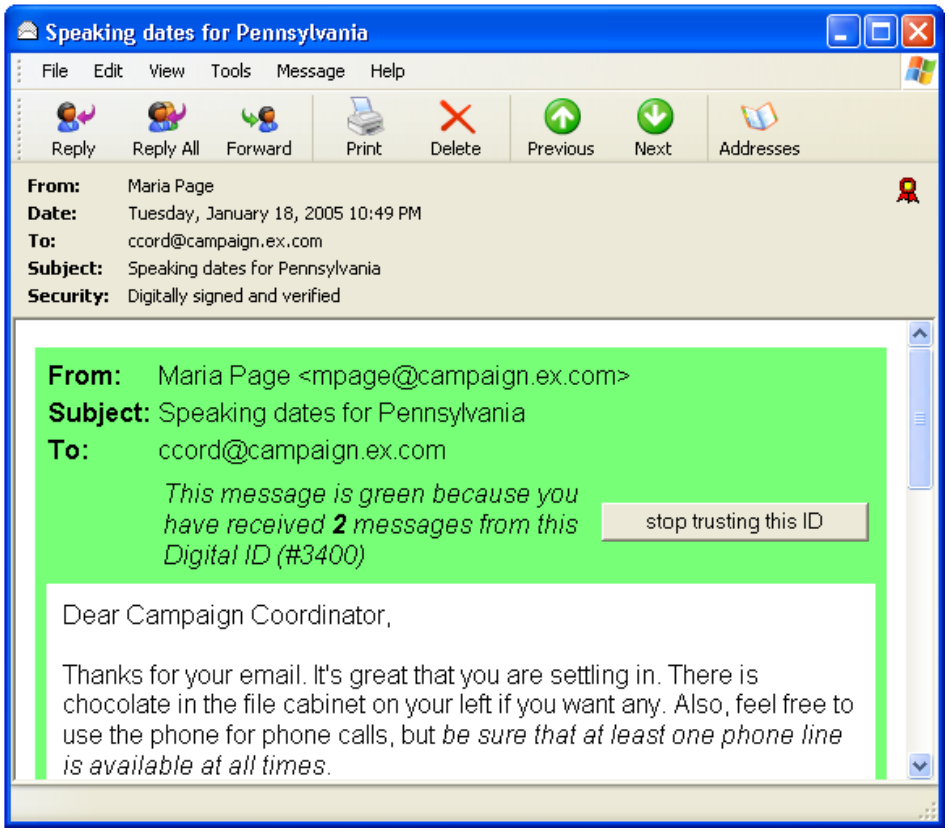
Johnny 2: A study of key continuity management



Simson L. Garfinkel & Robert C. Miller

MIT Computer Science and Artificial Intelligence Laboratory

`http://www.simson.net/johnny2`

Secure Messaging — email that is *signed* and *sealed* — seems to be the grand challenge of usability and security.

Public key cryptography was developed for secure messaging.

- 1976 — Diffie Helman
- 1977 — RSA
- 1987 — RFC 989 (PEM)
- 1991 — PGP Released
- 1996 — S/MIME

Today we use public key cryptography for SSH, SSL, and code signing — but there's virtually no secure email.

People do care about email security. (Garfinkel et al, FC05)

In our study of 400+ Amazon.com merchants:

- 59% thought that email receipts should be digitally signed.
- 47% thought receipts should be sealed

And they have the tools.

- S/MIME in Outlook, Outlook Express, Apple Mail, Thunderbird.
- Remove AOL and webmail,
  and 80%–90% of email users have S/MIME support.

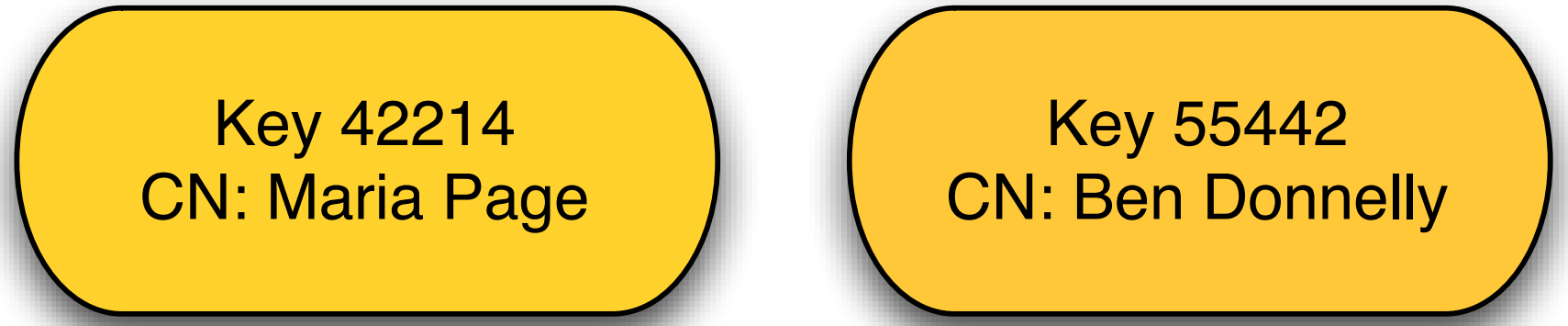
S/MIME automatically validates messages signed with keys that are certified by a well-known CA.

Signed by Thawte

Self-Signed

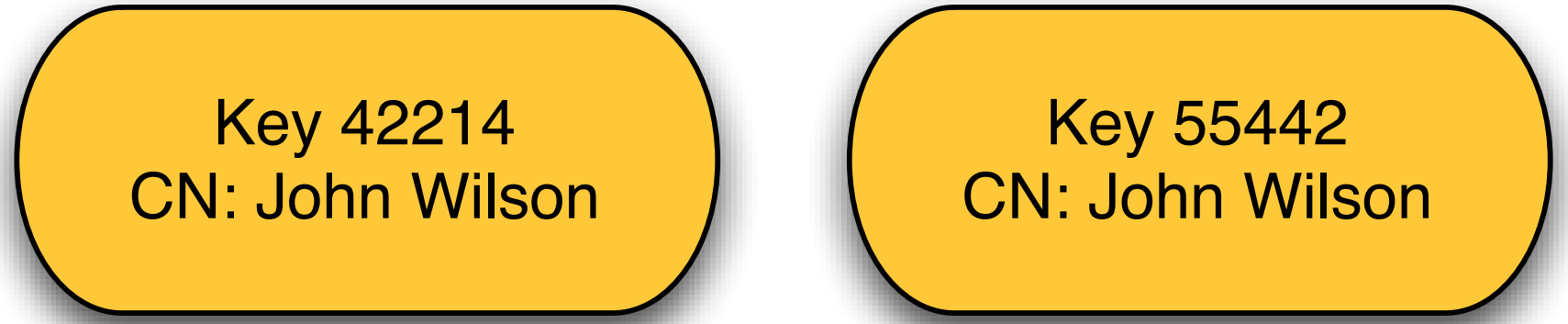
Getting a certificate can be difficult and expensive.

A certificate is a statement signed by a CA that binds a key to a particular Common Name (CN)



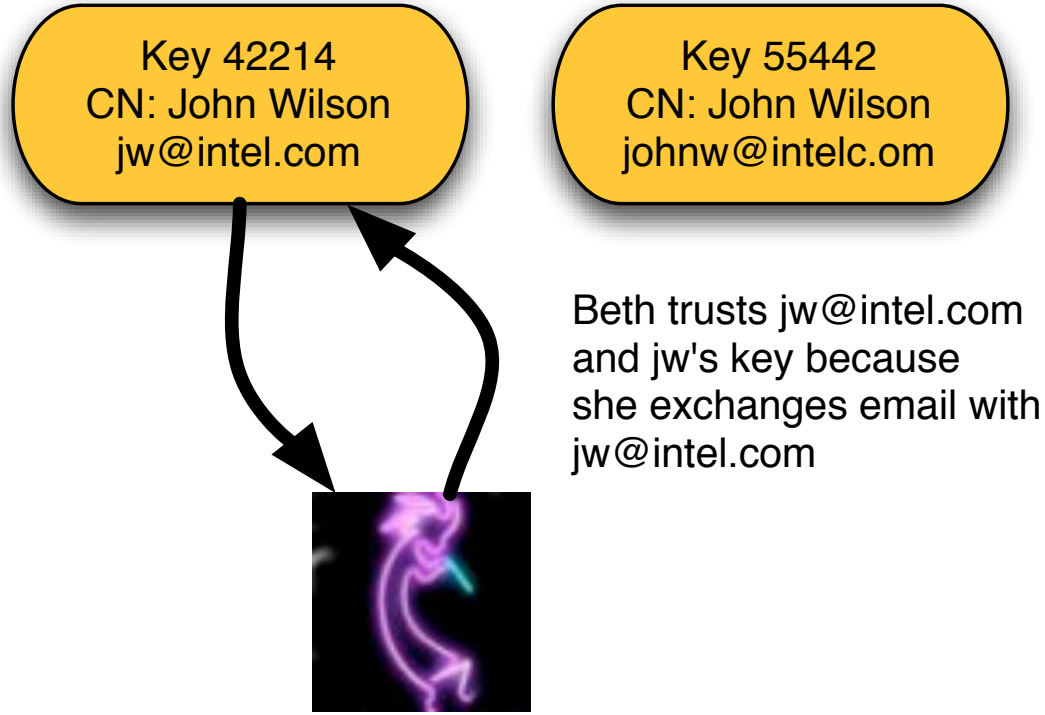
Theory: humans understand names, not public keys.

