Why Johnny Can't Encrypt: A Usability Study of PGP

Jan Sousedek
Technische Universität Berlin, Germany
Erasmus program
Summer semester 2008
Seminar: Internet Security
jan.sousedek@seznam.cz

Abstract

Interfaces of security software are very often clumsy or confusing. Users are dealing with bad usability standard, that this type of software offers. Users have to often get used to not very user-friendly interface. Because of these aspects, even if they try to use this security software, they give up and don't use security at all.

This seminar paper deals with some confusing aspects of user interface of PGP 5.0. Cognitive walkthrough analysis and laboratory test revealed some interface design flaws that cause user to make security failures or not to use encryption at all. Despite of attractive user interface of PGP 5.0, users are very often confused and make mistakes. The difference between PGP 5.0 and its newer version PGP 9.0 are discussed as well.

1. Introduction

This seminar paper is mainly about case study of Alma Whitten and J. D. Tygar: Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0 [1] and another consequential study of Steve Sheng, Levi Broderick, Colleen Alison Koranda, Jeremy J. Hyland: Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software [2]. These studies pointed out that even if the strong encryption and correct protocols are used, security really depends on users. If the users do not have enough knowledge of the communication protocol (which cryptographic key to use) or they often forget to encrypt by clicking on the right button, the security can be easily compromised. Another problem is security software configuration. Average computer users are not very familiar with computer technology and they may experience big troubles with using security software. By small mistake users can publish their private keys and the whole concept of security is useless.

Because security can be effective only when used properly, the first study pointed out what problems are users having when using PGP 5.0. PGP 5.0 was selected as good representative of a general standard of security software. PGP is software primarily used for email security. This software was first tested by cognitive walkthrough analysis, which analyzed user interface and how the user can misinterpret it. Secondly laboratory user test was used. Average citizens were supposed to use PGP 5.0 to encrypt and sign email, to exchange keys etc. Most of them were unable to do that in deadline of 90 minutes.

These problems are very often caused by improper interface design. These studies revealed, that user interface for security programs should be designed in little bit different way, so also computer novices can use it.

2. Overview of PGP

The abbreviation PGP stands for Pretty Good Privacy and it is the name of software product used mainly for email encryption and authentication [3][4]. This is achieved by key pair – public and private key. Public key is distributed and published to other users and then used by other side to encrypt the message. This encrypted message can be afterwards seen only by the person, who owns the proper private key from the key pair used. Private key should be always kept and stored on secure place and nobody else should have access to it. Private keys are also used for authentication. When somebody wants to sign message, private key is used to do that. Then everybody who has the sender's public key can validate the message.

3. Problems of security usability

Because usability of user interface has different meanings in different situations, design should focus on how the software will be used. For example in some situations efficiency and flexibility are the priorities. In the security context the priorities must comply with whatever that makes security to be used effectively and without big mistakes. As said in [1], security software is usable only if the people who are expected to use it are reliably made aware of the security tasks they need to perform, they are able to figure out how to successfully perform those tasks, don't make dangerous errors and are sufficiently comfortable with the interface to continue using it.

Properties of security deal with various problems. One of them is very often problem with unmotivated users to use security. Their primary task is to work with computer and not to manage their security all the time. They are often expecting, while working, security will be used automatically to protect them. Interface designers and programmers should be aware, that ordinary PC users are not motivated to read thousands of manual pages just to send one encrypted mail, if they don't need to do so.

Rather that that, they may find it very difficult and may give up on it altogether. Another big problem is lack of feedback property. Security software should provide user with good feedback to manage security functions. Very often is configuration of security software too complex and setting it up may only confuse users. Dangerous wrong configurations can be created and security can be then easily compromised. Feedback should warn users before deploying such mistakes. Once the secret has been accidentally unprotected, even after restoring it back to the original state, nobody can be sure, that possible attacker didn't reveal the secret. User interface design should therefore place high priority on avoiding users making dangerous errors and to make sure they understand the way how security works. The rule of the old proverb: "the chain is as weak as its weakest link" applies to the concept of security as well. That means, that users should take care of all aspects of the security and not only part of it. Properly designed interface should help them to achieve that.

