.

Although it might appear that an unsigned message should be a red condition, it turns out that there are many instances in which legitimate email is sent by agents that do not have posession of the necessary private key. For example, Microsoft's "Outlook Web Access" will validate S/MIME signatures, but has no provisions for signing outgoing messages. Many people read and respond to email using handhelds such as the Treo 650, which can access mailboxes shared with desktops through an IMAP or POP server. Unfortunately, none of the mail clients that run on PalmOS have S/MIME support.

4 User Test Details

User testing was done on a Dell Optiplex GX270 computer with a 2.4GHz Pentium 4 CPU, 1 GB of RAM and a 40 GB hard drive running Windows XP Professional. The display was a 17-inch Dell LCD display set at a resolution of 1280x1024 pixels, although the resolution was lowered to 1024x768 for one user who had problems reading the small text.

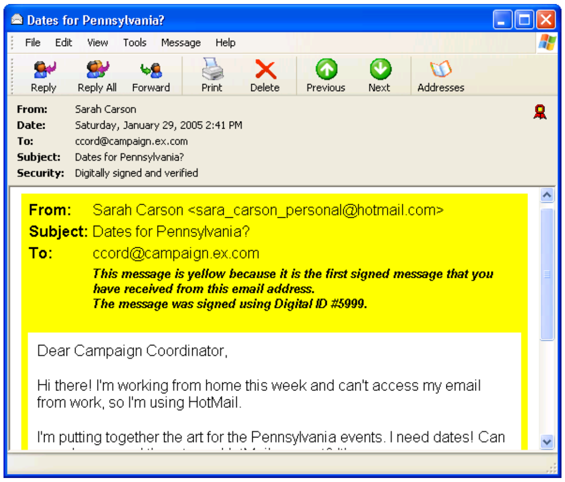
Testing was done with a specially created account named "Campaign Coordinator." The email program was Microsoft Outlook Express 6 SP2. OE6 was preconfigured with a single account named "email" with the email address ccord@campaign.ex.com. POP3 mail was downloaded over SSL.

The email account's Security tab was configured with a certificate called "Campaign Coordinator." The certificate was issued to "Campaign Coordinator" by the "Certification Manager."

As in *Johnny*, each of the five campaign team members were represented by an email account that was accessible to the experimenter. Attacker accounts consisted of actual Hotmail accounts that had been obtained for the purpose of the experiment. All Digital IDs used in the experiment were created with OpenSSL, manually loaded in to the Outlook Express address book and explicitly trusted. Signed messages were sent using a program called the Johnny 2 Workbench that was created specifically for this purpose; the Workbench program also allowed to experimenter to take notes and automatically timestamped those notes every minute. Camtasia by TechSmith was used to record the computer's screen and make an audio transcript.

msg	CoPilot		
#	Color	Sender	Content
1	Yellow	Maria Page	Introductory message introducing Maria and giving the Campaign Coordinator details of the campaign worker's stories. The Coordinator is told to reply. This message provides the subject with information and verifies that they can read and respond to written instructions. This message is also an internal control: Subjects that do not respond to Message #1 within a reasonable amount of time are disqualified and withdrawn from the experiment.
2	Green	Maria Page	The Campaign Schedule and a command telling the Coordinator to send a copy of the schedule to Paul Butler and Dana McIntyre. This message further tests that the subject can respond to a written command from Maria. It also gets the subject into the rhythm of reading an email message and responding by sending out the schedule.
3	Green	Ben Donnelly	Ben asks the Campaign Coordinator for a copy of the schedule. The message is green because Ben's Digital ID was previously installed on the computer.
4	Red	Attacker Paul	Paul says that he is having computer problems and asks the Coordinator to send a copy of the schedule to both Paul's campaign account and his personal Hotmail account, `Paul_J_Butler@Hotmail.com`. This message is digitally signed with a Digital ID that claims to be from `butler@campaign.ex.com` but which is signed by a different Digital ID. This is a *new key attack.* Note: This message has a "Reply-to:" header that causes a reply to be sent to Hotmail. In retrospect the Reply-to header complicated the scenario and should not have been present.
5	Yellow	Attacker Sarah	Attacker Sarah sends email from her Hotmail account `sara_carson_personal@hotmail.com` saying that she is working at home and asking that the schedule be sent to the personal account. This message is digitally signed with a valid Digital ID — it is simply an email address and ID that the subject has not previously seen, making this a *new identity attack.*
6	Gray	Attacker Maria	If the subject does not succumb to both message #4 and message #5, then message #6 is sent. This message is an unsigned message that purports to come from Maria Page, the Campaign Coordinator's boss. Attacker Maria says that she has tried to call the office but that the phones are not working. Maria says she has been on the phone with both Paul and Sarah and that they both need copies of the schedule; please send them! Now! Do it! This is an *unsigned message attack*.
7	Green	Maria Page	In this message, the real Maria Page asks the Campaign Coordinator to send copies of the schedule to Ben Donnelly and Sarah Carson. Some subjects were confused that Maria sent this message, as they had already sent a copy of the schedule to Ben in response to message #3. (Presumably Maria didn't know that Ben had asked for the schedule.) Participants who fell for Attacker Maria in message #6 were especially confused; they couldn't understand why Maria was now asking them to email the schedule to Sarah's campaign address when she had just asked that the schedule be sent to Sarah's personal Hotmail address. This message was a very useful test message to probe precisely what the subject thought had happened in message #6.
8	Green	Maria Page	Maria thanks the subject for participating in the experiment and tells the subject that it is now time for the "Debriefing Interview." Although it wasn't strictly needed, this message gave the experimenter a gentle and in-scenario way to end the experiment.