Ellison argues that certified names are useless because names are not unique, not even within a company:



This is known as the John Wilson Problem. [Ellison 02]

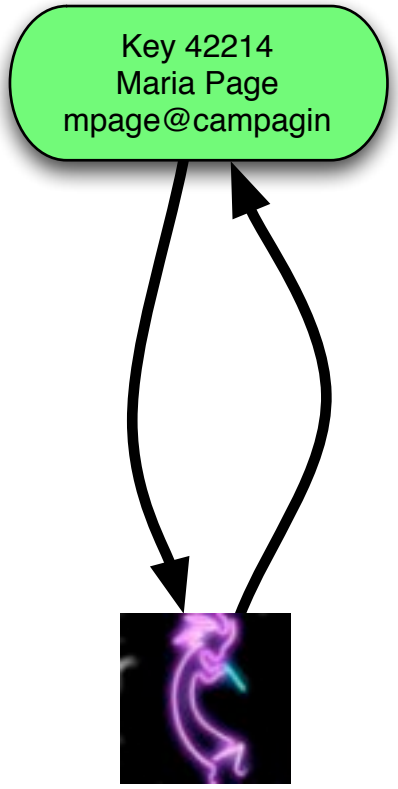
An alternative is to directly certify relationships:



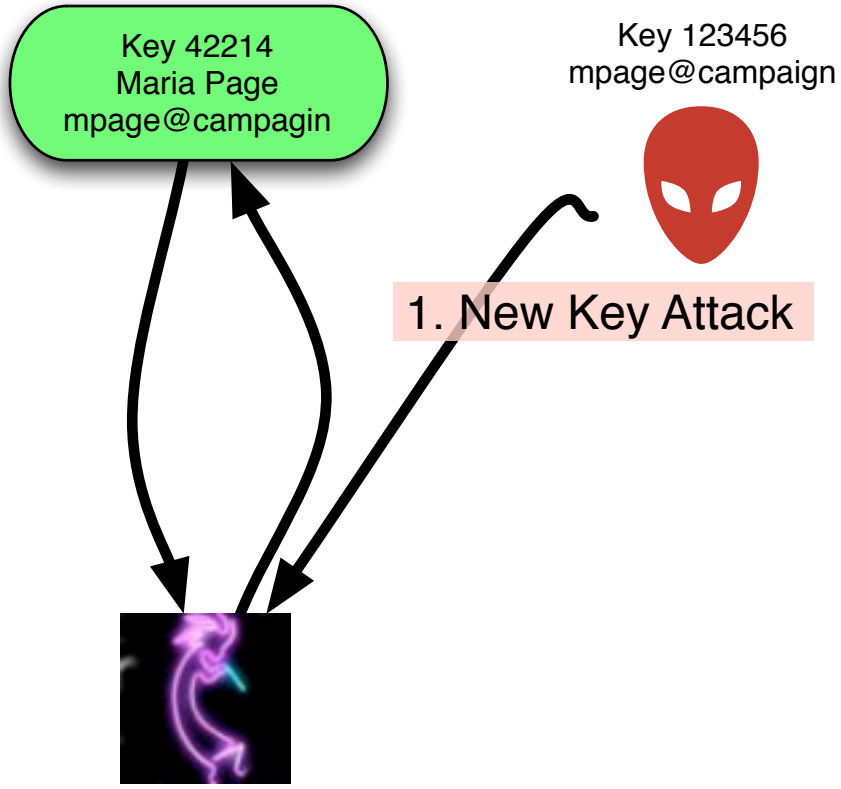
We rarely send confidential information on the first email.

This is called Key Continuity Management.
a.k.a. "the SSH model." [Gutmann 04]

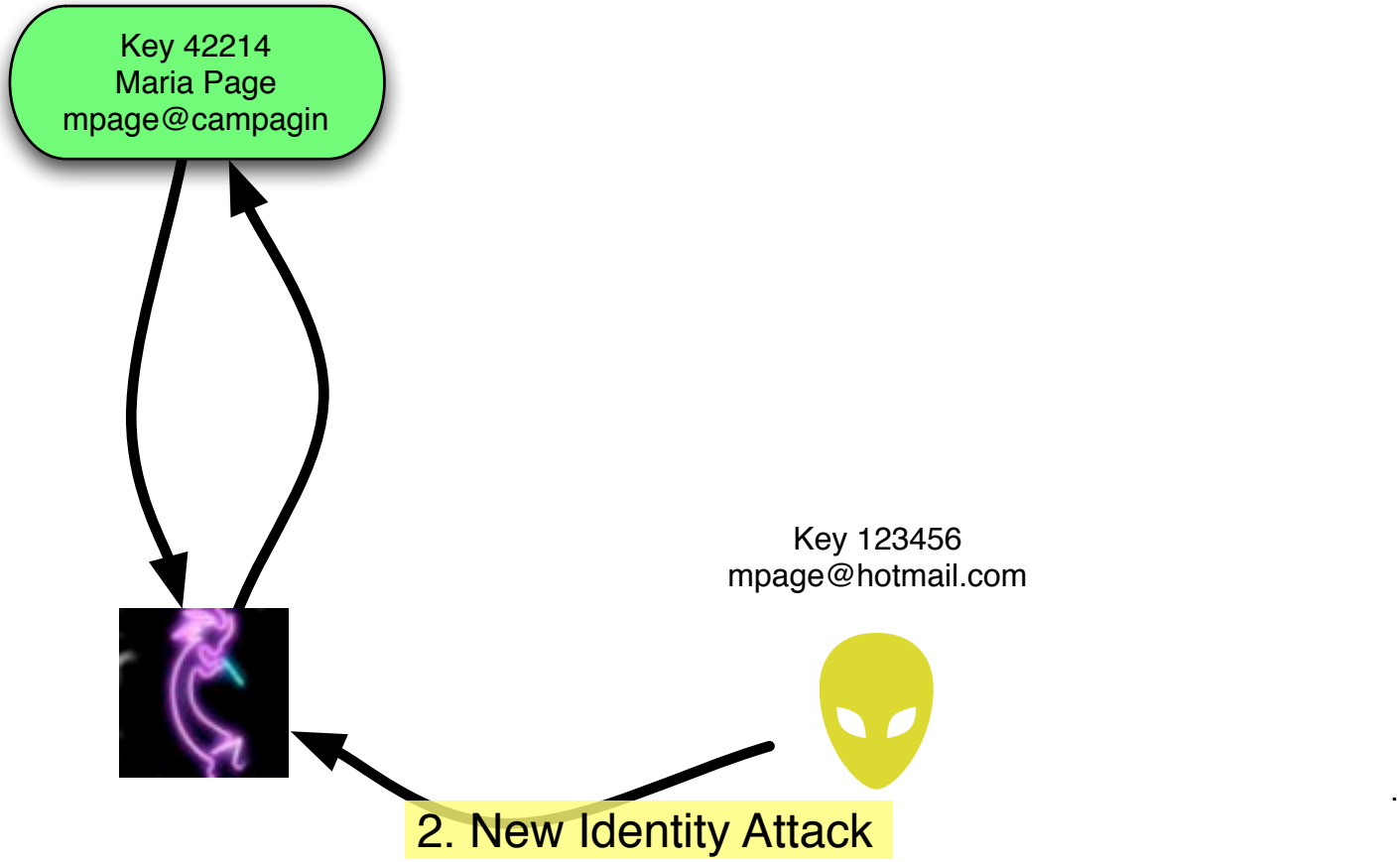
Unfortunately, KCM allows a number of attacks



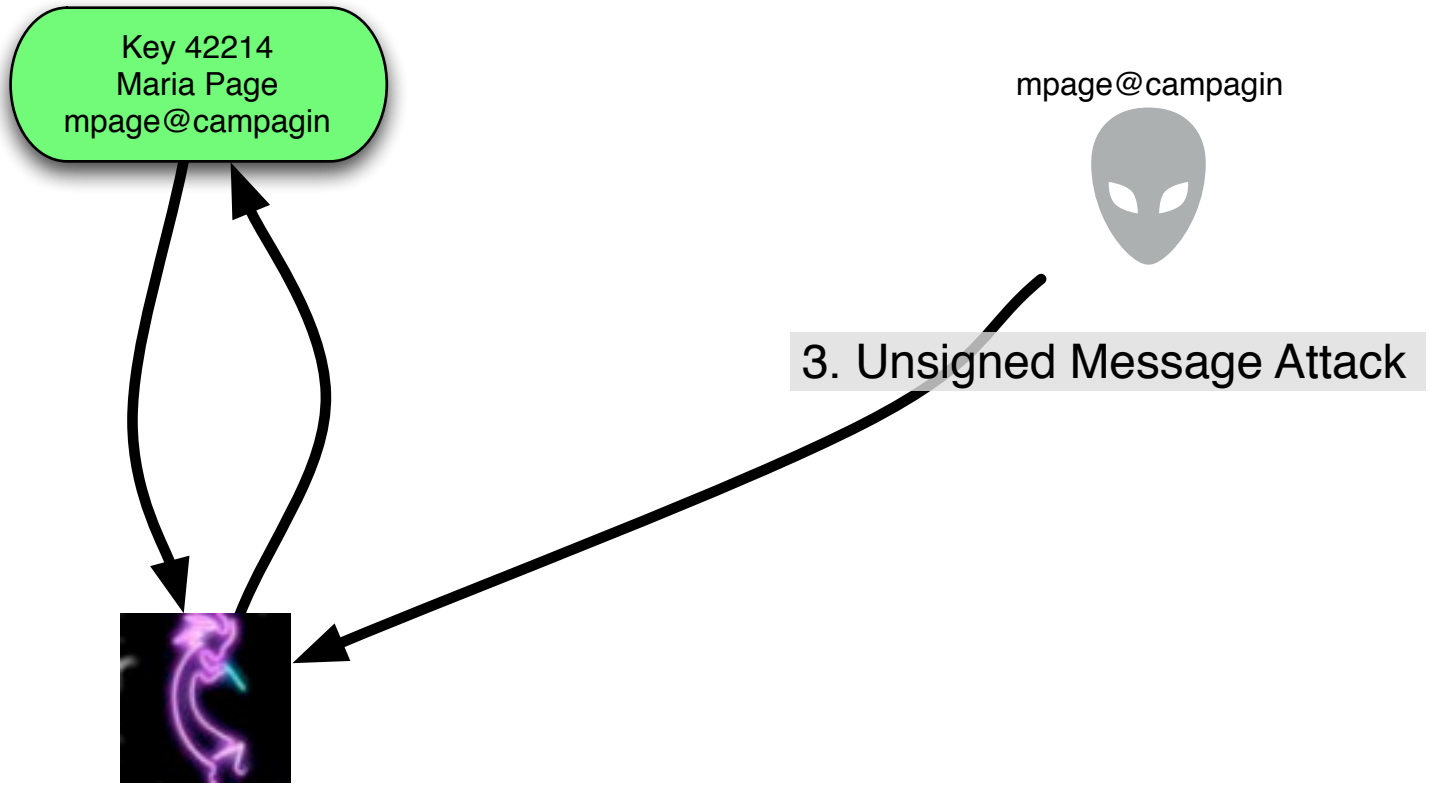
Unfortunately, KCM allows a number of attacks



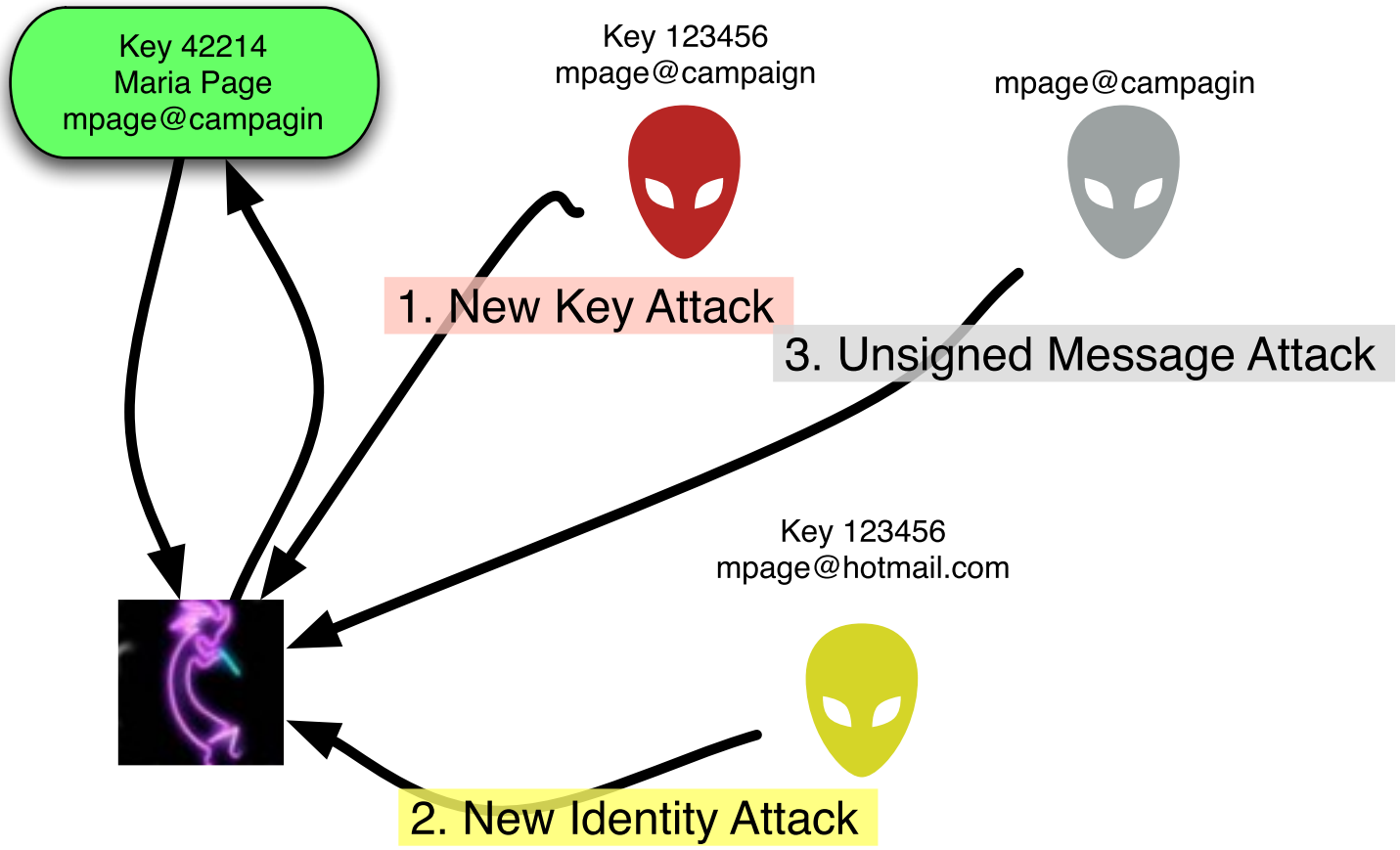
Unfortunately, KCM allows a number of attacks



Unfortunately, KCM allows a number of attacks

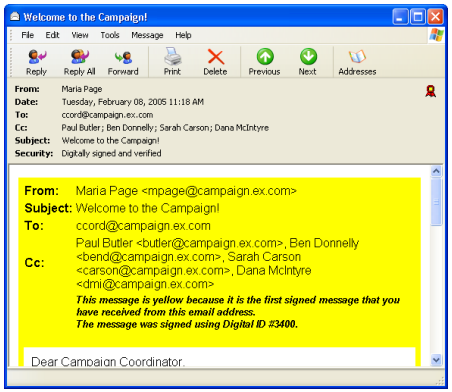


Unfortunately, KCM allows a number of attacks

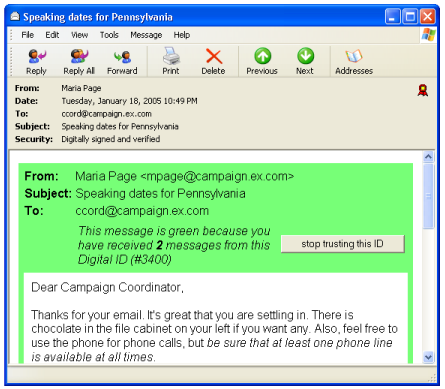


Now we have something we can test: can people resist these attacks?

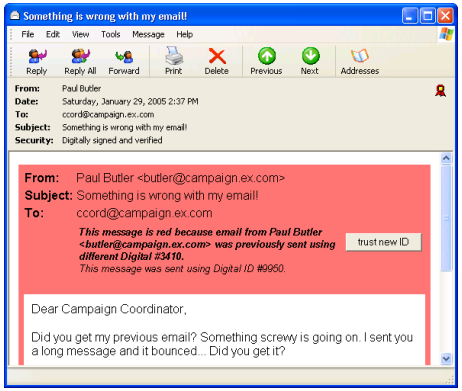
CoPilot Implements a Key Continuation Management interface on top of Outlook Express.



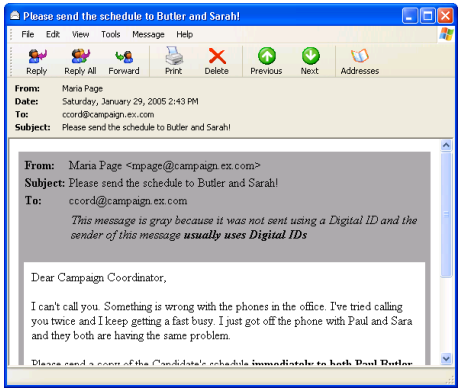
New Key



Same Key



Changed key



No Key

The original plan: Test with Whitten and Tygar's "Why Johnny Can't Encrypt" protocol

- Subject plays the role of a political campaign worker.
- Encryption used to protect email from opposing campaign.
- Use *Johnny* as our control group: see if KCM has a higher success rate and lower spoof rate than PGP.

Why Johnny Can't Encrypt:
A Usability Evaluation of PGP 5.0

Alma Whitten
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu

J. D. Tygar[1]
EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu

Abstract

User errors cause or contribute to most computer security failures, yet user interfaces for security still tend to be clumsy, confusing, or near-nonexistent. Is this simply due to a failure to apply standard user interface design techniques to security? We argue that, on the contrary, effective security requires a different usability standard, and that it will not be achieved through the user interface design techniques appropriate to other types of consumer software.

To test this hypothesis, we performed a case study of a security program which does have a good user interface by general standards: PGP 5.0. Our case study used a cognitive walkthrough analysis together with a laboratory user test to evaluate whether PGP 5.0 can be successfully used by cryptography novices to achieve effective electronic mail security. The analysis found a number of user interface design flaws that may contribute to security failures, and the user test demonstrated that when our test participants were given 90 minutes in which to sign and encrypt a message using PGP 5.0, the majority of them were unable to do so successfully.

We conclude 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. We close with a brief description of our continuing work on the development and application of user interface design principles and techniques for security.

1 Introduction

Security mechanisms are only effective when used correctly. Strong cryptography, provably correct protocols, and bug-free code will not provide security if the people who use the software forget to click on the encrypt button when they need privacy, give up on a communication protocol because they are too confused about which cryptographic keys they need to use, or accidentally configure their access control mechanisms to make their private data world-readable. Problems such as these are already quite serious: at least one researcher [2] has claimed that configuration errors are the probable cause of more than 90% of all computer security failures. Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make security manageable for even untrained users has become critical [4, 9].

This is inescapably a user interface design problem. Legal remedies, increased automation, and user training provide only limited solutions. Individual users may not have the resources to pursue an attacker legally, and may not even realize that an attack took place. Automation may work for securing a communications channel, but not for setting access control policy when a user wants to share some files and not others. Employees can be required to attend training sessions, but home computer users cannot.

Why, then, is there such a lack of good user interface design for security? Are existing general user interface design principles adequate for security? To answer these questions, we must first understand what kind of usability security requires in order to be

[1] Also at Computer Science Department, Carnegie Mellon University (on leave).

The idea of comparing results directly with *Johnny* didn't quite work out.

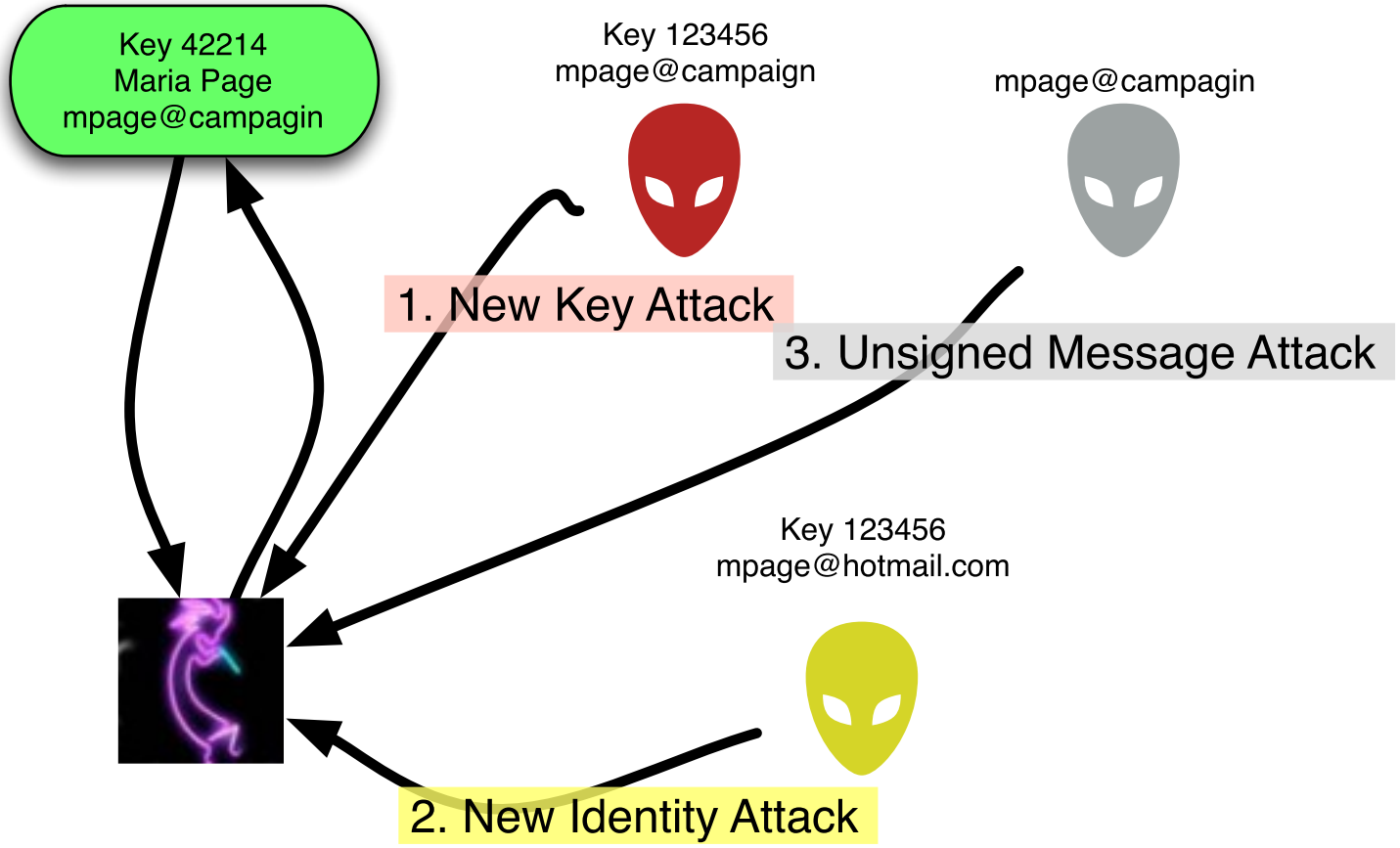
- *Johnny* didn't have an attacker
- *Johnny* didn't use third-party certification
  — it used email answerback certification.
- *Johnny* didn't have a control: the results were qualitative.