3.1. PGP 5.0 usability

The case study [1] focused on the question of usability of PGP 5.0. Creators and marketing presents PGP as a product intended to be used among different groups of users. That's why it has to comply with good usability standards. The question was whether users are able to perform following tasks in the given time period:

- Understand the concept of encryption and how to use it
- Understand how authentication works
- Understand why to generate key pair and how to do it
- Understand why to publish public key and how to do it
- Acquire others' public keys
- Avoid dangerous errors

Users tried to perform these tasks in the laboratory user test. Another test – cognitive walkthrough was used to describe user interface and to point out some confusing aspects of it. Labels, icons, menus and interface were inspected.

4. Cognitive walkthrough

Cognitive walkthrough is a usability evaluation technique, where evaluators try to perform tasks, as if they were novice users. They try to simulate what would novice do and what could cause misunderstanding in the using of software interface.

4.1. PGPtools

In Figure 1 you can see graphical user interface of PGPtools. This tool maintains all important functions of PGP. Buttons represent these functions: PGPKeys, Encrypt, Sign, Encrypt&Sign and Decrypt/Verify. This buttons should be intuitive enough, but some confusing aspects can arise as well. First of all the button PGPKeys doesn't distinguish between public keys for encryption and private keys for decryption. The symbol of lock can be understood in the wrong way, that it locks and unlocks something, in this context to encrypt or decrypt message. Other possible misinterpretation could be the Sign button. The symbol of quill pen could be understood as used for signing, but user probably won't understand, that they have to use their private keys to generate signatures. Something, that looks like inked handwriting would be better representative of signing.

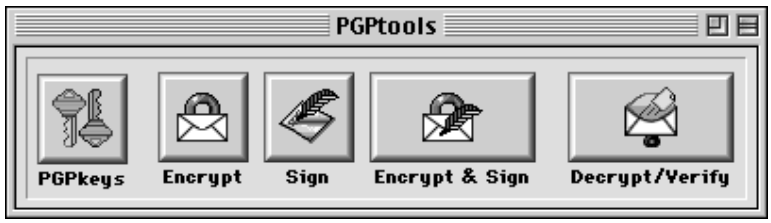

Figure 1

Signature verification is also not represented separately. The single button Decrypt/Verify with a symbol evoking only decryption could be misunderstood in a way, that this verification means, that decryption occurred correctly. Maybe the label showing a private key opening envelope and public key to unlock a signature would be more appropriate.

4.2. PGPKeys

Key management interface named PGPkeys can be seen In the Figure 2. In PGP 5.0 two different key types can be used. Originally PGP used RSA, and PGP 5.0 uses Diffie-Hellman/DSS keys. PGP would like to switch all users to use Diffie-Hellman/DSS keys. The difference between these two types of keys can be seen in the PGPkeys interface as two different icons left to the name of the key. RSA has old-fashioned key shape and Diffie-Hellman/DSS uses brass key with newer shape. Because the compatibility issue can arise, users are alerted by this way, that two different types of key can exist. Secondly they are warned when they try to use mixed

key types to encrypt documents, that recipients who have earlier versions of PGP may not be able to decrypt it. However, the information about the meaning of the two different key icons is difficult to find.

