Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software

Steve Sheng
Engineering and Public Policy
Carnegie Mellon University
shengx@cmu.edu

Levi Broderick
Electrical and Computer Engineering
Carnegie Mellon University
lpb@ece.cmu.edu

Colleen Alison Koranda
HCI Institute
Carnegie Mellon University
ckoranda@andrew.cmu.edu

Jeremy J. Hyland
Heinz School of Public Policy and Management
Carnegie Mellon University
jhyland@andrew.cmu.edu

ABSTRACT

Our research seeks to understand the current usability situation of email encryption software, particularly PGP 9 in comparison to previous studies of PGP 5. We designed a pilot study to find current problems in the following areas: create a key pair, get public keys, verify public keys, encrypt an email, sign an email, decrypt an email, verify a digital signature, and save a backup of public and private keys.

1. INTRODUCTION

In the seminal paper "Why Johnny Can't Encrypt", Whitten and Tygar [1] showed that users have great difficulty using email encryption software PGP. In the study, only 4 out of 12 participants were able to correctly sign and encrypt an email message in 90 minutes; and one quarter of them accidentally sent the secret email in clear text. They concluded from the usability test that "designing security software that is usable enough is a specialized problem, and user interface strategies that are appropriate for other types of software will not be sufficient to solve it [1]." Garfinkel, however, interpreted these results differently; he argued that the usability issues that Whitten and Tygar identified were driven by the underlying key certification model used by PGP [2].

Eight years passed, major changes have been made in PGP such as semi-automatic key creation and distribution, opportunistic encryption through email proxy, and automatic email decryption. The overall key certification architecture still has not changed.

Our research seeks to understand the current usability situation of email encryption software: What problems have the new features solved, what problems still remain, are there new problems been introduced? PGP claims that it is designed to support 'first time users,' as encryption is much more transparent.

We ran a pilot of the study with six novice users using PGP 9 and Outlook Express 6.0. Even though we only performed a pilot study, several patterns emerged early to indicate major problems in PGP 9. Users completed the following tasks: create a key pair, get public keys, verify public keys, encrypt an email, sign an email, decrypt an email, verify a digital signature, and save a backup of public and private keys. We also spoofed a decrypted email message to test user's response to PGP's automatic decryption.

2. MAJOR FINDINGS

2.1 Verify Keys

We found that key verification and signing is still severely lacking, such that no user was able to successfully verify their keys. Similar to PGP 5, users had difficulty with signing keys. Three of our users were not able to verify the validity of the key successfully and did not understand the reasoning to do so. Four users were not able to sign the key, these users attempted to but struggled with the interface. They did not understand that in order to 'verify,' they must 'sign' the key rather than just click 'verify.'

2.2 Encryption

We found that the transparency of the software's operation is problematic. The greatest difficulty for the users was in determining whether the software would operate as requested, as no indication was given during message composition as to whether or not the outgoing data would be encrypted or signed. Notification of successful encryption only occurs after the email has been sent. If the email is sent unencrypted, there is no visible feedback to indicate this to the user. The fact that users kept using the S/MIME toolbar in Outlook Express demonstrated that they were not aware of PGP's background automation. Thus, none of our six users were able to encrypt. The transparency in automatically decrypting emails also makes user susceptible to spoofing attacks against messages that appear to be PGP verified.

2.3 Digitally Sign

Digital Signing of messages is more problematic in PGP 9 than PGP 5 as none of the users were able to sign message using PGP 9, because there are no cues in the interface that support digital signatures. This can only be completed by right clicking, on the PGP system try icon.

3. ADDITIONAL FINDINGS

3.1 Create Keys

Users generally had no problem creating keys. This is an improvement in PGP 9 because a key generation wizard.

3.2 Send Public Keys

Two users were unable to send their public keys to others. In PGP, the 'Email this key' option appears only after the key is selected and it was difficult to identify the key location.

3.3 Get Public Keys

Three out of six people were able to get all public keys. For two of the users, the problem was that they typed in a partial name or email address, using PGP's 'contains' field but could not find the key. In PGP, the search relies on entering the text regardless. In addition, one user could not identify the location for key search.

3.4 Decryption

All users were able to decrypt. This is because PGP automatically decrypts emails when they appear in Outlook Express. We attempted to spoof emails by sending text that looked like it was decrypted. Two out of five users were unable to correctly identify legitimate emails manually, by comparing the correct key in the email to the key in PGP. Even though decrypting occurs automatically, we feel that further research should be done to evaluate PGP's automation decryption and spoofing decryption.

3.5 Key Backup

Four out of six people were able to create their backup keys. This task was relatively simple compared to the previous tasks. For the users that were unable to complete this task, one did not notice the 'Include Private Key(s)' checkbox at the bottom of the otherwise standard Windows save file dialog. Another user was never able to figure out that he needed to 'Export' his key to save a backup. Users were searching for the word backup in the interface, and those that were able to complete the task, spent a lot of time searching for it.

4. IMPROVEMENTS TO PGP

In summary, compared with Whitten's study of PGP 5, PGP 9 made strides in automatically encrypting emails. The key certification process becomes the key to the issue in PGP 9 has not made any improvements. PGP 9's presents multiple instances where the interface does not provide enough cues or feedback for the user. Based on the pilot test, we suggest the following design improvements for PGP:

a) For novice users, the location of 'your key' needs to be more apparent. The actions that users want to perform with their key should be better supported, such as emailing their key and encryption.

b) Deeper integration or a clearer link between PGP and mail client is required so users understand what actions can be performed in each location.

c) The search interface for obtaining others' keys needs to be clearer. The 'contains' option is misleading and prevents users from accomplishing their task.

d) The interface for signing an email is not apparent. The common tasks that PGP allows should be predominant in the main interface, and not put solely in a system tray icon.

e) More prominent cues are required for users to validate a key. Clicking on the different options that display validity should direct users to how they can sign the key to make the validity turn green.

f) Give users feedback prior to encrypting. This could occur by letting the users determine when they want an email to be encrypted and when they do not. Users need to be able to know ahead of time if their email will be encrypted successfully or not.

g) Users need a simple way to verify email validity. Many users requested a button that will connect email client to PGP to find out if the email matches the information in PGP

5. ACKNOWLEDGMENTS

We would like to thank our Usable Privacy and Security Professors: Lorrie Cranor, Michael Reiter, and Jason Hong for their help and inspirations.

6. REFERENCES

[1] Alma Whitten and J.D. Tygar, Why Johnny Can't Encrypt: A Usability Case Study of PGP 5.0. Proceedings of the 8th USENIX Security Symposium, August 1999.

[2] Simson L. Garfinkel and Robert C. Miller, Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express. Symposium On Usable Privacy and Security (SOUPS), 2005.