Table 2: The Johnny 2 Messages


Dates for Pennsylvania?
File Edit View Tools Message Help
Reply Reply All Forward Print Delete Previous Next Addresses

From: Sarah Carson
Date: Saturday, January 29, 2005 2:41 PM
To: ccord@campaign.ex.com
Subject: Dates for Pennsylvania?
Security: Digitally signed and verified

From: Sarah Carson <sara_carson_personal@hotmail.com>
Subject: Dates for Pennsylvania?
To: ccord@campaign.ex.com

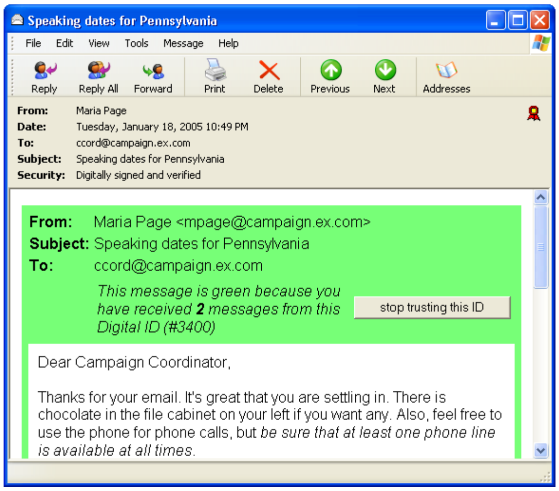
This message is yellow because it is the first signed message that you have received from this email address.
The message was signed using Digital ID #5999.

Dear Campaign Coordinator,

Hi there! I'm working from home this week and can't access my email from work, so I'm using HotMail.

I'm putting together the art for the Pennsylvania events. I need dates! Can

Figure 5: CoPilot Yellow: First time a new identity is seen CoPilot displays a yellow border. (Message #5 is displayed.)


Speaking dates for Pennsylvania
File Edit View Tools Message Help
Reply Reply All Forward Print Delete Previous Next Addresses
stop trusting this ID

From: Maria Page
Date: Tuesday, January 18, 2005 10:49 PM
To: ccord@campaign.ex.com
Subject: Speaking dates for Pennsylvania
Security: Digitally signed and verified

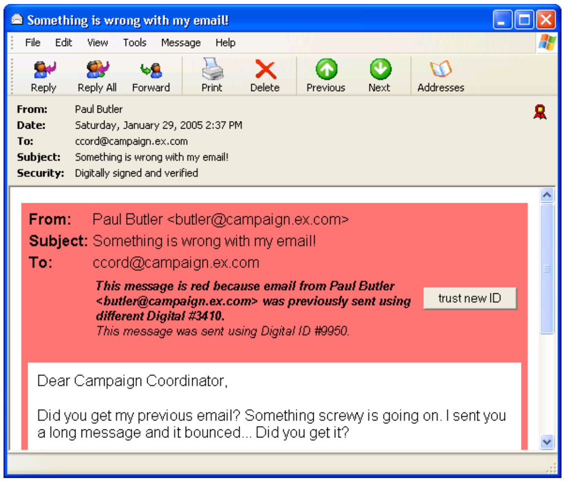
From: Maria Page <mpage@campaign.ex.com>
Subject: Speaking dates for Pennsylvania
To: ccord@campaign.ex.com

*This message is green because you have received **2** messages from this Digital ID (#3400)*

Dear Campaign Coordinator,

Thanks for your email. It's great that you are settling in. There is chocolate in the file cabinet on your left if you want any. Also, feel free to use the phone for phone calls, but *be sure that at least one phone line is available at all times.*

Figure 6: CoPilot Green: Each successive time that a specific Digital ID is seen paired with specific email address, CoPilot displays a green border. (Message #2 is displayed.)


Something is wrong with my email!
File Edit View Tools Message Help
Reply Reply All Forward Print Delete Previous Next Addresses
trust new ID

From: Paul Butler
Date: Saturday, January 29, 2005 2:37 PM
To: ccord@campaign.ex.com
Subject: Something is wrong with my email!
Security: Digitally signed and verified

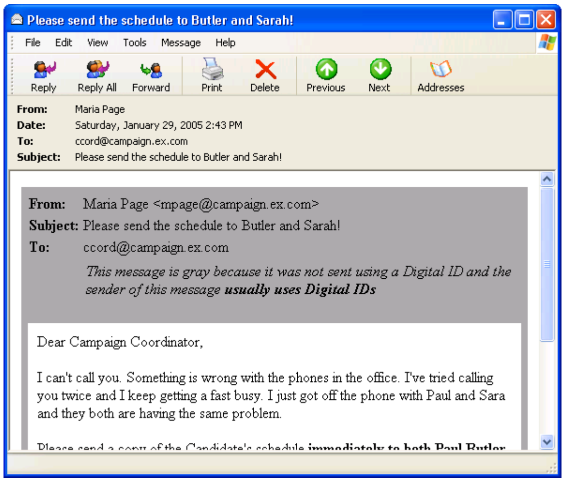
From: Paul Butler <butler@campaign.ex.com>
Subject: Something is wrong with my email!
To: ccord@campaign.ex.com

This message is red because email from Paul Butler <butler@campaign.ex.com> was previously sent using different Digital #3410.
This message was sent using Digital ID #9950.

Dear Campaign Coordinator,

Did you get my previous email? Something screwy is going on. I sent you a long message and it bounced... Did you get it?

Figure 7: CoPilot Red: When the Digital ID associated with an email address changes, CoPilot alerts the user with a red border. (Message #4 is displayed.)


Please send the schedule to Butler and Sarah!
File Edit View Tools Message Help
Reply Reply All Forward Print Delete Previous Next Addresses

From: Maria Page
Date: Saturday, January 29, 2005 2:43 PM
To: ccord@campaign.ex.com
Subject: Please send the schedule to Butler and Sarah!

From: Maria Page <mpage@campaign.ex.com>
Subject: Please send the schedule to Butler and Sarah!
To: ccord@campaign.ex.com

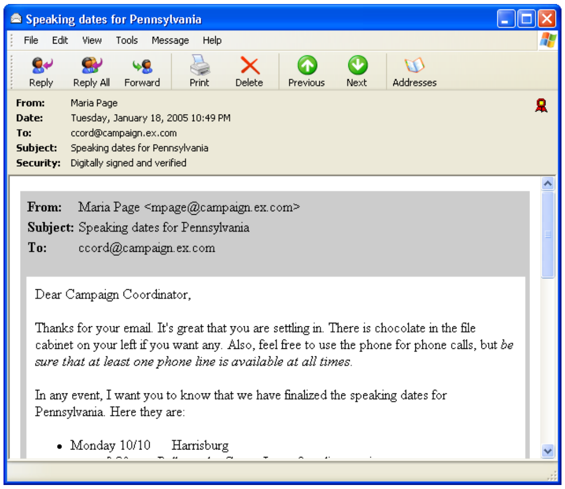
*This message is gray because it was not sent using a Digital ID and the sender of this message **usually uses Digital IDs***

Dear Campaign Coordinator,

I can't call you. Something is wrong with the phones in the office. I've tried calling you twice and I keep getting a fast busy. I just got off the phone with Paul and Sara and they both are having the same problem.

Please send a copy of the Candidate's schedule **immediately to both Paul Butler**

Figure 8: CoPilot Gray: When a message arrives that is not signed from an email address that normally uses signatures, CoPilot displays the message with a gray border. (Message #6 is displayed.)


Speaking dates for Pennsylvania
File Edit View Tools Message Help
Reply
Reply All
Forward
Print
Delete
Previous
Next
Addresses

From: Maria Page
Date: Tuesday, January 18, 2005 10:49 PM
To: ccord@campaign.ex.com
Subject: Speaking dates for Pennsylvania
Security: Digitally signed and verified

From: Maria Page <mpage@campaign.ex.com>
Subject: Speaking dates for Pennsylvania
To: ccord@campaign.ex.com

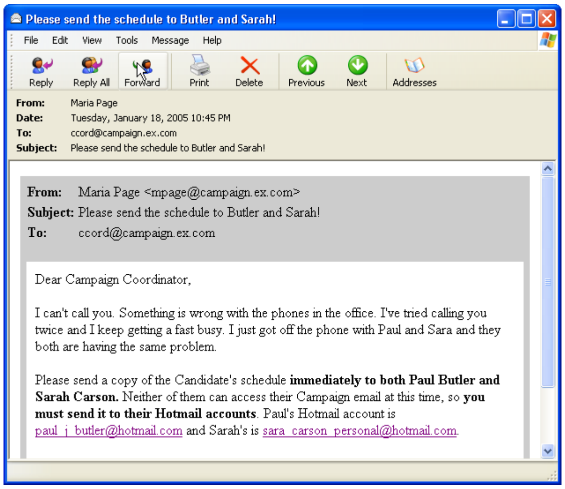
Dear Campaign Coordinator,

Thanks for your email. It's great that you are settling in. There is chocolate in the file cabinet on your left if you want any. Also, feel free to use the phone for phone calls, but *be sure that at least one phone line is available at all times*.

In any event, I want you to know that we have finalized the speaking dates for Pennsylvania. Here they are:

- Monday 10/10 Harrisburg

Figure 9: CoPilot Off: Message #2 displayed to a member of the **NoColor** cohort with CoPilot turned off. The message is still digitally signed and CoPilot is displaying larger versions of th From:, Subject:, and To: fields, but the CoPilot's information about the Digital ID has been suppressed. (Message #2 displayed.)


Please send the schedule to Butler and Sarah!
File Edit View Tools Message Help
Reply
Reply All
Forward
Print
Delete
Previous
Next
Addresses

From: Maria Page
Date: Tuesday, January 18, 2005 10:45 PM
To: ccord@campaign.ex.com
Subject: Please send the schedule to Butler and Sarah!

From: Maria Page <mpage@campaign.ex.com>
Subject: Please send the schedule to Butler and Sarah!
To: ccord@campaign.ex.com

Dear Campaign Coordinator,

I can't call you. Something is wrong with the phones in the office. I've tried calling you twice and I keep getting a fast busy. I just got off the phone with Paul and Sara and they both are having the same problem.

Please send a copy of the Candidate's schedule **immediately to both Paul Butler and Sarah Carson.** Neither of them can access their Campaign email at this time, so **you must send it to their Hotmail accounts**. Paul's Hotmail account is paul_j_butler@hotmail.com and Sarah's is sara_carson_personal@hotmail.com.

Figure 10: CoPilot Off: When members of the **NoColor** cohort received unsigned message #6, the only indication that the message was not signed was the fact that the tiny little certificate icon was missing from the window.

4.1 Methodology

Test subjects were recruited using posters on the MIT campus and messages sent to a variety of non-technical mailing lists. Approximately 85 people responded to the advertisement. These people were sent by email a brief "Intake Questionnaire." Of those responding to the questionnaire, 28 were disqualified because they were familiar with public key cryptography, PGP, or S/MIME.

In the end, we tested a total of 43 subjects for this paper. Subjects ranged in age from 18 to 63 ($\overline{x} = 33; \sigma = 14.2$) The participants had all attended at least some college; 21 were either graduate students or had already earned an advanced degree. Professions were diverse, including PhD candidates in engineering and biology, administrative assistants, clerks, and even a specialist in import/export. Two of the subjects (S12 and S19) appeared to have significant difficulty understanding the English messages in the test, although they were nevertheless able to complete the experiment. We neglected to check our subjects for color-blindness, but none of our subjects mentioned during or after the test that they had any problems distinguishing the colors.

Subjects were told that they would play the role of the Campaign Coordinator, that the Campaign had decided to use special security software to protect email from unauthorized disclosure or modification, and that members of the opposing campaign might attempt to steal the information. Subjects were given a brief tutorial on the use of Outlook Express.

After the OE6 tutorial, subjects were presented with a one-page briefing labeled "Initial Task Description" which gave them the names of the Campaign Manager, the other members of the campaign, and all campaign members' email addresses. Subjects were invited to read this page; it was also read to them by the experimenter.

The briefing included a single sentence about Digital IDs that was typeset in bold:

> **NOTE: Digital IDs for Paul, Ben, Sarah and Dana have been pre-loaded onto your machine by the IT Coordinator.**

4.2 NoColor, Color, and Color+Briefing

In order to test the effectiveness of CoPilot's notices, subjects were randomly divided into one of three groups: **NoColor**, **Color**, and **Color+Briefing**.

- Subjects in the **NoColor** group were presented with an interface that had CoPilot's Key Continuity Management system disabled and all messages were surrounded with a gray border. There were 14 subjects in the **NoColor** group.
- Subjects in the **Color** group were presented with CoPilot's standard multi-color interface, as discussed in Section 3.3. There were 14 subjects in the **Color** group.
- Subjects in the **Color+Briefing** group were presented with CoPilot's standard interface and given a briefing (Figure 11) describing what a Digital ID is and what the different CoPilot colors might

Digital IDs allow Outlook Express to authenticate the sender of email messages.

A Yellow Border will appear around an email message the first time a particular Digital ID is used with an email address.

A Green Border will appear around an email message each successive time that a particular Digital ID is used with an email address.

A Red Border will appear around an email message if the Digital ID used with that email address changes. This might indicate that the sender has moved to a different computer, or that someone else is trying to impersonate the sender.

A Gray Border indicates that no Digital ID was used to send the message. The sender might have forgotten or have a computer problem. Alternatively, the message might be sent by someone else who is trying to impersonate the sender.

Figure 11: The briefing received by the subjects in the **Color+Briefing** group. Each box was typeset with the background of the box being the color that the box purported to describe. On average, the briefing took 50 seconds to read out loud to the subjects.

> mean. This briefing was included on the "Initial Task Description" document that the subjects received and additionally read to the subjects by the experimenter. There were 15 subjects in the **Color+Briefing** group.

It is important to note that the only difference between these three groups was the activity of the CoPilot program and the presence (or absence) of the written briefing. All three groups received the same digitally signed (or unsigned) messages. All three groups were free to use the security features in Outlook Express to learn more about the Digital IDs that were used to sign the messages.

At the conclusion of the briefing subjects were reminded to "think out loud" while they participated in the experiment.

The experiment began when the first email message was sent. During the experiment new email messages were sent when it was clear that the subjects had finished responding to the current email message that they were on, or when roughly 10 minutes had passed since the sending of the previous email message. Questions that the subjects asked regarding non-security features of Outlook Express (for example, how to forward a message) were answered, but any question regarding the operation of an Outlook Express S/MIME feature, the Outlook Express address book, or the CoPilot interface were answered "I don't know." Subjects that asked for additional information regarding the briefing were referred back to the briefing and, in some cases, had the briefing read to them a second time.

Subjects who were quiet for extended periods of time were reminded "don't forget to think out loud."

At the conclusion of the experiment, subjects were given a "Debriefing Questionnaire" and asked additional questions by the experimenter to clarify their understanding and the motivation of the actions that they had taken.

5 Results and Discussion

A total of 43 subjects were run, with 15 in the **Color+Briefing** group and 14 in the other two. Runs averaged 40 minutes in time, with the shortest run lasting 17 minutes and the longest lasting 53. This section summarizes the most significant results observed from the subjects. When reported in tables, χ^2 values were calculated using a logistic regression.

5.1 Task Comprehension

Overall, our subjects clearly comprehended both the task in which they were asked to participate, and the tools that they were given for performing that task. No users were confused by the fact that they were being sent messages that were digitally signed.

In follow-up interviews it was clear that users generally understood that signing a message allowed a recipient to verify who had sent the message and that "encrypting" (or sealing) the message prevented "the wrong people" from viewing the message's contents. Several of the users who received the unsigned message attack from Attacker Maria asked her to resend the message signed with her Digital ID so that they could verify that the message really did come from her. Most of were not sure if they were being attacked or not, but they felt that they could rely on the Digital ID to tell them if the Maria Page who sent message #6 was the same Maria Page who had sent the initial campaign email messages.

Interestingly, these same users were generally unaware that signing a message also provided integrity guarantees. In our experience, most email users are not aware of the fact that a message can be intentionally and maliciously modified as it moves through a computer network or waits for delivery on a mail server. Although we did not specifically ask our users if they realized this possibility, only one (S39) of the users in the study raised this possibility in either the "thinking out loud" or in the follow-up interviews. That user was so paralyzed by the notion that a malicious attacker might be modifying the email messages she was receiving that she was unable to complete the majority of the experiment.

		% subjects resisting attacks		Clicked "encrypt" to seal email	
Cohort	n	sometimes	always	sometimes	always
NoColor	14	43%	0%	50%	21%
Color	14	50%	29%	36%	36%
Color+Briefing	15	87%	33%	20%	13%
χ^2		6.13	3.61	2.96	0.29
$p =$		0.013	0.57	0.087	0.59

Table 3: Summary Results of Johnny 2 User Study

Many users, especially those in the **NoColor** group, struggled for some way to verify the authenticity of the attack messages. Some settled on a form of Email Based Identification and Authentication[2]: they sent an email message to the attacker's apparent campaign address to see if the attacker could read and reply to such messages. Unfortunately, this approach was the undoing of serval subjects, who succumbed to the unsigned message attack from Attacker Maria because the message appeared to have been written in response to a message that the subject had just written.

5.2 Evaluating KCM

Table 3 shows overall attack rates for each cohort. We found that CoPilot's KCM interface significantly ($p < 0.013$) enabled users in the **Color** and **Color+Briefing** groups to resist some attacks. Table 4 breaks down success rates for individual attacks, showing that the **Color** and **Color+Briefing** groups did significantly ($p < .05$) better with the "new key attack" and the "unsigned message attack." But the interface did not inoculate subjects against the "new identity attack."

5.2.1 KCM Against the New Key Attack

KCM worked significantly ($p = 0.001$) better against the new key attack than no KCM—especially when the subjects were briefed that a new key might indicate that "someone else is trying to impersonate the sender" (Figure 11). The improvement was dramatic for the **Color+Briefing** group. One explanation is that these users were specifically briefed that two likely conditions that might result in a red message: "that the sender has moved to a different computer, or that someone else is trying to impersonate the sender."

5.2.2 KCM Against the New Identity Attack

KCM did not work significantly ($p = 0.31$) better than no KCM against the new identity attack. It can be argued that this is because the subjects were not primed that a yellow border could be an attack. We do not think that this criticism is warranted, however, because many subjects verbally debated whether or not the yellow message was in fact from the real Sarah who was in the campaign or from some other Sarah. Many rationalized that the key and the email address were different because Sarah was using her home computer—the justification present in message #5. Our subjects knew that they *might* be under attack: they simply decided to trust Attacker Sarah.

Only two subjects noticed that Attacker Sarah's Hotmail address had a misspelling in the name. S27 discovered the inconsistency before sending the message to Attacker Sarah but decided to send the message anyway; S33 used the misspelling to help confirm the decision not to trust a yellow message.

5.2.3 KCM Against the Unsigned Message Attack

KCM was more successful against the unsigned message attack, conveying statistically significant ($p = 0.046$) immunity from spoofing to those in the **Color** and **Color+Briefing** cohorts.

We were surprised that the unsigned message attack wasn't more successful against users in the **NoColor** group. During the follow-up interview, we were told that what frequently protected subjects from following Attacker Maria's instructions was not the fact that message #6 was not signed: the indications in Outlook Express 6 that messages are signed are very subtle, and as a result not a single user in the **NoColor** group realized that message #6 was not signed while the other messages were signed.

Instead, what seemed to protect users in the **NoColor** cohort from responding to message #6 was that fact that Attacker Maria was asking them to send the secret campaign schedule to a Hotmail address: many subjects said that they simply did not trust Hotmail's security.