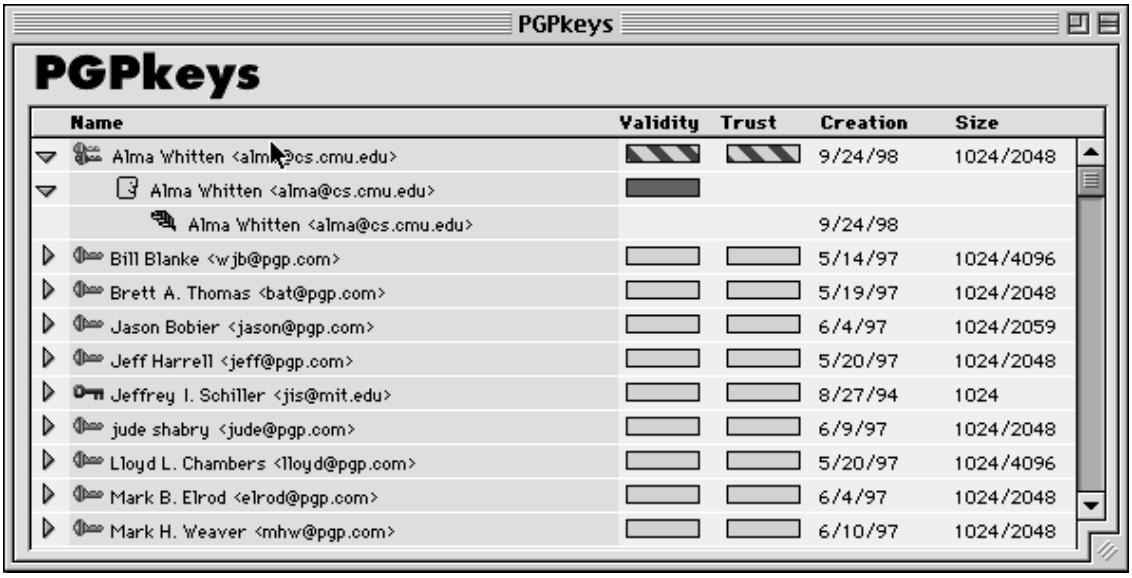

Figure 2

PGP stores two ratings for each public key. First of them is Validity to indicate how sure the user is that the key belongs to the person whose name it is labeled with. Key may be labeled as completely valid, marginally valid or invalid. The second rating is trust, which indicates how much do we trust another user as a certifier of other keys. Similarly, key can be trusted, marginally trusted or untrusted. The problem, that user can experience is when he accidentally change those ratings. These ratings are set automatically by PGP, but user may not know that. By accidentally exchanging these two rating, dangerous consequences may arise.

Another problem with PGPkeys interface may be too information. Information that is shown: owner's name, validity, trust level, creation date and size, plus signatures can be displayed on each key. This amount of data can easily confuse user, who may than focus on unimportant information. For example majority of user will anyway use simple mail transfer, only in certain situations they will encrypt their sensitive data. For this reason they actually don't have to really set up validity and trust level. This setting can be left automatically handled by PGP. Also creation date is totally useless information for the user. Rather that these information, PGPkeys user interface should focus on different key types and the actual model of private and public keys. Another information, that could be left out is the size of the key. This information is useful only for those, who are afraid of cryptographic attack, but for basic users is nearly useless. This loss of information may be for some experienced users alarming, but can help novice users to better understand the main concept of PGP.

4.3. Key server

Public keys can be published on the Internet via key server. This makes public key accessible to others. PGP offers three key server functions shown on Figure 3. These operations may be useful for users, but users may not even notice this function in Keys menu. This function would be better situated on the top of PGPkeys display. The way in which to distinct between operations that access remote machines and those that are purely local would be also helpful for users. Also identity of remote machine that is being accessed should be shown.

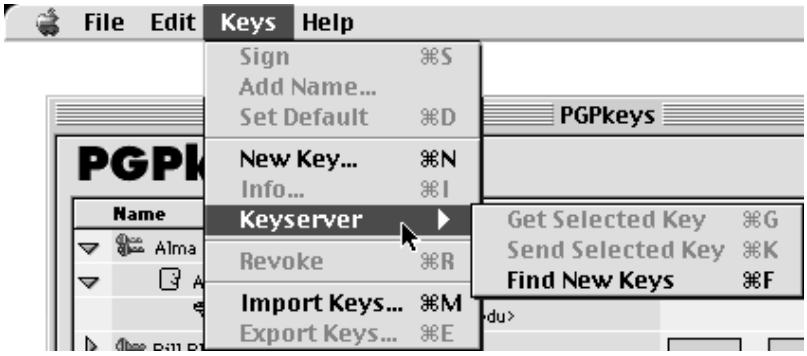

Figure 3

In PGP 5.0, PGPKeys do keep any records of accessing key server. No information about date and source from where was key downloaded or updated is available. Showing this information would improve PGP. Key revocation is the process when new public key is published and the old one should be automatically considered invalid. PGP automatically doesn't send new certificate to the server, so user may consider it done, but reality may be different. This should be improved.

