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

Alma Whitten and J. D. Tygar

Presentation by Jukka Valkonen
jukka.valkonen@tkk.fi
25.10.2006
Outline

1. Background
2. Test methods and results
3. Conclusions
Pretty Good Privacy (PGP)

- Pretty Good Privacy
- First version in 1991
- PGP 5.0 released in 1997
- Encryption and digital signing of email messages based on public key cryptography
- Additional software used together with an email client
- Currently ~9.0
Why PGP?

• PGP 5.0 was at the time the best candidate for the study
  – The user interface seemed to be appropriate
  – The documentation marketed the software to be designed with usability in mind
    • “... graphical user interface makes complex mathematical cryptography accessible for novice computer users”

• PGP 6.0 was already released
Security software is usable, if the users

1. are reliably made aware of the security tasks they need to perform
2. are able to figure how to successfully perform those tasks
3. don’t make dangerous errors
4. are sufficiently comfortable with the interface to continue using it
The big question is:

• “If an average user of email feels the need for privacy and authentication, and acquires PGP with that purpose in mind, will PGP’s current design allow that person to realize what needs to be done, figure out how to do it, and avoid dangerous errors, without becoming so frustrated that he or she decides to give up using PGP after all?”
Thus the users need to

- Understand the need of different keys, own key pair, the public keys of other’s
- Understand encryption
  - figure out how to encrypt/decrypt
- Understand authentication
  - digital signatures
- Be able to do this quite easily
Cognitive Walkthrough

• Use the software and try to identify problems
• The evaluators think as novice users
• Also some features from heuristic evaluation
  – Evaluation performed against some usability principles
Some observations

- Keys and locks used heavily in the UI
- “Same key” used in encryption and decryption?
- No keys used to sign?
- What is verification after decryption?
More...

• Different figures to represent different key types (RSA or Diffie-Hellman/DSS)
  – The meaning of the figure is told only in the manual (132 pages)
• Too much information
  – An average user doesn’t care about the key length
• Too easy to do irreversible actions
Still...

• Key management
  – Validity and trust
  – PGP automatically derives the validity ratings
  – This should be more obvious for the user

• Consistency of terminology
  – encrypting <-> encoding
Some proposal for improvement

• Renew the figures
  – different keys used to encrypt/decrypt (still, the keys should form some figure, as the keys work together)
  – Somehow attach the key to signature
  – Decryption/verification: private key unlocking the secret, public key unlocking the signature

• Renew the given information
User test

- 12 participants, who were experienced users of email
- None of the participants knew the difference between public and private key cryptography
- Eudora was used as mail client
  - Tutorial given before the test
  - Questions related to Eudora were answered during the test
The participants were provided

- A secret message
- The names and email addresses of five recipients
- A request to send the secret message to the participants signed and encrypted
- 90 minutes
- Software and connection to a key server
Also

• To make life more complicated
  – One recipient had RSA key, while others had Diffie-Hellman/DSS keys
  – After succeeding in the initial task, the participants received (encrypted) emails from the original recipients with more instructions.

• The evaluators answered to emails sent to the fake recipients
All they had to do, was

- Generate a key pair
- Get the public keys of participants
- Publish their own public key
- Type the message
- Sign the email using private key
- Encrypt the message using recipients public key

This can’t be difficult, right?
Some results

• 11/12 were able to encrypt (with some key)
• 5/12 were able to encrypt and sign using the right keys
  – Only 2 managed to decrypt without problems
• 10/12 managed to publish their public keys
• 8/12 successfully got the public keys of the recipients
• 3 revealed the secret message by sending it in plaintext
Conclusions

• “It (PGP) does not make public key encryption of electronic mail manageable for average computer users”

• Building security-specific user interfaces needs different principles and techniques if compared to normal UI designing

• Also evaluation methods of usable security need to be revised (are they already?)
Thank You!

Questions?