5.3 Evaluating the CoPilot Interface

CoPilot's HTML-based interface was designed to look like the program had been tightly integrated with Outlook Express. As it turns out, the integration was a little too transparent. Although in the debriefing interview every subject in the **Color** and **Color+Briefing** group said that they saw the colored borders, we observed that users in the **Color** group frequently did not read the text that CoPilot displayed underneath the "To:" header (See Figures 5 through 8). CoPilot integrated so well that users were ignoring it!

The "Trust this ID" button was never explained to the subjects. Only a few subjects experimented with the button to see what it did. Two subjects (S31 and S39) misunderstood: when they saw the green-bordered message with the button labeled "stop trusting this ID," these users thought that the legend on the button was an instruction *from* CoPilot *to them*, telling them that they should *stop trusting this ID*! Both users clicked the but-

	% of subjects that tried to send the schedule when requested by:					
				new	new	unsigned
				key	identity	message
Group	Maria 1	Maria 2	Ben	attack	attack	attack
NoColor	100%	92%	100%	71%	79%	75%
	(14/14)	(11/12)	(14/14)	(10/14)	(11/14)	(9/11)
Color	93%	100%	92%	64%	50%	58%
	(13/14)	(13/13)	(11/12)	(9/14)	(7/14)	(7/12)
Color+Briefing	100%	100%	100%	13%	60%	43%
	(13/15)	(14/15)	(13/14)	(2/15)	(9/15)	(6/14)
χ^2	2.20	0.018	0.79	10.61	1.02	3.98
	$p = 0.14$	$p = 0.89$	$p = 0.37$	$p = 0.001$	$p = 0.31$	$p = 0.046$

Table 4: Percentage of subjects that sent email containing the secret campaign schedule in response to commands from Maria and Ben, and in response to the three attacks. Numbers in parenthesis indicate the number of subjects who responded compared to the number who were subjected to the test condition. Subjects who misinterpreted the Maria 1 message and sent email to *all* campaign workers did not feel the need to comply with the Maria 2 or Ben messages because they had already done so; they were omitted from the sample. Because of the way in which the subjects responded to earlier messages in the scenario, not all subjects were exposed to the unsigned message attack.

ton, the CoPilot border changed from green to red, and the users were pleased that they had complied with the computer's instructions and apparently gotten the correct result.

5.4 "Encrypt"

Unprompted by the instructions but given the option by the Outlook Express interface, roughly a third of the users in our study clicked the "encrypt" button to seal the candidate's confidential schedule before it was sent by email.

The OE6 "encrypt" button is a toggle switch. Pressing the button once causes a little blue icon to appear next to the `To:` field in the message composition window. No cryptographic operations happen, though, until the user tries to send the message. At this point OE6 scans the Outlook Express Address Book to see if there is an S/MIME certificate on file that matches each `To:` address. If all of the addresses match, the message is sealed and sent. If one or more of the addresses do not match, a warning appears (Figure 3).

Users who did not have the CoPilot Key Continuity Management interface were significantly ($p = 0.097$) more likely to use encryption than those who had the interface. Interviews with users revealed that many were using the intended recipient's ability to *unseal* a message as a proxy for *recipient authentication*. That is, subjects believed that only members of the campaign team would be able to unseal messages that had been properly sealed. In follow-up interviews, several subjects said the campaign IT coordinator should have configured Outlook Express so that it would *only* send sealed messages if sealing messages was a campaign priority.

However, subjects were mistaken: OE6 was very happy to seal the message for Attacker Sarah, as Attacker Sarah's "yellow" message had been digitally signed and, as a result, her certificate (and, indeed, her Hotmail address) were automatically incorporated into the OE6 address book when the message was viewed. Users didn't understand that a message could be sealed for an attacker: those who were asked said that they thought that something about the CoPilot system would prevent sealed messages being sent to someone who was not affiliated with the campaign. Subjects were also confused by the sudden appearance of Attacker Sarah's Hotmail address in the OE6 Address Book: for some subjects, this provided the added confirmation that they needed to trust Attacker Sarah.

Every subject who discovered the "Encrypt" button and tried to send a message to Attacker Paul was confused when they could not send a sealed to the Hotmail address (Figure 3). They couldn't do this, of course, because Attacker Paul's message was digitally signed with a certificate that had the address `butler@campaign.ex.com`, and not `paul_butler@hotmail.com`. (It is appropriate that Attacker Paul was able to obtain such a certificate because the Campaign is using Key Continuity Management, and not third-party certification.) A handful referred to the online help or did web searches with Google to try to diagnose the problem: all of these individuals determined that the problem was that they did not have a Digital ID on file for Attacker Paul's Hotmail address. Several users attempted to change the email address on Paul's campaign Digital ID to his Hotmail address so that they could send sealed mail to his Hotmail account; others tried in vein to figure out how to "make" a Digital ID for the Hotmail Account. two of the users several sent mail to Attacker Paul telling him that they could not send him the schedule until he got a Digital ID and sent him instructions for obtaining one.

6 Conclusion

Using a modified version of the *Johnny* study developed by Whitten and Tygar, we tested the Key Continuity Management (KCM) proposal. We found that significant increases in security can be achieved with relatively minor enhancements to the way that programs like Outlook Express handle and display digitally signed mail.

We have shown that even though the deployment of KCM could improve security, it is not the panacea to the mail security problem for which we are looking. In particular, KCM provides users with no readily-apparent tools for deciding whether or not to trust new identities that show up with new keys. But this isn't a problem that is created by KCM. With today's PKI-based systems, for example, an attacker can create a new Hotmail address and then get a public key certificate for it certified by VeriSign or Thawte. And this problem is no different from similar problems in the offline world. You receive a letter from an old friend asking a favor: is it really from the same person?

Different kinds of email need and can employ different kinds of certification. [4] Within some organizations a centralized PKI might be appropriate; other organizations that do not have the wherewithal to manage such a deployment might choose to employ KCM—perhaps with some form of additional out-of-band certification, as was suggested by several of our subjects. It isn't hard to see that the current CoPilot interface could be modified to support two kinds of "green" messages: those that simply reflect trust from continued use of a key, and those that have been explicitly certified.

Today S/MIME technology is widely deployed but rarely used. There is much to be learned from exploring the use of this technology in usability experiments, and much to be gained from encouraging its continued deployment using approaches like KCM.

Acknowledgments

The authors would like to express their thanks to both Peter Gutmann and Alma Whitten for answering numerous email messages that we sent to them during the course of this project. Thanks are also due to Microsoft's security team, especially David Ladd and Scott Culp, who have offered insight, suggestions, and answered numerous questions regarding this project at various times over the past year. David Clark, Karen Sollins, and John Wroclawski at MIT's Advanced Network Architecture group provided useful feedback on the design of CoPilot and the underlying KCM model that it implements. Jean Camp, now at Bloomington, provided much of the initial guidance on this and several related projects. Ariel Rideout and Beth Rosenberg both provided useful feedback on this paper. Finally, we would like to thank Jeff Schiller at MIT Information Services, who has been a constant source of useful information and answered questions on the topic of S/MIME history and email security.

Availability

We have made the entire Johnny 2 experimental setup available for download so that other researchers can it as a standardized scenario for testing secure messaging systems. For further information, please visit `http://www.simson.net/johnny2/`.

References

[1] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[2] Simson L. Garfinkel. Email-based identification and authentication: An alternative to pki? *Security & Privacy Magazine*, 1:20–26, Nov. - Dec. 2003.

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

[4] 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. In *Financial Cryptography and Data Security 2005*, 2005.

[5] 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/.

[6] Loren M. Kohnfelder. *Towards a practical public-key cryptosystem*. PhD thesis, MIT, Cambridge, MA, May 1978. Supervised by L. Adleman.

[7] Kevin D. Mitnick and William L. Simon. *The Art of Deception*. John Wiley & Sons, 2002.

[8] B. Ramsdell. RFC 3851: Secure/multipurpose internet mail extensions (S/MIME) version 3.1 message specification, July 2004.

[9] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *1999 AT&T Software Symposium*, pages 172–194, September 15 1999.

[10] Thawte Consulting. Certification practices statement version 2.1, January 09 2004.

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

[12] Alma Whitten and J. D. Tygar. Usability of security: A case study. Technical report, December 1998.

[13] 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.

[14] T. Ylonen. SSH - secure login connections over the internet. In *Proceedings of the 6th Security Symposium) (USENIX Association: Berkeley, CA)*, page 37, 1996.