4.4. Accidents

Some accidents that may happen can have really big consequences. Unfortunately most of them are irreversible. One of them may be accidental private key deletion. If public key is lost, it can be restored back from key server, or from other users, who may have it. However when user deletes private key, which is not backed up somewhere else, serious problems may arise. User won't be able to read decrypted messages from other users and to make revocation certificate for that public key. However message is showed, when user tries to delete private key, some additional

warning message would be appropriate as well.

Another confusing aspect may be publication of keys. When somebody adds key to the key server, this key can be revoked, but actually it is never deleted. When user is novice to PGP, may generate more key pairs, which he then add to the key server. User may not notice that, but after deletion of few of these keys, he will not be to generate a revocation certificate and therefore revoke old or deleted keys.

By accidentally revoking key user can get himself into big troubles. This action can by undone only by revoking backup copy of old keys. PGP shows user a warning, that with new key revocation, some users may not be able to encrypt data. However some users may understand it in the way that revoking the key also automatically distributes the key.

Another possibly confusing thing for beginners may be the concept of revocation certificate. Revocation certificate is recommended to store on a safe place, so the revocation process can be done in the future. This can help user in such a situation as forgetting the pass phrase. PGP offers users the possibility to backup their keys, but this can be improved by choosing another folder then the default one (which the user will probably use).

5. User test

Another test used in [1] was user test. This test tried to evaluate PGP 5.0 usability standard in praxis. Participants were supposed to solve the given security tasks. The motivation to use security was very easy – modeled scenario was a political campaign, where campaign coordinators were supposed to send their data encrypted, so secrets could not be revealed. Because political campaign is also about the money, participants were well motivated to use security.

Because PGP doesn't handle email operations itself, users were given Eudora email client with PGP plug-in installed. Participants were also given tutorial, how to use this email client, so they should not have any problems with anything that has nothing to do with PGP. Also the given scenario was described to the participants. Then they received the secret message, the names and email addresses of four campaign members and one campaign manager. This secret message was supposed to be sent to other five members in a signed and encrypted email. To achieve this, they were supposed to generate a key pair, get others public keys, distribute their own public key, send encrypted and signed message and send the result. In addition to this, one of the members had RSA keys and others had Diffie-Helmann/DSS keys. This means, that they had to use mixed key types. The campaign manager's private key was used to sign his own keys and also another members' keys. Also test monitor was used as a member of the campaign team that under some circumstances replied to the users from appropriate dummy email account.

Test session lasted 90 minutes, in which participants were supposed to finish their tasks. Manuals for Eudora and PGP have been at disposal.

Twelve different participants took part in this user test. All of them had previous

knowledge about emails, but nobody had previous knowledge of the key pair cryptography model. All of them had attended college. Their age was from 20 to 49 and they had various professions and job positions.

5.1. Results of the user test

Even tough the task was as simple as the operation that the user will do every time when using PGP, participants didn't achieve very optimistic results.

Three of the twelve participants sent secret message in unencrypted email. One of them didn't even notice that. He supposed the encryption would be transparent for him. All these mistakes have been done in the initial phase of exploring the system.

Another problem for participants was to figure out how to actually encrypt with key. One of them tried to find how to "turn on" encryption, and believed he had done it by modifying some preferences in PGPkeys. Another one of twelve took 30 minutes to find out how to encrypt.

Most of the participants had problems with understanding the public key model. Seven of twelve participants used their own public key to encrypt message to the team members. One of them used only one public key of campaign manager to send encrypted message to all others. Some of them didn't understand the problem that other members are not able to decrypt the message even after receiving feedback from the test monitor explaining what should be done. A lot of them were unable to obtain other member's public keys. Another one totally misunderstood the model and generated key pairs for each team member rather then for himself and used them afterwards.

Only five participants received encrypted email from a team member. One of them tried to decrypt it 25 minutes. Another one of them thought, that he received public key of another member, not the encrypted message. One had difficulties to decrypt for 10 minutes, that he succeeded. Last two were able to decrypt without problems.

Only two participants had problems with key distribution, because they were overwhelmed with previous tasks. Other ten sent keys to the key server or mailed them to another members. The main confusion of users arose from misunderstanding which icon represents their public keys. They were worried, that by sending their key to the key server, they could accidentally publish their private key.

Eight of twelve participants successfully obtained their team members' public key through the key server. Two of the other four, who failed to obtain keys, never understood the concept that they had to do so. Other two spent a lot of time trying to do so, but finally failed.

Handling of the mixed key types was another problem. Only one of the successful participants, who sent encrypted message, didn't have to deal with this problem. Other three received message from test monitor, that one of the members uses RSA keys and is unable to decrypt the message. Only one of those three was finally able to correct the failure.

Everybody who was able to sent encrypted email message was able to sign it as

well. Also all of them, who were able to decrypt the message, were able to verify the signature, because it is the part of decryption process. But to answer the question, whether they were aware that they are doing so, was not possible to find out from this test.

Another problematic thing to find out was whether users are aware of making backup of their revocation certificate. Only the three participants, who succeeded to send encrypted email and decrypt a reply, were asked to do so. None made it successfully.

Only three users concerned about the trustiness of the keys. One of them assumed, that when they are signed by campaign manager's key, they could be trusted. But none of them used validity and trust labeling in PGPkeys management interface.

6. Summary of PGP 5.0 user interface

Results of cognitive walkthrough and user test are supporting the theory, that usability of interface design of security software (in this case PGP 5.0) is not sufficiently user-friendly. Cognitive walkthrough revealed a lot of confusing aspects of graphical user interface of PGP 5.0. Furthermore user test pointed out, that only third of the participants was able to finish the task successfully in 90 minutes. And even thought it was not smooth. A lot of users accidentally revealed secret, which is what should not happen at all. So PGP 5.0 failed to offer sufficiently user-friendly interface and effective security to novices. These users were unable to understand the public key model. These attributes lead the user not to use this software at all.

7. Changes in PGP 9.0

The following case study [2] that was made several years after [1] tried to compare the older version PGP 5.0 with the newer version PGP 9.0. In Figure 4 you can see the graphical user interface of PGP Desktop 9.

A lot has been done with PGP since it's version 5.0. For example semi-automatic key creation and distribution and automatic email decryption functionalities were added. However underlying public and private key concept remained the same. PGP now advertises that it is the product also for first time users with no previous experiences. That's why case study [2] tried to perform small test with PGP 9.0 and Outlook Express. Several conclusions described below were found.

Users experienced problems with key verification and signing. Users didn't understand why it is important and how to do it.

Another confusing aspect is transparency of software's operations. User often assumed, that software will behave transparently, but no indication of encryption or signing is showed. Only after encrypted mail was sent, notification showed up. But if

the email is about to be sent unencrypted, user is not warned about that. Because of this nobody from test participants in [2] was able to encrypt. Transparency in decryption process may be another issue. Because users have no idea that decryption occurred, this can be susceptible to spoofing attack. As test revealed, two of five users were unable to identify legitimate emails, when comparing the key in the email to the key in PGP.

