WHY JOHNNY CAN’T ENCRIPT: A USABILITY EVALUATION OF PGP 5.0

Alma Whitten, J. D. Tygar
OVERVIEW

- Offer a specific definition of usability for security
- Identify several significant properties of security as a problem domain for user interface design.
- Perform a detailed case study of the usability of PGP 5.0 which was representative of the best current user interface design for security by general standards.
Published: USENIX Security Symposium, 1999

Subject: Security usability of the interface of PGP 5.0

**PGP (Pretty Good Privacy):** an encryption program that provides cryptographic privacy and authentication for data communication (email for this paper).

**PGP 5.0:**
- Released around June, 1996
- Support OS: Windows '95, Windows NT and Macintosh
- Come up with a good user interface by general standard.
- Major upgrade and improvement to PGP 2.6.2:
  - New public key infrastructure based on the Diffie-Hellman (DH) while RSA algorithm is still supported.
  - Automated keyservcer support
  - Plug-ins for Eudora and other major mail utilities
WHAT IS USABILITY AND HOW TO MEASURE IT?

- Usability: The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. (ISO Definition)

- Discussion: what aspects should we consider when measuring the usability?
  - Stability
  - How easy it is to learn: learnability
  - (Un)Accessibility: can be used by large group of people

  - Speed: How quick can the task be accomplished? (timing)
  - Efficiency: how many mistakes are made in accomplishing this task? (counting errors)
  - Learnability: how easy is it to learn to use this system? (drawing a chart)
  - Memorability: once learned, how easy is it to remember how to use this system? (drawing a chart)
  - User preference: what do users like? (survey)
DISCUSSION: REGARDING USABILITY, WHAT’S THE DIFFERENCES BETWEEN GENERAL SOFTWARE AND SECURITY SOFTWARE? (E.G. RESULT OF BAD/GOOD USABILITY)

▪ General software:
  ▪ Security is secondary, must be default or inbuilt

▪ Security software:
  ▪ Cannot mess up, cannot make a single mistake
  ▪ Users have good context
  ▪ Balance between user control and prevent user mistakes
  ▪ User’s intention might vary
**Definition:** Security software is usable if the people who are expected to use it:

- 1. are reliably made aware of the security tasks they need to perform;
- 2. are able to figure out how to successfully perform those tasks;
- 3. don’t make dangerous errors;
- 4. are sufficiently comfortable with the interface to continue using it.
Discussion: What security or privacy scenarios where usability might play a big role?

- Warnings and passwords
- Encrypted communications
- Permissions and access control
- Mobile apps
- Firewall
- Biometrics
- Certificates
- Just work
- Setting up devices, like WIFI
- Software development tools
EVALUATION METHODS

- In the paper
  - Inspection: Cognitive walkthrough
  - User test: Performance Measurement

- Discussion: Is there any other evaluation methods for the paper’s study?
  - Log down users’ behaviors in normal environment
  - Survey: what do they expect for certain operation
  - Limitation for User test:
    - a lot for first timers.
    - Limit how the user could get info
  - Cognitive walkthrough:
    - Too subjective

https://www.usabilityhome.com
**Cognitive Walkthrough - PGPTools**

**Problems**
- Failed to show the difference between public and private keys
- Failed to show signing is accomplished by keys
- Verify doesn’t have its own icon, people don’t know what is being verified.

**Solutions**
- Use icons like puzzle pieces
- Show private key as the nib of the pen
- An icon showed a public key unlocking the signature inside

- Discussion: Do you have other problems / better solutions?
COGNITIVE WALKTHROUGH — PGPKEYS

Problems

Hard to understand differences between DH and RSA keys

Too much information

Solutions

Double-click to show the key type info

Show advanced information only if users choose to

Discussion: What information about keys do you think is necessary to show to users at all levels?
COGNITIVE WALKTHROUGH — ENCRYPTION

Problems
Choose people to receive messages rather than choose keys to encrypt the message

Validity and Trust could be confusing

Solutions
Let users choose which key to encrypt instead of the receiver

Let users know how did validity generate

Discussion: Which one do you think is better? To choose key or the receiver? And why?