The Johnny 2 Scenario:

It's the original *Johnny* scenario, except:

- The personas are developed
- There are good guys and bad guys
- The bad guys are trying to spoof the experimental subject.

Big Questions to answer:



What will the users do when faced with the attacks?

Other questions that we can answer:

- Do users understand difference between signing and sealing?
- If users can trivially sign and/or seal their email, will they?
- If users can seal confidential information before they send it, will they be less concerned about the destination?

The big question we don't need to answer:

Is it just as secure as CA model?

This isn't a fair question. . .

. . . KCM doesn't replace the CA model,
it replaces no crypto at all.

. . . You can *still* use the CA model, if you can find a CA.

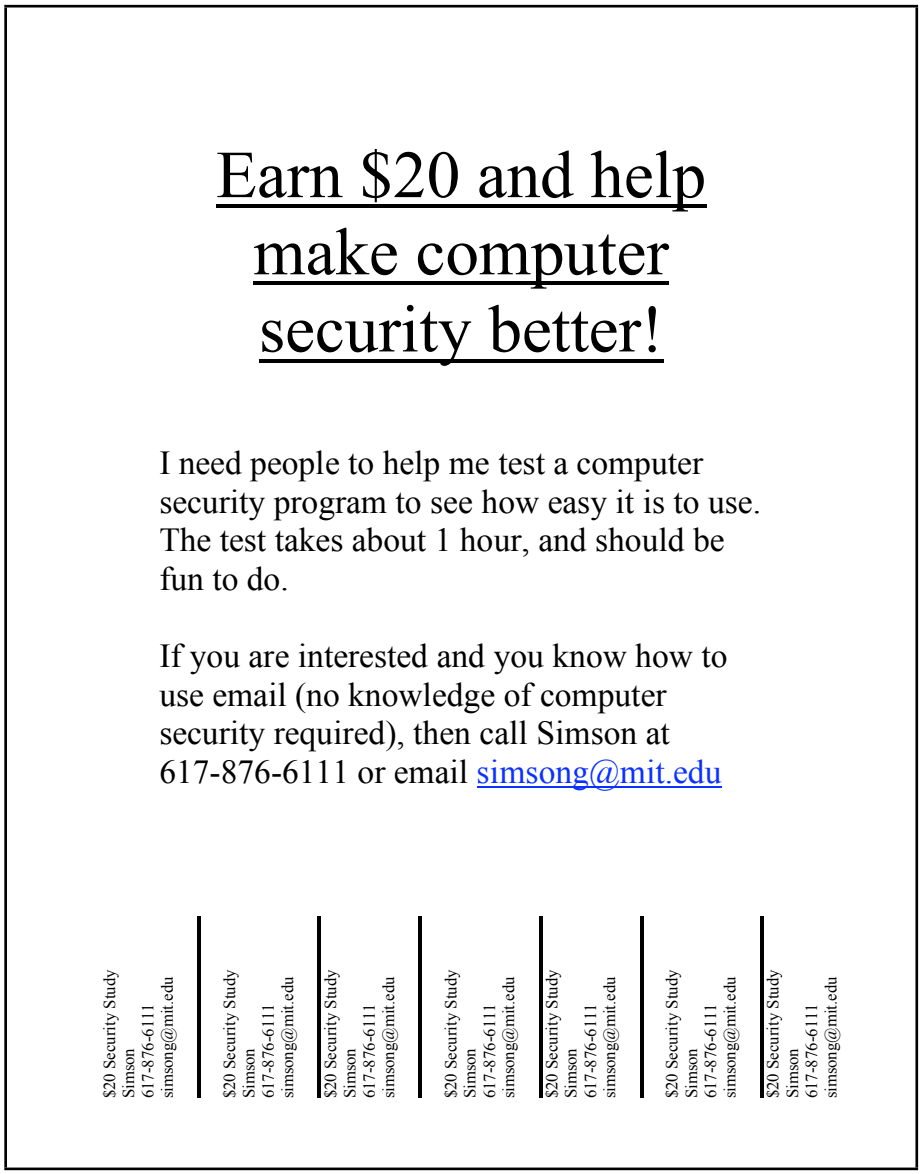
Johnny 2 User Study

Subjects recruited by posters at MIT.

43 subjects aged 18–63 ($\overline{x} = 33, \sigma = 14.2$)

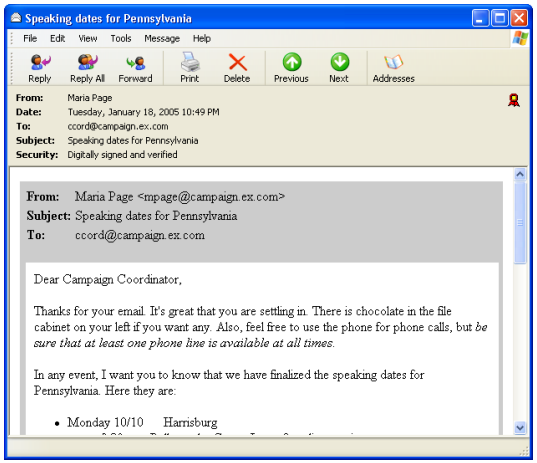
19 Men, 24 Women

17 to 57 minutes ($\overline{t} = 41, \sigma = 10.32$)

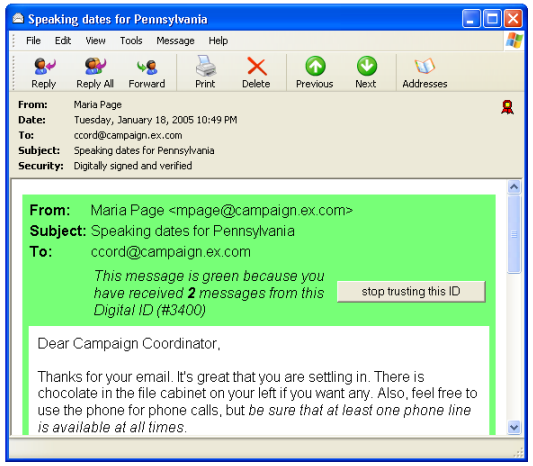


Three Cohorts were compared for statistically-significant differences.

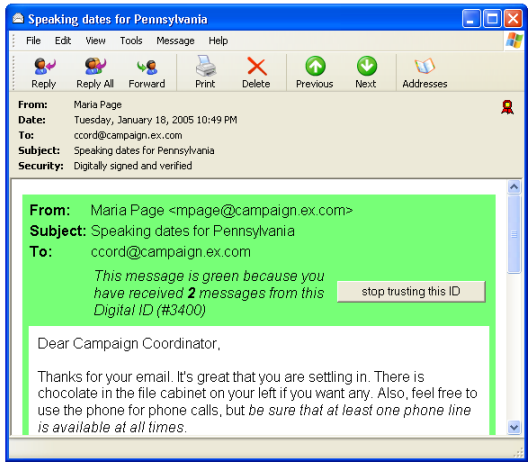
No Color



Color

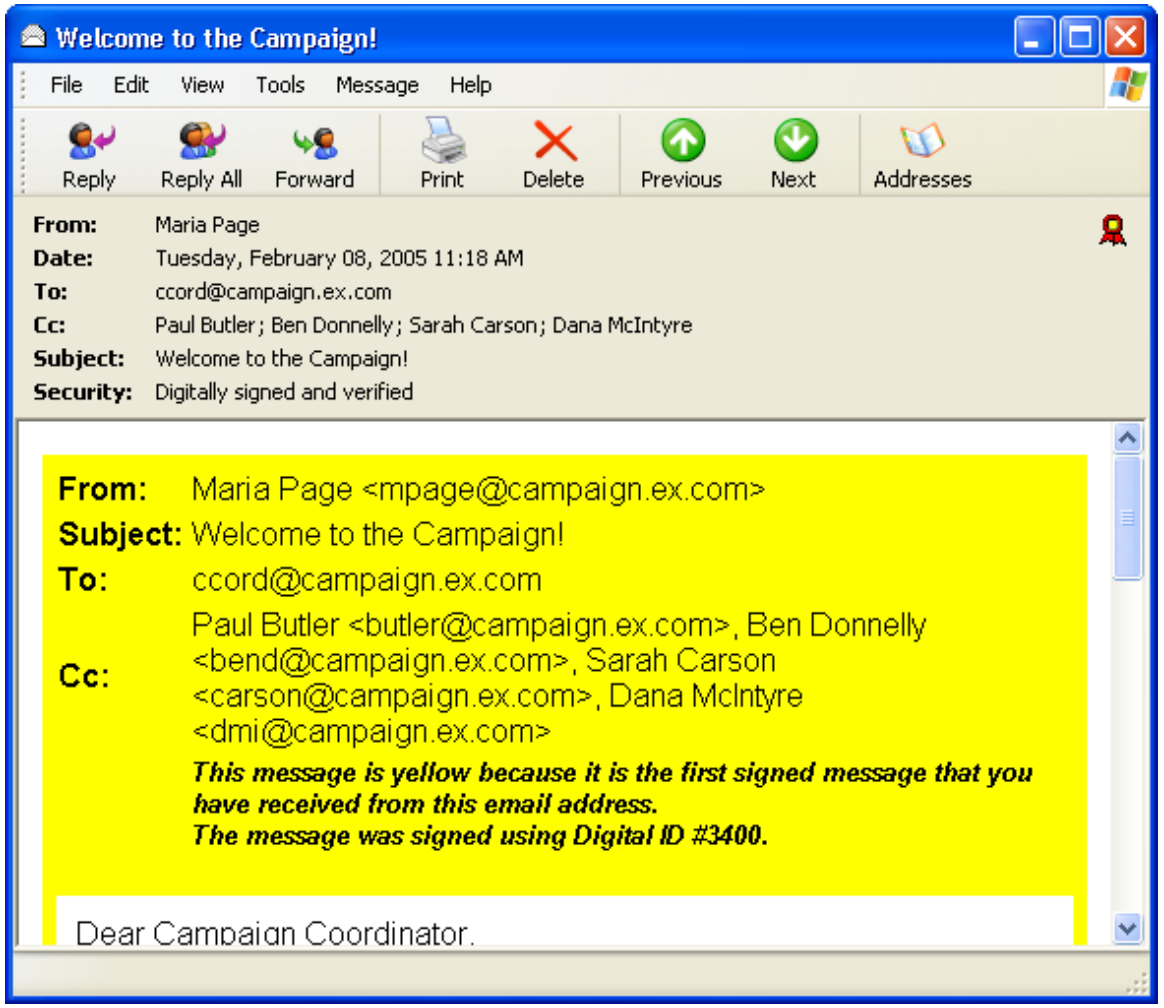


Color + Briefing



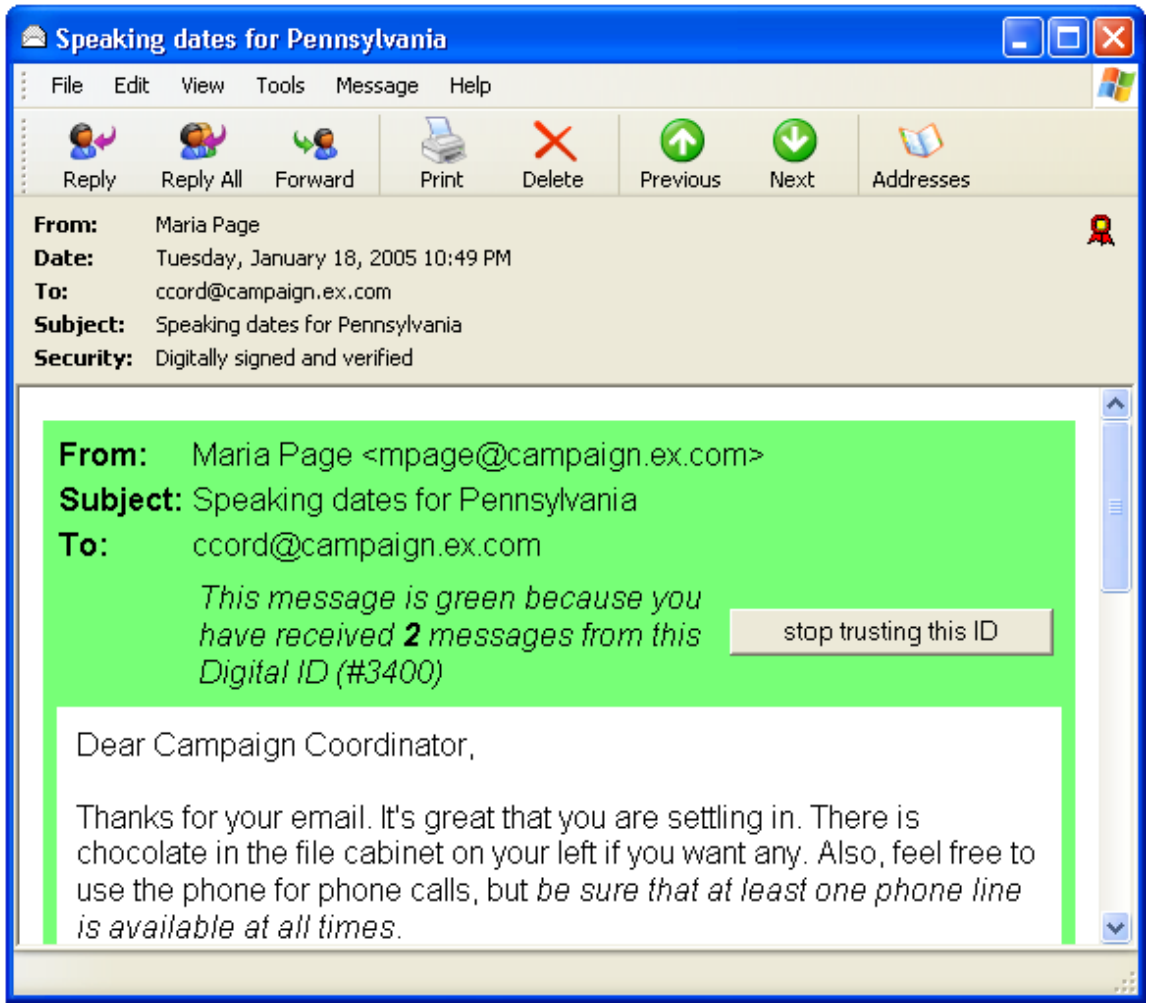
A Green Border will appear around an email message each successive time that a particular Digital ID is used with an email address.

Scenario Message 1:
Greetings from Maria Page



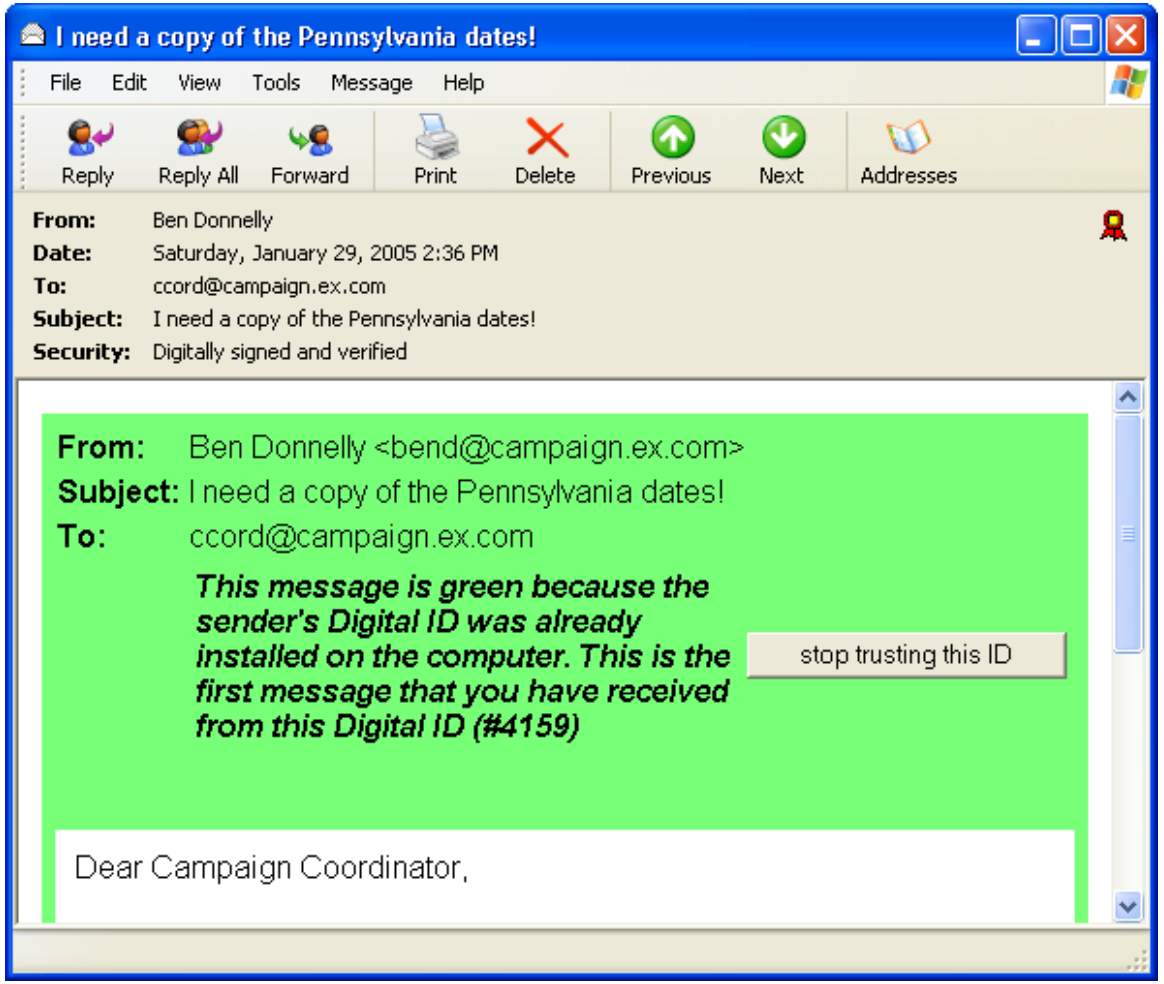
Orients user and provides list of campaign worker roles.

Scenario Message 2:
Maria sends the schedule



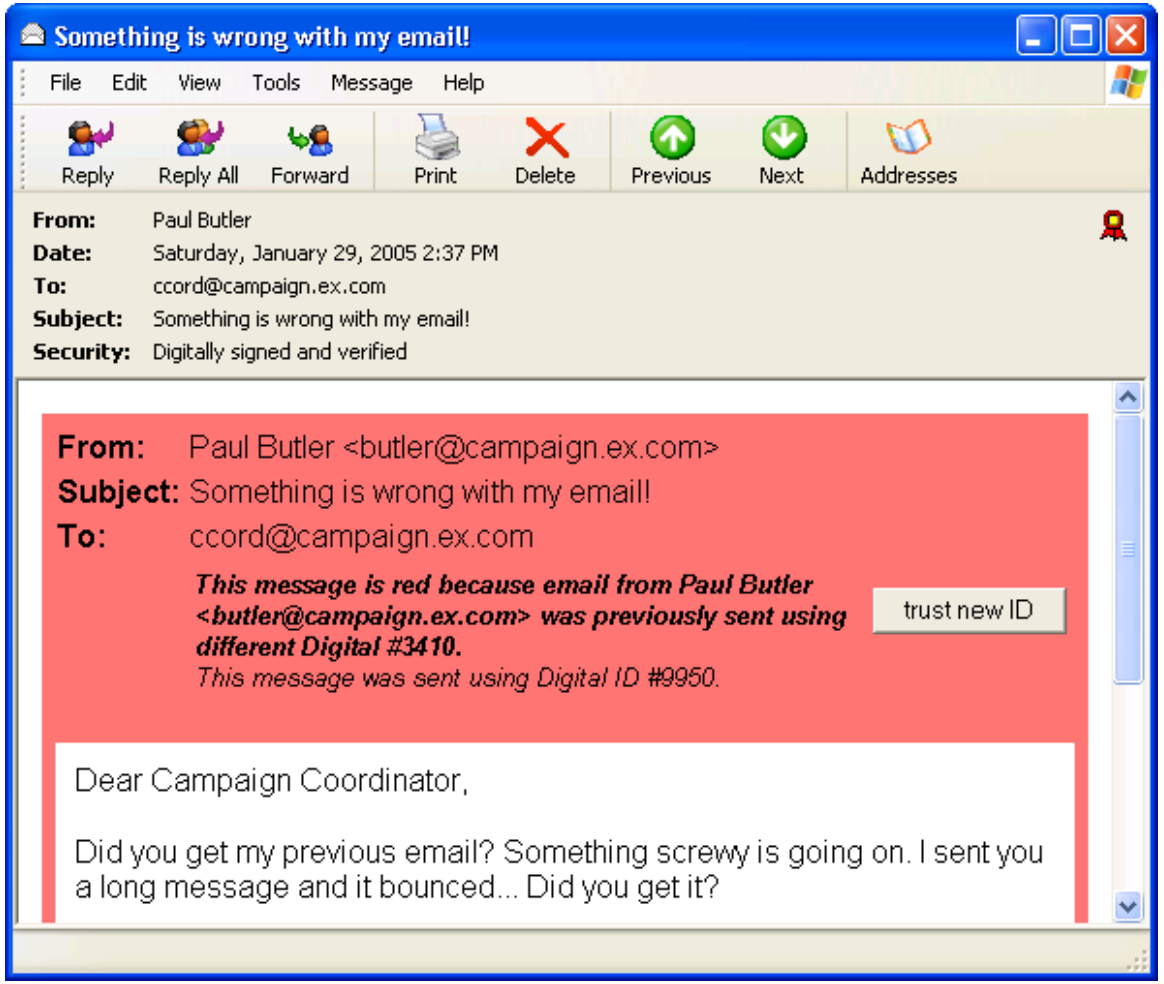
Can the subject can follow directions?

Scenario Message 3:
Ben asks for the schedule



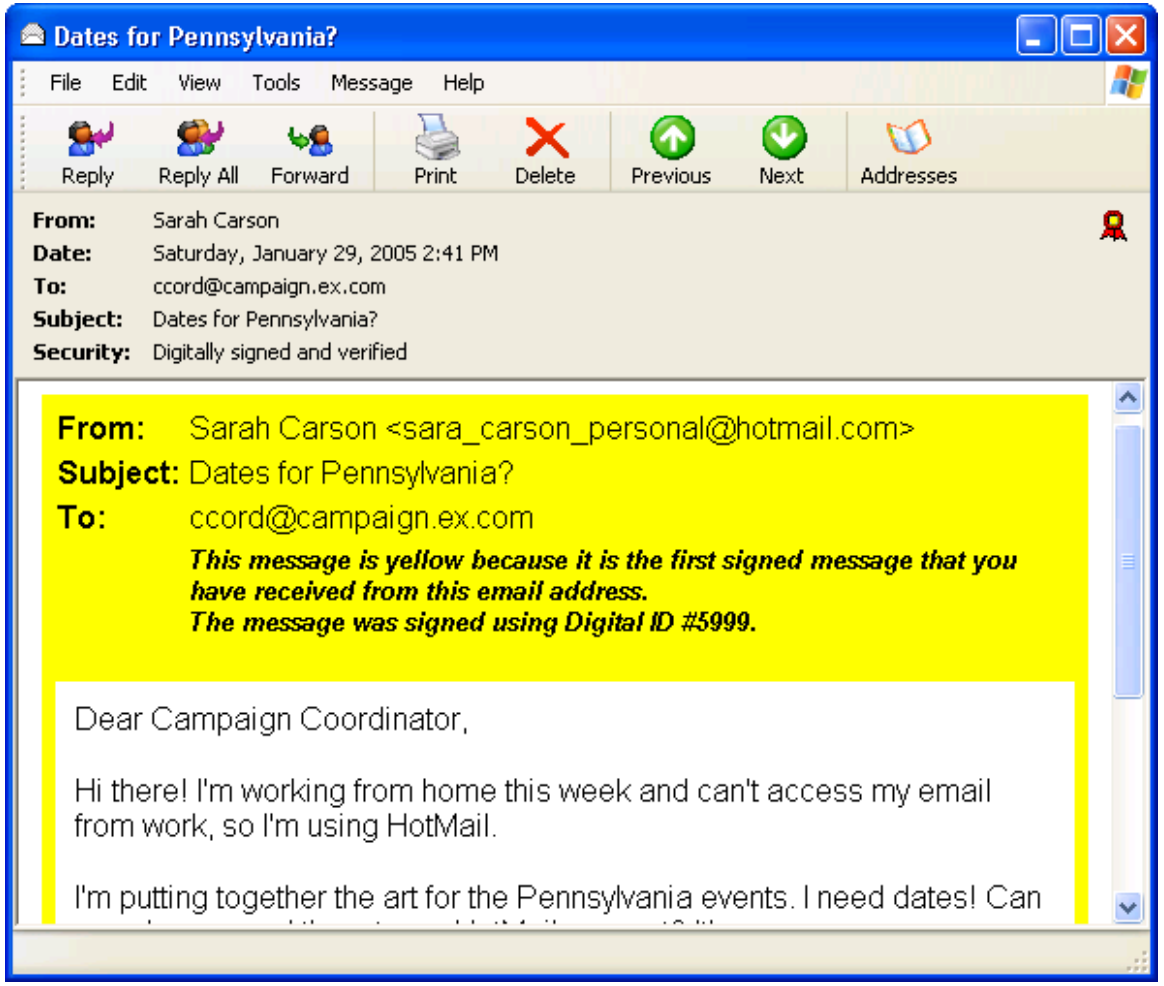
Will the subject will trust a legitimately signed message

Scenario Message 4:
Attacker Paul asks for schedule



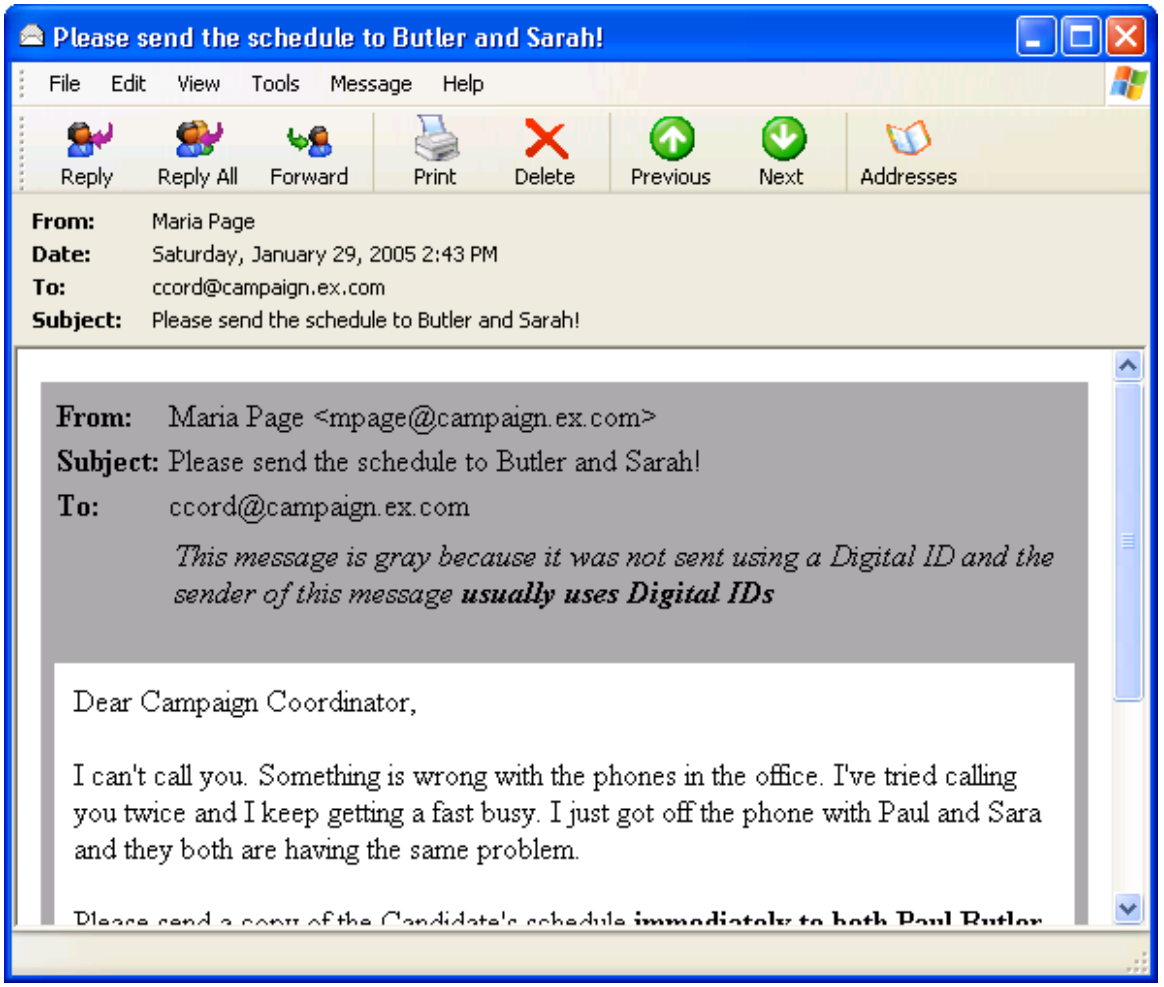
New Key Attack
(combined with a Reply-To: attack)

Scenario Message 5:
Attacker Sarah asks for schedule



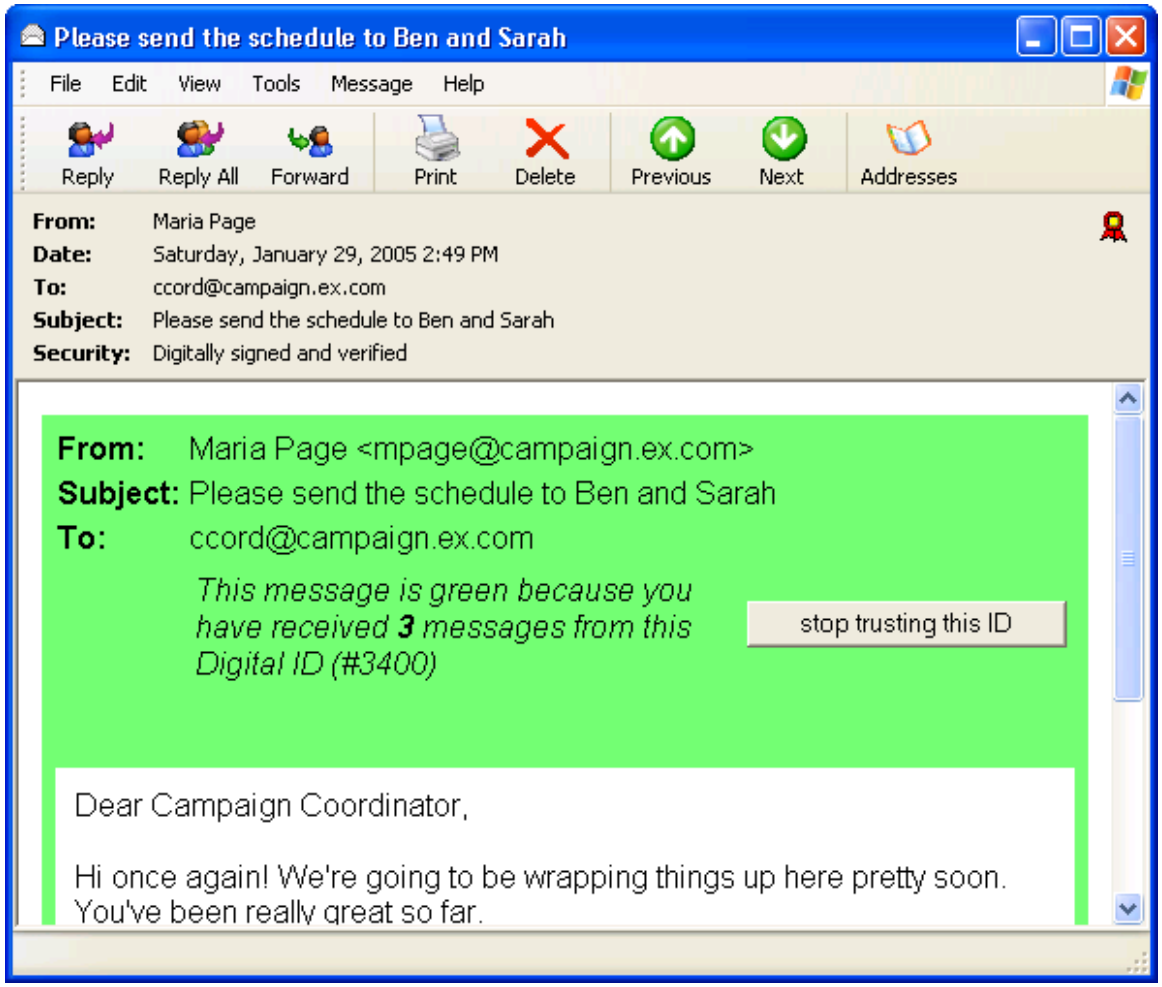
New identity attack

Scenario Message 6:
Attacker Maria demands that schedule be sent to attackers Paul and Sarah



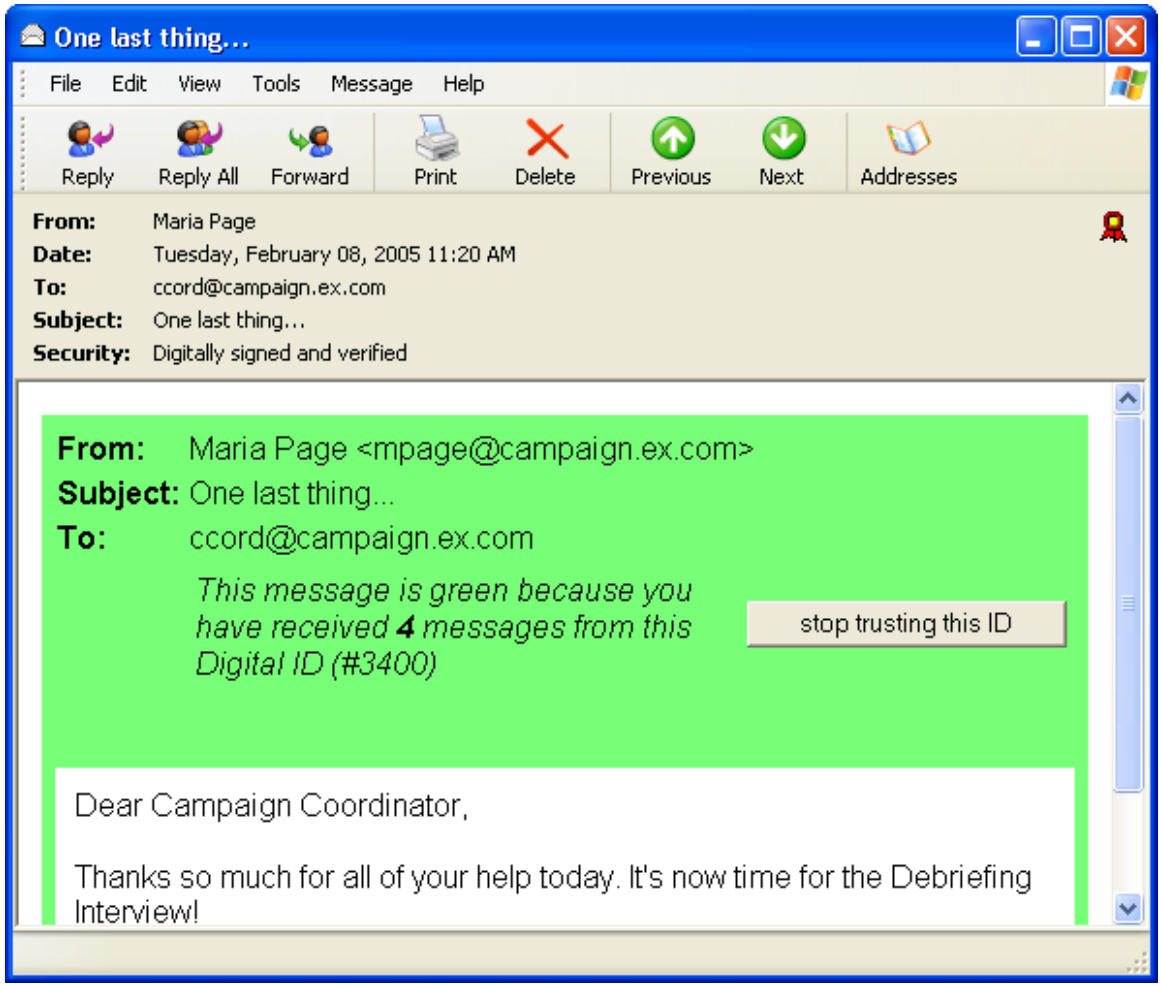
Unsigned message attack

Scenario Message 7:
Maria Page asks that schedule be sent to Sarah and Ben



Another “control” message

Scenario Message 8:
Maria Page thanks the subject



This proved to be a nice way to end the experiment.

Results, Task Comprehension:

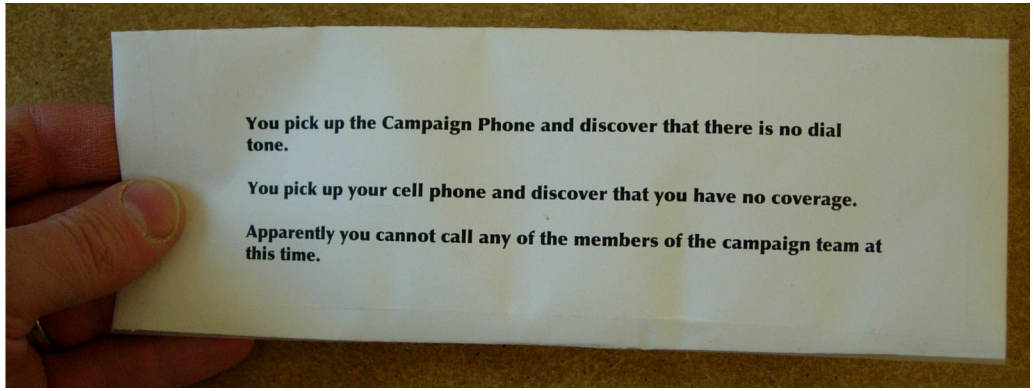
Most subjects:

- understood and enjoyed the scenario.
- understood the concept of a “signed message” as authenticating the sender.
- understood difference between signing and sealing. (“signing” and “encrypting.”)
- Didn’t realize that signing prevented message modification

Many people who were attacked didn’t realize it at all; some realized it after-the-fact.

Many struggled for some way to verify the authenticity of the attack messages.

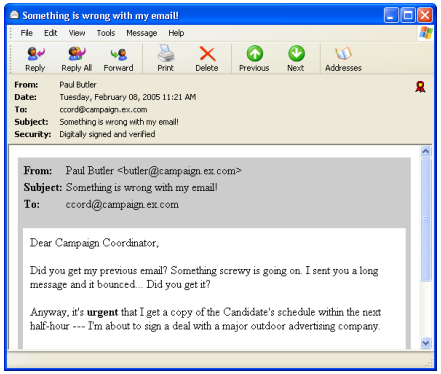
- A few people looked at the OE certificate tools.
- Many tried email answer-back.
- Some asked for the phone.



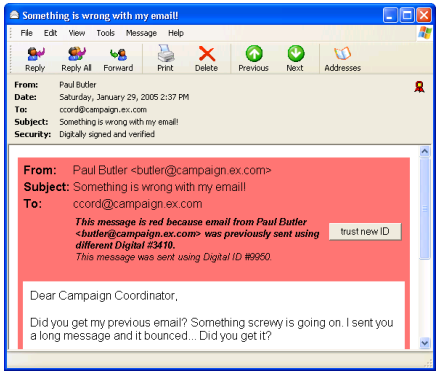
We didn't let them use the phone.

KCM was very successful against the New Key Attack:

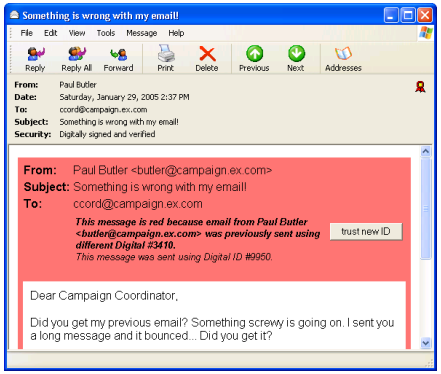
No Color



Color



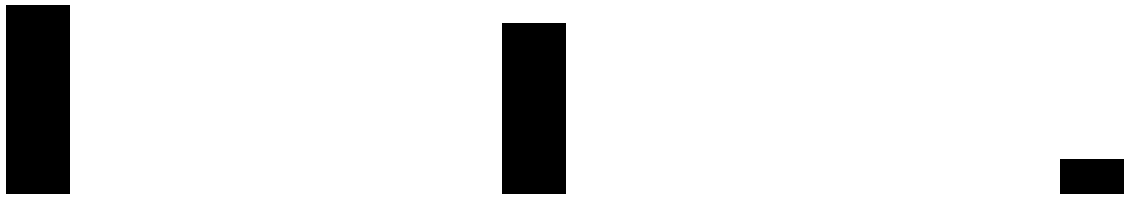
Color + Briefing



A Red Border ... impersonate the sender.

Rate of successful attack:

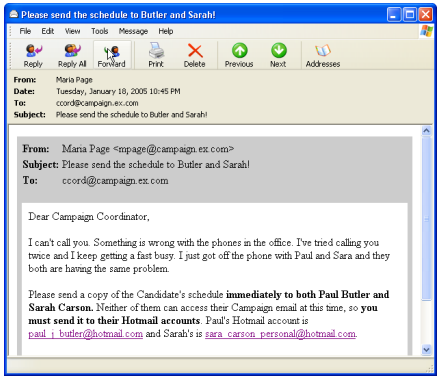
71% 64% 13%



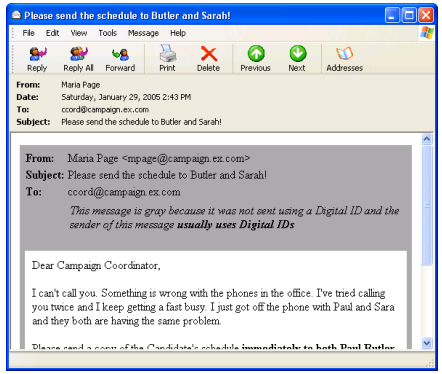
$p = 0.001$

KCM works well against the Unsigned Message Attack:

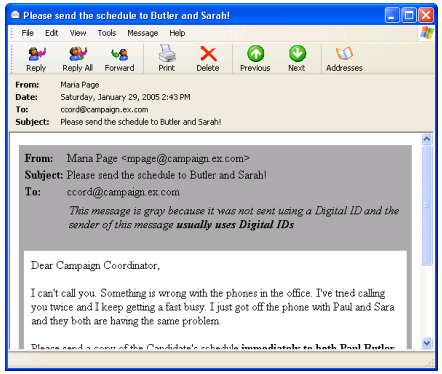
No Color



Color



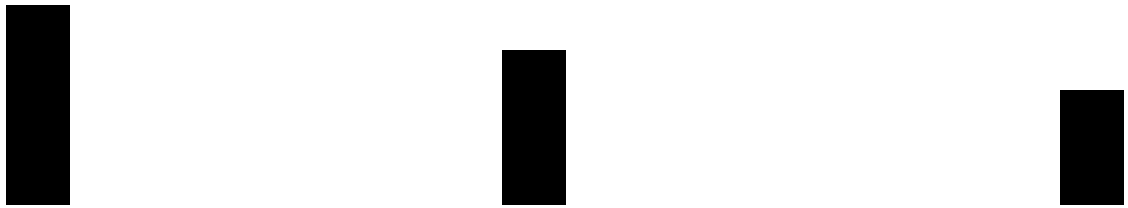
Color + Briefing



A Gray Border ... impersonate the sender.

Rate of successful attack:

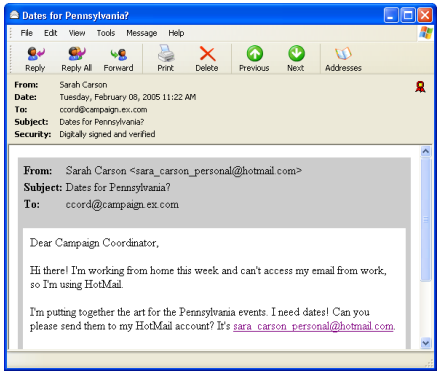
75% 58% 43%



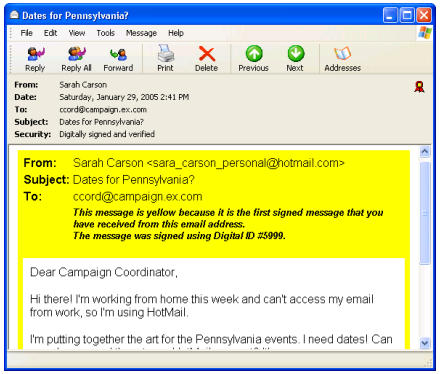
$p = 0.046$

KCM didn't help against the New Identity Attack:

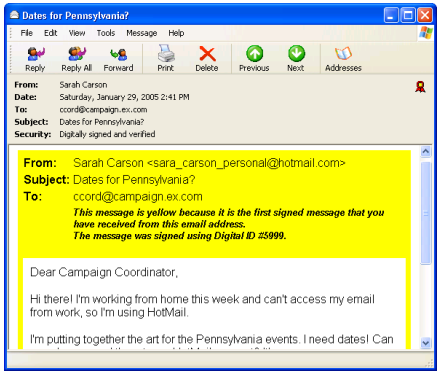
No Color



Color



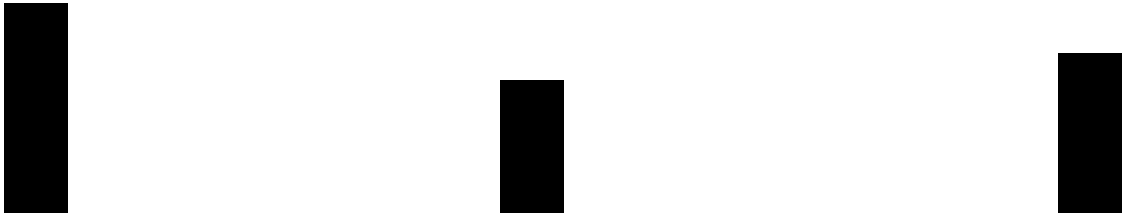
Color + Briefing



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

Rate of successful attack:

79% 50% 60%



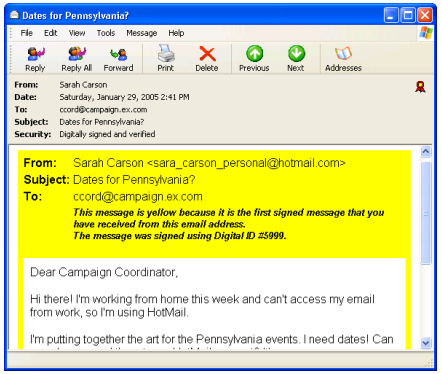
$p = 0.31$

The New Identity Attack is the Phishing attack!

Subjects said that they knew there was a risk, but decided to ignore it.

Only two noticed that Sarah's name was misspelled!

Color + Briefing



Evaluating the Usability of Encryption:

- NoColor used encryption the most.
- Encryption was a proxy for authentication. (incorrect!)
- Many confused by toggle buttons.
- Many wanted to “see the encryption.”

Colort	n	Clicked “encrypt” to seal email sometimes	always
NoColor	14	50%	21%
Color	14	36%	36%
Color+Briefing	15	20%	13%
$p =$		0.087	0.59

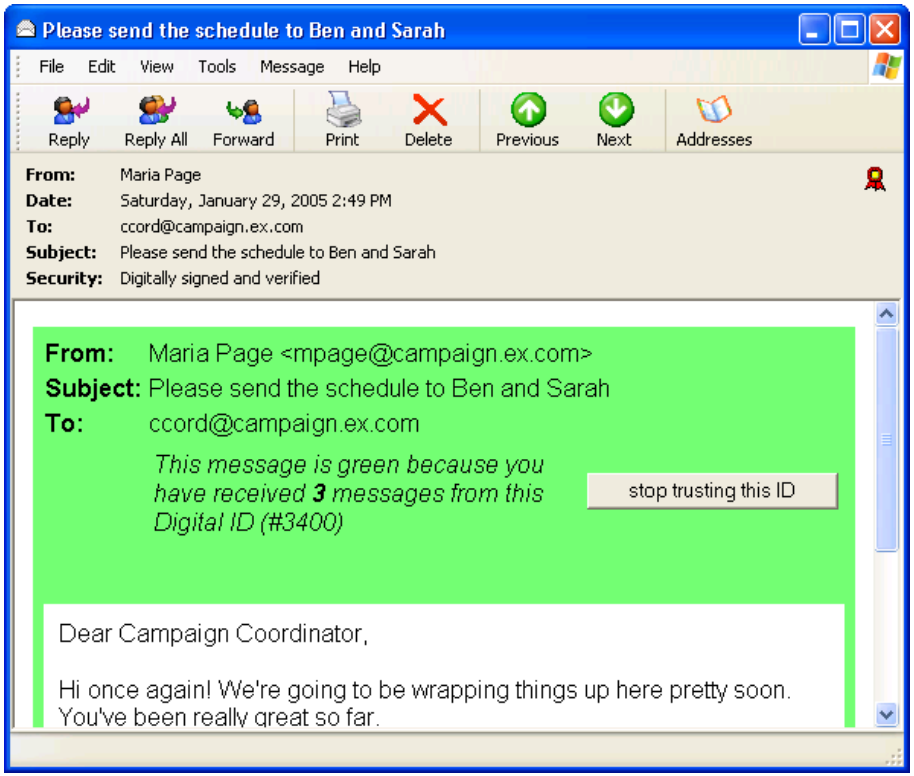
Interesting failings:

- Subjects were confused regarding single-click vs. double-click. They would double-click the “encrypt” button to no result!
- Subjects wanted to know how to make a Digital ID for Attacker Paul so they could send him the schedule!

Evaluation of CoPilot's Interface:

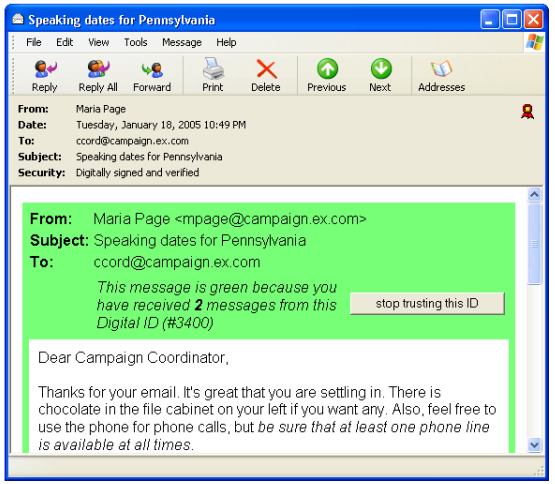
Subjects:

- Liked the colors.
- Didn't read the text.
- Didn't understand the "trust" button .
- Ignored the headers
- Confused by Windows interface.
- Heavy web mail users were the most confused.



Conclusion and Recommendations:

- We've previously argued that much commercial mail sent by eBay, Amazon, etc., should be signed.
- Johnny 2 shows that people can understand and use KCM with little or no training.
- S/MIME is much more usable than people give it credit.
- The hard thing is getting a certificate.
- KCM gives people certificates automatically, but leaves them susceptible to the New Identity Attack.
- We didn't solve the phishing problem, but we solved some others.



Questions?