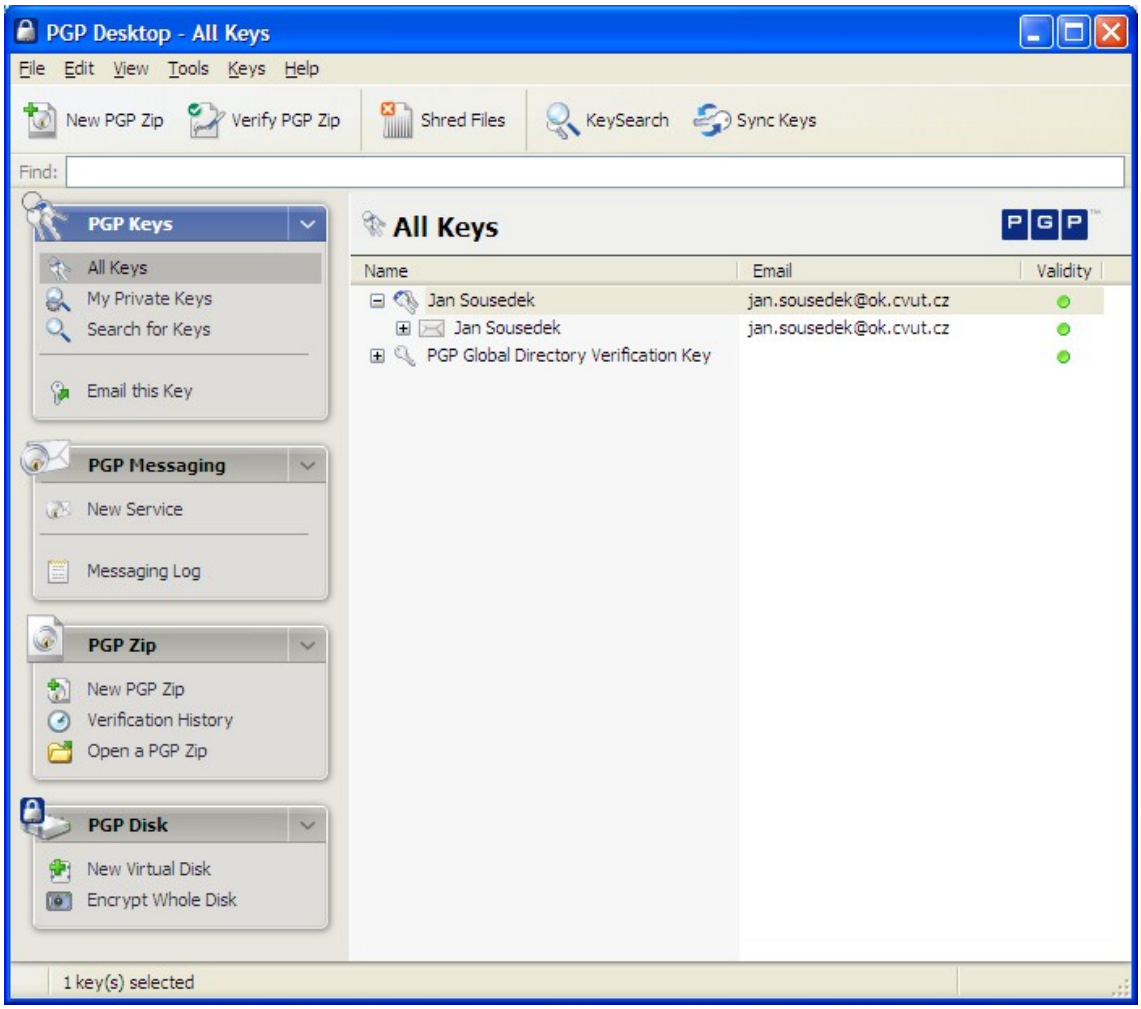

Figure 4

Digital signing is even more problematic then in PGP 5.0. None of test participants was able to perform this operation. This is because of lack of button to represent signing. This can be achieved only by right clicking on the PGP system tray icon.

Even though the process of creation of the keys was improved, sending and obtaining public keys is still problematic. Users noticed problems with 'Email this key' option that is available only after the key is selected.

8. Conclusion

Although the newer version of PGP is better then the previous PGP 5.0, it can be still improved for better usability. Transparency and easier use were added to PGP 9.0, but this may also have side affects marked before. Underlying problem of misunderstanding the whole concept of certification key model is still an issue. Despite the very interesting graphical user interface, users are very often unable to properly use it. These results confirm the theory, that when it comes to security software, special care should be paid when designing user interface.

9. References

[1] Alma Whitten and J. D. Tygar: Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0
[2] Steve Sheng, Levi Broderick, Colleen Alison Koranda, Jeremy J. Hyland: Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software
[3] Pretty Good Privacy, http://en.wikipedia.org/wiki/Pretty_Good_Privacy
[4] PGP Corporation, http://www.pgp.com