- Key:

- Receiver:
COGNITIVE WALKTHROUGH — KEY SERVER

**Problems**

- Users perhaps don’t know the existence of key server
- PGP retrieves keys from server in MIT by default
- Revocation operation does not send the certificate to the key server

**Solutions**

- Show key server at the top-level
- Show remote machine and its identity
- Remind users to publish their certificates after revocation
COGNITIVE WALKTHROUGH – IRREVERSIBLE ACTIONS

- Irreversible actions
  - Accidentally deleting the private key
  - Accidentally publicizing a key
  - Accidentally revoking a key
  - Forgetting the pass phrase
  - Failing to back up the key rings
What other problems have been found?
- Default never expiring key (although recommended)
- Default not to send the key to the key server

What’s the pros and cons of a never expired key pair?
- Pros:
  - Users do less
- Cons
  - Keys might be leak
  - Signatures will expire
Discussion: what principles should be followed when choosing the participants for this study (evaluate the usability of PGP 5.0)? And how will these principles impact the study?

- Spread in demographics (occupations, educations)
- Motivation: Need for private communication (dark web forums)
- Beginners
USER STUDY – TEST DESIGN

- Background: political campaign
- Participants
  - 12 people
  - Age range from 20 to 49
  - Experienced with email
  - At least college degree
  - None of whom could describe the difference between public and private key
- Tasks
  - Initial task: send the secret message to the five team members in a signed and encrypted email (one of the team members had an RSA key while the others all had Diffie-Hellman/DSS keys).
  - If successfully finish the initial task: receive an email from one team member. Decrypt the email and verify the signature.
USABILITY STANDARD FOR PGP

- For a user:
  - understand that privacy is achieved by encryption, and figure out how to encrypt and decrypt emails;
  - understand that authentication is achieved through digital signatures, and figure out how to sign email and verify signatures.
  - understand that a key pair must be generated to sign email and to send encrypted email, and figure out how to do so;
  - understand that they must publish their public key to allow others to verify their signature and to send them encrypted email, and figure out how to do so;
  - understand that in order to verify signatures on email from other people and send encrypted email to other people, they must acquire those people’s public keys, and figure out some way to do so;
  - manage to avoid such dangerous errors;
  - be able to succeed at all of the above within a few hours of reasonably motivated effort.
What operations are required to complete the initial task?
- generate a key pair;
- get the team members’ public keys;
- make their own public key available to the team members;
- type the (short) secret message into an email;
- sign the email using their private key;
- encrypt the email using the five team members’ public keys;
- send the result;

Which usability standards do these operations try to evaluate?
- understand that privacy is achieved by encryption, and figure out how to encrypt emails;
- understand that authentication is achieved through digital signatures, and figure out how to sign email.
- understand that a key pair must be generated to sign email and to send encrypted email, and figure out how to do so;
- understand that they must publish their public key to allow others to verify their signature and to send them encrypted email, and figure out how to do so;

Initial task: send the secret message to the five team members in a signed and encrypted email (one of the team members had an RSA key while the others all had Diffie-Hellman/DSS keys).
USER STUDY — SECOND TASK

- What operations are required to complete the second task?
  - decrypt the email using the participant’s private key;
  - verify the signature using the five team members’ public keys;

- Which usability standards do these operations try to evaluate?
  - understand that privacy is achieved by encryption, and figure out how to decrypt emails;
  - understand that authentication is achieved through digital signatures, and figure out how to verify signatures.
  - understand that in order to verify signatures on email from other people and send encrypted email to other people, they must acquire those people’s public keys, and figure out some way to do so;

- Second task: receive an email from one team member. Decrypt the email and verify the signature.
## USER STUDY — RESULT

<table>
<thead>
<tr>
<th>Tasks</th>
<th>participants</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deciding whether to trust keys from the key server</td>
<td>notice</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>notice</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>notice</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Avoiding dangerous errors</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Hard</td>
<td>Hard</td>
<td>Hard</td>
<td>Failed</td>
<td>Failed</td>
<td>Hard</td>
<td>Hard</td>
<td>Hard</td>
<td>Failed</td>
</tr>
<tr>
<td>Publishing the public key</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Hard</td>
<td>Hard</td>
<td>Hard</td>
<td>Hard</td>
<td>Failed</td>
<td>Failed</td>
<td>Hard</td>
<td>Hard</td>
<td>Hard</td>
<td>Failed</td>
</tr>
<tr>
<td>Getting other people's public keys</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Figuring out the correct key to encrypt with</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Signing an email message</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>send correctly encrypted email</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Handling the mixed key types problem</td>
<td></td>
<td>Didn’t meet</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Decrypting an email message</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Verifying a signature on an email message</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Creating a backup revocation certificate</td>
<td></td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
</tbody>
</table>
IMPROVEMENTS

PGP 5.0:

PGP 8.0: