Bachelor thesis
Computing Science

Radboud University

Johnny can Encrypt? A Usability Study of IRMAseal

Author:
Niels Starren
s1020246

First supervisor/assessor:
Dr. Hanna Schraffenberger
hanna.schraffenberger@ru.nl

Second assessor:
Prof. Dr. Bart Jacobs
b.jacobs@cs.ru.nl

January 18, 2022

Abstract

This thesis presents the results of a comparison in usability between two secure email mechanisms, PGP and IRMAseal, and we present ideas for possible improvements on IRMAseal. We have performed a user interface analysis and usability study on both these secure email mechanisms in order to determine which one is more user-friendly. In the user interface analysis, we found that PGP still contains design flaws and that it gets outperformed by IRMAseal. In the usability study, nine participants attempted to encrypt and decrypt messages with IRMAseal, and to encrypt messages with PGP. Results from this study showed that where all nine participants were able to complete the tasks using IRMAseal, only five out the nine participants were able to do this using PGP. Furthermore, IRMAseal scored significant better than PGP with respect to the System Usability Scale scores of both, with a score of 85.3 for IRMAseal and 46.1 for PGP. We concluded that the usability of PGP has in increased in recent years, but it is still not usable enough for the masses. Overall, from the results of our user interface analysis and usability study, we concluded that IRMAseal is a more usable alternative in comparison to PGP.

Contents

Acknowledgment

Throughout the writing of this bachelor thesis, I received a lot of support. I would like to thank my supervisor, dr. Hanna Schraffenberger, for her support and effort during this project. In addition, I would like to thank Daniel Ostkamp, for providing feedback on my thesis, and all participants who have been participating in this study for their time and effort.

Chapter 1

Introduction

Cryptography, it has been around some time. It was in the Renaissance that cryptography began to be studied in depth and its techniques were recorded and taught. In addition to diplomacy and war, another important use of cryptography has been to protect love letters. Marie Antoinette used cryptography expertly and with good effect for her love life (Davies, 1997). Already back then, people wanted to prevent any third party from reading their communications.

Nowadays, a common way of communicating information between two parties is email. For securing those emails, end-to-end encryption can be used. End-to-end encryption ensures that two parties can communicate without a third party being able to access the communications. S/MIME (Secure/Multi-purpose Internet Mail Extensions) and OpenPGP (Pretty Good Privacy) are examples of mechanisms implementing end-to-end encryption. S/MIME is commonly used in corporations and governments. OpenPGP is used by the technological community and recommended to people working in high-risk environments (Schwenk et al., 2020). Furthermore, PGP has long been the primary IETF[1] standard for encrypting email. Although, it suffers from widespread usability and security problems that have limited its adoption (Halpin, 2020). Stransky et al. underline this adoption problem in their paper in which they analysed 81 million emails from 37 thousand email accounts through 27 years and found out that only 0.06% of these emails were encrypted. Multiple usability studies on PGP have been performed with regular email users trying to send encrypted emails and decrypt encrypted emails. Whitten and Tygar (1999) were the first who performed a usability study on PGP 5.0. They concluded that it was not usable enough to provide effective security for most computer users. Sheng et al. (2006) tried to understand the usability situation on PGP 9 and also found major usability issues. The most recent usability study on PGP has

[1] Internet Engineering Task Force, `https://www.ietf.org/`

been performed by Ruoti et al. (2015), but with no different results. All these studies identified key management as the main problem affecting the usability of PGP.

These usability issues of PGP were partially addressed with IRMAseal. IRMAseal is an encryption mechanism which aims to be easy to use by the masses. It is currently in development at the Radboud University by a development team consisting of, among others, Merel Brandon, Leon Botros and Daniel Ostkamp. The main usability problem of PGP, key management, is tried to be solved within IRMAseal by relying on a trusted third party for key generation and distribution. Together with a simple user interface, IRMAseal could be a more user-friendly alternative for PGP.

The main research question we are trying to answer in this thesis is as follows: Is IRMAseal a more usable alternative for email encryption in comparison to PGP? For answering this question, we will focus on the first step of using secure email, which is sending an encrypted email. However, we will also discuss the process of decrypting an encrypted email. Another interesting sub-question is: Has the usability of the current version of PGP improved in comparison to the earlier version, which has been analysed by Ruoti et al. in 2015?

We start this thesis with explaining some preliminaries and a technical comparison between PGP and IRMAseal in Chapter 2. Related work is discussed next in Chapter 3. Chapter 4 contains a user interface analysis on both user interfaces. In Chapter 5, we discuss the usability study we performed in this thesis where nine participants were asked to encrypt and decrypt emails using IRMAseal and encrypt emails using PGP, followed by a questionnaire and post-study interview. The results of the user interface analysis and the usability study are discussed in Chapter 6, where we also present possible improvements on IRMAseal. We conclude this thesis in Chapter 7.

The results of this thesis are going to be used as feedback in the further development of IRMAseal. Additionally, we hope that this thesis helps to contribute to a broader adoption of secure email by regular email users.

Chapter 2

Preliminaries

This thesis requires preliminary knowledge on the topics usability, asymmetric cryptography and IRMA, therefore we discuss these preliminaries in the following sections. Furthermore, we provide a technical comparison between IRMAseal and PGP in this chapter.

2.1 Usability

Since we will determine the usability of IRMAseal and PGP, we need some sort of definition for it. Quesenbery (2004) argued in her paper that "usability" has become a catch-phrase for a set of ideas about relationships between users, designer, developers and the software. She used the word "usability" as "the quality or characteristic of a product that meets the need of people who use it, allowing them to work – or play – with it for their own purposes and in a way that it is appropriate for them"(Quesenbery, 2004, p 1). In order to determine the usability of software, we need some qualities of software to measure. Quesenbery proposed these qualities as the 5Es and defined these as follows:

- **Effective**, this addresses whether the software helps users achieving their goals.
- **Efficient**, the speed with which work can be done.
- **Engaging**, how satisfying or interesting an interface is to use.
- **Error Tolerant**, includes how well users are prevented from making a mistake or how to recover from making one.
- **Easy to learn**, involves how well the product supports first uses and deeper learnings.

Depending on the goal of the product, weights on these qualities can be set. For example, we can argue that a Learning Management System should have

the focus on *Effective*, *Easy to Learn* and *Error Tolerance*, and a graphics program will be focussing on *Effective*, *Engaging* and *Easy to Learn*. In this thesis, we will adopt Quesenbery's view and definition of usability. We are using the 5Es for discussing and evaluating both email encryption systems where *Effective*, *Efficient*, *Error Tolerant* and *Easy to Learn* are the main focus.

2.2 System Usability Scale

Throughout this thesis, we are discussing the System Usability Scale (SUS) scores of PGP and IRMAseal. This scale has been introduced by Brooke (1995). It has been designed to assess the usability of a system in an easy way. The SUS is used after a participant had the opportunity to use a system. The participant is going to be asked to answer the questions about the system, which are displayed in Figure A.1 in the Appendix. The final score can be calculated as $(X' + Y') \cdot 2.5$ where $X' = X - 5$, $Y' = 25 - Y$, and X is the sum of the odd numbered questions and Y is the sum of the even numbered questions. The final score will be in range from 0 to 100. In 2009, Bangor et al. proposed to add an Adjective Rating Scale to the SUS Scores (Bangor et al., 2009). This scale has the following ratings: Worst Imaginable, Awful, Poor, Ok, Good, Excellent and Best Imaginable. Figure 2.1 shows the mapping between the SUS scores and adjectives.

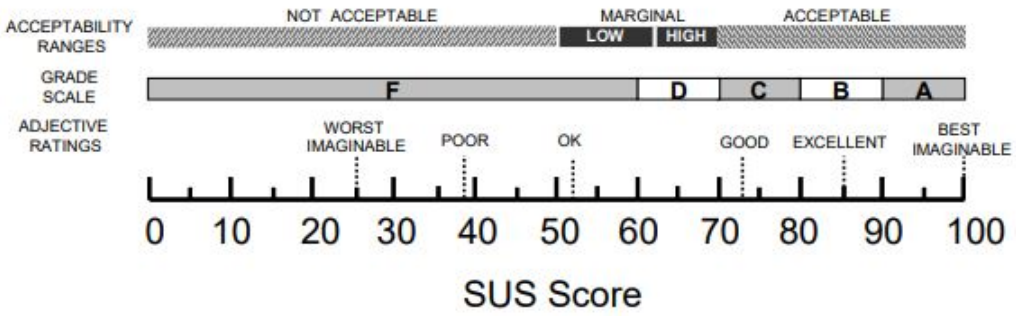

Figure 2.1: Relation between SUS scores and the adjective ratings (Bangor et al., 2009)

2.3 Asymmetric Cryptography

This thesis requires some basic knowledge about asymmetric cryptography, which we discuss in this section.

When Johnny and Jane want to send encrypted messages to each other,

both need to generate a key pair. This key pair consists of a private and a public key. The public key is used for encryption and is, just as its name suggests, publicly available. The private key is used for decrypting encrypted messages and message signing. As its name suggests, the private key is meant to be private and may not be shared. Whenever a private key leaks and an adversary has access to the communications, all security has been compromised. In Figure 2.2, a simple encryption scheme is shown. Johnny and Jane both have generated a key pair. Johnny encrypts the message using Jane's public key and sends the ciphertext to her. Only Jane will be able to decrypt the message using her private key.

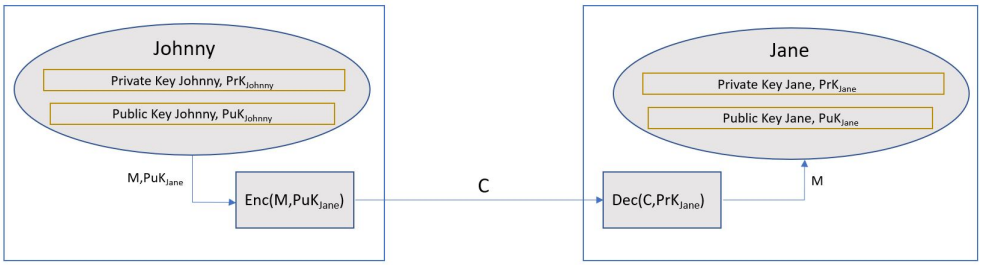

Figure 2.2: A simple asymmetric cryptography encryption scheme for Johnny communicating with Jane

2.4 IRMA

Another returning topic in this thesis is IRMA, which is at the heart of the IRMAseal encryption mechanism and used to authenticate the recipient of a with IRMAseal encrypted email. IRMA stands for I Reveal My Attributes[1] and is a mobile identity management app created by the Privacy by Design Foundation and SIDN (Privacy by Design Foundation, b). We can see it as a virtual identity wallet in which one can collect virtual identity cards, so called attributes. On the website of IRMA is explained that IRMA can be used for signing and authentication in a privacy-friendly way (Privacy by Design Foundation, a). Furthermore, there is explained that whenever Johnny needs to authenticate himself, he only reveals the necessary attribute. These attributes can be retrieved through an attribute issuer, for example banks, who can issue attributes like bank numbers, or national authorities, who can issue attributes as names, addresses, etc. In order for Johnny to obtain these attributes from an issuer, Johnny needs to authenticate himself to this instance. Let us assume Johnny wants to obtain the attribute containing his name and birth date, because he wants to be able to play roulette in an online casino, which happened to use IRMA for authentication. He needs to

[1] `https://privacybydesign.foundation/irma/`

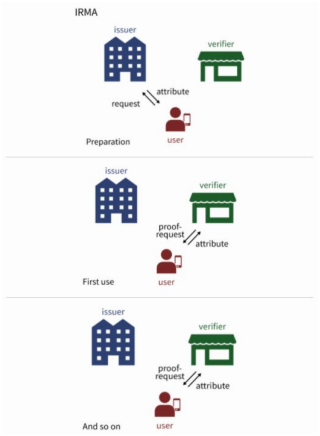

Figure 2.3: Overview of IRMA (Privacy by Design Foundation, a)

authenticate himself to the local authorities, who can look up the attributes known from Johnny. Johnny can now choose which attributes provided and signed by the issuer, he wants to download on his IRMA app. Once Johnny has downloaded the attributes, he can prove to the online casino that he is indeed Johnny and that he is over 18 years old, so he is eligible to play roulette. An overview of these steps can be found at Figure 2.3.

Similarly, as we saw for the online casino, IRMAseal also uses IRMA for authentication. Let us assume Johnny has received an email encrypted with IRMAseal. Only if Johnny can authenticate himself using IRMA, he will be able to read the contents of the email. In short, IRMA is used in IRMAseal for authentication of the recipient of a with IRMAseal encrypted email.

2.5 Technical comparison of PGP and IRMAseal

This section discusses the encryption schemes of PGP and IRMAseal. PGP and IRMAseal are encryption mechanisms used for providing secure email communications. Both mechanisms require keys for encrypting and decrypting messages. The main difference between PGP and IRMAseal is that PGP relies on manual key management and IRMAseal on automatic key management. We will discuss how manual and automatic key management are implemented in PGP and IRMAseal respectively.

2.5.1 Manual Key Management with PGP

As we already have seen, whenever Johnny wants to send encrypted messages to Jane, he needs his own key pair and Jane's public key. But how does he get Jane's key? And how does Johnny know if that key really belongs to

Jane? Here comes in the concept of Public Key Infrastructure.

Public Key Infrastructure

Public Key Infrastructure (PKI) is a solution for assuring authenticity of public keys via qualified digital certificates (Hableel et al., 2013). In more simple words, PKI ensures that public keys are authenticated to the person who claims to own a certain public key. For example, when Jane wants to let her public key certified, she needs to go to a Certificate Authority (CA). When Jane can prove to the CA she really owns her public key, she gets a public certificate. Now, when she sends her public key and certificate (via email) to Johnny, he can check the certificate and see if Jane's key matches the key in the certificate. When this is the case, Johnny knows that the public key is really Jane's.

In the previous example it may seem that there is only one Certificate Authority. This is actually an oversimplification. The most implementations of a PKI are hierarchical. In this trust model, the hierarchy consists of a series of CAs that are arranged based on a predetermined set of rules and conventions (Weise, 2001). At the top of this hierarchy there are the root CAs. The certificates of these CAs are self-signed, which means that people need to trust the root in order to trust the issued certificates lower in the hierarchy.

It can happen that a CA gets compromised. If this happens, as for example the DigiNotar hack in 2008 (Prins and Cybercrime, 2011), all issued certificates need to get revoked. These revocations are placed on a list called the Certificate Revocation List (CRL) (Weise, 2001). Furthermore, if Jane changes her public key before the expiration date of her former key, that certificate needs to get revoked. Hence, if Johnny wants to be really sure that he has the right public key of Jane, he should check besides her public key certificate the CRL to make sure that Jane's certificate has not been revoked.

PGP

Instead of a hierarchical trust architecture with Certification Authorities, PGP enables a certification model where any entity can verify another entity (Ulrich et al., 2011). This is called the Web of Trust (WoT). Ulrich et al. explain that the WoT is a user-centric and self-organized form of PKI where users 'issue certificates' to each other by signing another key with their private keys. As a result of this approach, events as Key-signing Parties are organized at conferences and meetings. Public key certificates are central to PGP. Each certificate contains the key owner's user ID, the public

key itself, a key ID and the creation date. Ulrich et al. (2011) explain that PGP uses a trust metric to allow users to assess the trust between the key and owner binding. They continue that there are two notions of trust in PGP: the trustworthiness of an introducer, referring to how much another user is trusted to apply care when identifying an identity, and the trustworthiness of a public-key certificate, which is the degree to which a user is sure whether the binding between the key and owner is correct. In more simple words, the trustworthiness of an introducer indicates whether another user can be trusted to verify an identity in a correct and precise way and the trustworthiness of a public-key certificate is the certainty in which a user believes the key-owner combination is correct.

Let us assume Johnny wants to send an encrypted email using PGP. He will be needing his own private key and Jane's public key. How can Johnny know if Jane's key really belongs to Jane, using the WoT? Suppose Johnny trusts Joannie and Joannie has signed Jane's key. Because Johnny trusts Joannie, he can trust that Jane's key really belongs to Jane.

Another option for Johnny to verify if Jane's key really belongs to Jane is to verify the key himself. Suppose Johnny has received Jane's public key via an email and he wants to verify if it is really Jane's public key. This requires Johnny to verify Jane's fingerprint via phone or a physical meeting, to be sure he is indeed communicating with Jane. When the fingerprints match, Johnny is sure he has indeed Jane's public key.

Once Johnny and Jane established, exchanged and verified their (public) keys, they are able to send encrypted emails to each other as discussed earlier (and shown in Figure 2.2).

2.5.2 Automatic Key Management with IRMAseal

Where Johnny needed to manage his own keys in order to encrypt an email with PGP, he does not need to when he uses IRMAseal. The exact workings are discussed in the following sections.

Identity-Based Encryption

IRMAseal makes use of Identity-Based Encryption (IBE). Identity-Based Encryption is an encryption scheme in which a user's public key can be an arbitrary string, for example an email address. Shamir (1984) introduced this scheme as an encryption scheme which enables any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, without keeping directories, and without using the services of a third party. Although Shamir's proposition did not make use of

a trusted third party, modern implementations do in the form of a Private Key Generator (PKG). When Johnny wants to send an encrypted email to Jane using IBE, he needs to obtain a master public key from the PKG. Jane's public key will be computed from the identifier of Jane combined with the master public key. Johnny can now encrypt the message with Jane's public key and send it to her. For Jane to decrypt Johnny's message, she needs to identify herself to the PKG. Once she did this, the PKG will respond with Jane's private key, enabling her to decrypt Johnny's message. An overview of these steps is shown in Figure 2.4.

The main advantage of IBE is that users do not have to manage keys. However, the use of a Trusted Third Party could also lead to dangerous situations. If the Trusted Third Party gets compromised in one way or another, all previous and upcoming communications are compromised (whenever an attacker gets access to the communications), since key management is taken care of by this party.

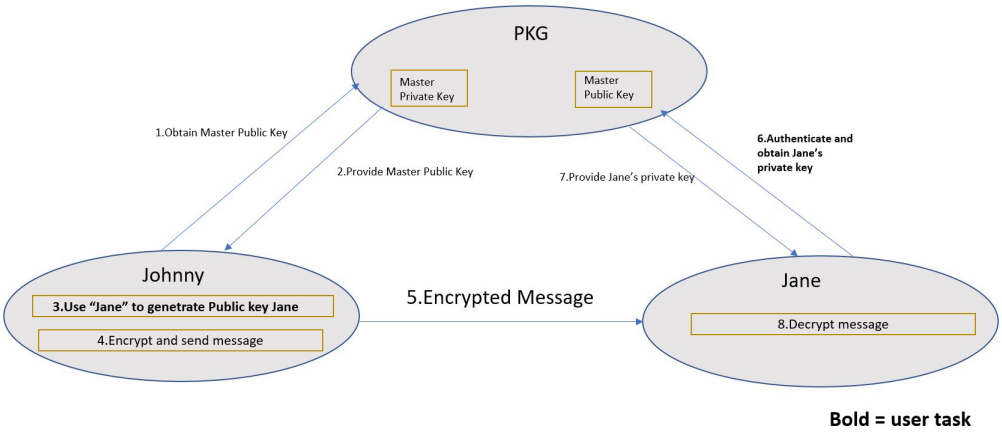

Figure 2.4: Johnny sending an encrypted message to Jane using Identity-Based Encryption.

IRMAseal

We already saw in the previous section about IBE that any arbitrary string can be used as a user's public key. Hence, also an attribute (or set of attributes) of an identity-platform like IRMA. When we are using IRMA attributes for encryption with IBE we get a scheme as can be seen in Figure 2.5. IRMAseal implements this scheme.

Let us assume Johnny wants to send an encrypted email to Jane using her email-address for the encryption process. He starts with retrieving the

master public key from the PKG. From the retrieved key and Jane's email address, Jane's public key can be derived and the message can be encrypted using this public key, before sending the encrypted email to her. In order for Jane to decrypt Johnny's message, she needs to retrieve her private key from the PKG. When Jane asks the PKG for her private key, the PKG will start a session with the IRMA server. At this stage, Jane has to authenticate herself to the IRMA server by disclosing the asked attribute. When this happens, the IRMA server communicates the session result to the PKG, who distributes Jane's private key to her. With this key, Jane is able to decrypt Johnny's email (Botros and Ostkamp).

When using IBE in combination with IRMA, Johnny knows that if he is using the right email address of Jane, only Jane will be able to read the email (or someone having access to Jane's email account). This is because Jane has to authenticate herself in order to retrieve her private key for decrypting the email. A downside of this approach is the use of trusted third parties, namely the Private Key Generator and the IRMA server. As mentioned for IBE earlier, if for some reason any of those services get compromised, all previous and upcoming communications are compromised (whenever an attacker gets access to the communications).

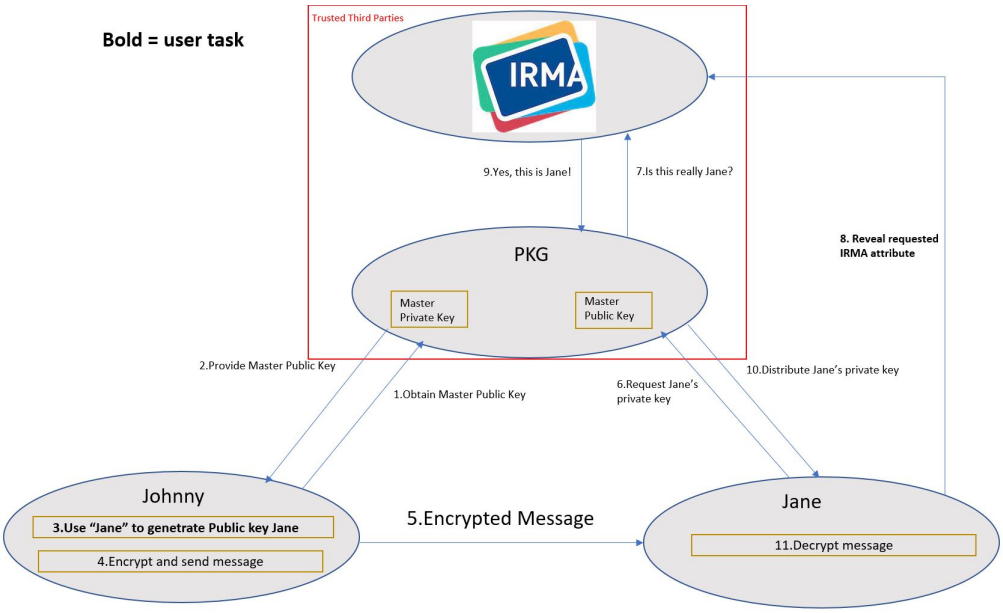

Figure 2.5: Identity-Based encryption with IRMA based on (Botros and Ostkamp)

Chapter 3

Related Work

As we already briefly mentioned in the introduction, there have been a lot of usability studies on email encryption systems. In this chapter, we will look at some of them. We do this in order to identify the problems found in email encryptions systems, focussing on PGP. Furthermore, we want to find inspiration on how to perform a usability study on email encryption tools. Lastly, we want to identify the current perception and usage of secure email.

The first usability study of PGP 5.0 has been performed by Whitten and Tygar (1999). They evaluated PGP through two methods: an informal cognitive walkthrough, a structured approach to evaluate the usability, in which they reviewed PGP's user interface and a user test performed with participants selected to be a reasonably representative population of general email users. From the cognitive walkthrough, Whitten and Tygar (1999) concluded that there were a number of user interface design flaws that may contribute to security failures. In the user test, participants were given a task description containing a secret message and a list of email addresses to send the secret message to using encrypted email. Completing this task required the participant to create a key pair, get the public keys of the recipients, make their own public key available to the recipients, type the secret into the email, sign the mail with their private key, encrypt the mail using the public keys of the recipients and sending the result. In case the participant succeeded in completing this task, they were sent an encrypted and signed email to test whether they could decrypt and read it successfully. Only two of the twelve participants were able to use PGP 5.0 to correctly sign and encrypt email messages within the given 90 minutes. Three participants exposed the secret they were meant to protect by sending these without encryption. However, they believed they had encrypted the email but, it turned out that they had not. Other problems encountered were for example failure in understanding the public key model, since seven out of the eleven participants who figured out how to encrypt used their own

public key to encrypt their email with, and participants being unable to decrypt the received encrypted message after completing the initial task. Overall, Whitten and Tygar (1999) concluded that the user interface design of PGP 5.0 is not sufficient to make computer security usable for the masses.

The work of Whitten and Tygar (1999) gave rise to a lot of attention to usability in email encryption, and much work has been conducted since. In the next paragraphs, we will look at some of these.

Six years after, as a follow-up on Whitten and Tygar (1999), Garfinkel and Miller (2005) published "Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express". In this paper, they argued that the usability problems in the original *Johnny* user study were not driven by PGP 5.0, but by the underlying key certificate management. In *Johnny 2*, Garfinkel and Miller (2005) wanted to replicate *Johnny* as closely as possible, except using a system that implements Key Continuity Management (KCM). KCM automates key management, which means the users do not have to manage their own keys. Garfinkel and Miller explain that with KCM, an application would remember the key given by an entity when communicating for the first time in order to check if that key still remains the same when they communicate another time. When the keys matches, it is safe to assume that we are communicating with the same entity as last time. In the *Johnny 2* experiment, participants were asked to send encrypted emails with a program called CoPilot that implements KCM. The results of the user test showed Garfinkel and Miller that the participants comprehended both the task they were given and the tools they needed to use. They concluded that KCM could improve security, but it is not the panacea to the email security problem, since it does not provide tools for users to decide if a new key is trustable or not. But this problem also exists in PKI-based systems, according to Garfinkel and Miller.

In 2006, when PGP had reached version 9, Sheng et al. (2006) sought to understand the usability situation of PGP in their paper "Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software". They designed a pilot study to find 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. In their pilot study, six novice users used PGP 9 and Outlook Express 6.0. Sheng et al. found that none of the six users were able to encrypt an email. Furthermore, they found some major problems in PGP 9. For example, key verification and signing is still lacking, since users had issues in verifying and signing keys. Also, users found it difficult to determine if the software operated as requested, since notification of successful encryption only happened after sending the email. Additionally, sending an

unencrypted email does not trigger a notification at all. Lastly, they found that signing messages is more problematic in PGP 9 than PGP 5, because there isn't a dedicated button for it. They summarised that in comparison to Whitten and Tygar (1999), PGP 9 improved on automatically encrypting emails, but key certification (i.e., verification) is the main issue, since the interface does not provide enough cues or feedback to the user.

In 2015, Ruoti et al. (2015) performed a usability study on modern PGP tools. They used Mailvelope, a browser extension that integrates with a user's webmail, as a PGP tool to send encrypted emails with. In the study, twenty participants were grouped into ten pairs who attempted to exchange encrypted emails. The scenario involved sending sensitive information of participant A to participant B using Mailvelope. When participant B received the email, he or she responded with an (encrypted) email containing a confirmation code to confirm that the email had been received correctly. Immediately after completing the task, participants were asked to fill in the SUS questions and were asked several questions related to their experience with the system. Following on the survey, participants were interviewed by the study coordinators who focussed on issues arisen during the study. The results from the study were according to Ruoti et al. disheartening. From the 10 pairs of participants who attempted to exchange encrypted email, only one pair succeeded to send an encrypted email, decrypt that email and respond with a confirmation code. All others were not able to complete the task with various reasons. The most common mistake was the sender trying to encrypt the email with their own public key. Others vary from modifying the PGP block after encryption to sending along the private key with the password. Two pairs were not able to read the encrypted message because they were unaware that Mailvelope was required to read the message. Ruoti et al. concluded that PGP is still unusable by the masses, with a SUS score of 34.5 which is rated as "Poor" usability. They proposed some potential improvements based on the post-study questionnaires and interviews. First, integrated tutorials could help first-time users. Furthermore, explaining public key cryptography in an easy way could help users with using PGP for the first time.

Ruoti et al. (2018) performed a usability study on three different key management schemes: passwords, public key directory (PKD, a trusted directory where users can submit their public keys) and identity-based encryption. Pairs of participants used these three secure email systems to communicate sensitive information to each other. When the tasks for one system were completed, the participants were asked to answer questions about their experience with the system used to perform the task. The questions consisted of the SUS questions and some questions where the participants needed to describe what they liked most and what not, and if they would apply changes

to the system. After the participants completed the tasks for all three secure email systems, they had to indicate which system was their favourite and explain why they liked it. Furthermore, the participants were asked to answer the following two statements; "I want to be able to encrypt my email" and "I would encrypt my email frequently". They could answer with Strongly Disagree, Disagree, Neutral, Agree or Strongly Agree. In the post-study interview, participants were asked about their general impressions of the study and the secure email systems. Furthermore, the events where participants struggled with while performing the task were discussed. Lastly, the participants were asked to explain how the security models of the three systems work and what an attacker needed to do in order to be able to read the encrypted messages. When all participants knew how the security models of the three systems worked, they had the chance to revise their opinion on their favourite system. Ruoti et al. found that the results of the System Usability Scale score were slightly in the advantage of IBE. Where IBE had a score of 77.3, the runner-up, PKD, had a score of 75.7. Furthermore, they found that all participants understood password-based encryption and only a few the security models of PKD and IBE. The favourite system among the participants turned out to be IBE, with a slight advantage. Lastly, Ruoti et al. found that most participants want to be able to encrypt their email, but they are not sure how often they would use email encryption in practice.

In the following papers, we will look at the perception on and the usage of secure email. From these papers, we want to identify the current usage of secure email and the arguments for using secure email or not using secure email. Ultimately, we want to identify potential design strategies that can help or hinder the adoption of IRMAseal.

Renaud et al. (2014) carried out a qualitative study in order to identify users' mental models of email security and find out why end-to-end encryption is used so limited in their paper "Why doesn't Jane protect her privacy?". They performed a study consisting of semi-structured interviews and a subsequent qualitative analysis. Renaud et al. used the quantitative analysis for determining the reasons behind participants not using end-to-end encryption. From the analysis of the interviews in combination with verification from literature, Renaud et al. could confirm that a lack of concern of privacy violation, misconceptions of how to protect email, no perceived need to take action and not being able to end-to-end encrypt emails are explanations of the limited use of end-to-end encryption (Renaud et al., 2014).

Braun and Oostveen (2019) analysed in their work "Encryption for the masses? An analysis of PGP key usage" 4.27 million PGP public keys generated between 1991 and May 2016. From the analysis of those 4.27 million keys, complemented by a survey filled out by former and current PGP users,

Braun and Oostveen estimated that PGP might draw 550,000 active users at the moment their paper was published. Furthermore, from the answers of the survey, Braun and Oostveen found that 71.2% chose PGP out of curiosity and 31.2% as a response to government activities. The vast majority of former PGP users gave up on PGP because they had no one else to communicate with, while 25.3% had no need to encrypt information, 23.8% felt it lacked an intuitive software interface, or had no PGP availability on their other platforms (Braun and Oostveen, 2019).

In July 2021, Stransky et al. (2022) published their paper where they analysed metadata for 81,612,595 emails from 37,089 email accounts through 27 years at a large university in order to find out at what scale secure email is used through these years. They developed a privacy-friendly data collection pipeline in order to analyse large amounts of email data, with the focus on S/MIME and PGP usage. From the analysis of 81 million emails it turned out that only 2.8% were digitally signed and even less, 0.06% were encrypted. Stransky et al. (2022) concluded from this research that although the usage of email has been grown exponentially, the fraction of encrypted emails remained consistently small.

The goal of this section was to identify open problems found in email encryption systems, to find inspiration on how to perform a usability study on email encryption tools and to identify the current perception and usage of secure email. As we have seen in Whitten and Tygar (1999), Garfinkel and Miller (2005), Sheng et al. (2006) and Ruoti et al. (2015), easy to use key management is still the largest obstacle in usable secure email. Furthermore, we have seen some examples on how to perform a usability study on secure email systems. These often involve a sender trying to communicate with a recipient using encrypted email systems, followed by answering the SUS questions and a post-study interview. From the results of these studies we also noticed that regularly, participants were not able to send an encrypted email, resulting in not being able to decrypt an encrypted message. For this reason, we will focus on sending an encrypted email as a main task and decrypting an encrypted email will be considered an optional follow-up task. Lastly, we have seen in the papers from Stransky et al. (2022), Ruoti et al. (2018) and Braun and Oostveen (2019) that encrypted email is rarely used for different reasons. However, according to Ruoti et al. (2018), people want to be able to encrypt their email messages if necessary, but they are not able to. When we consider the reasons for giving up on PGP by Braun and Oostveen (2019) and take these into perspective of IRMAseal, availability on multiple platforms, such as mobile devices could help in its adoption. Additionally, we found that a lack of intuitive software interface hindered the adoption as well. This indicates that in order to help the adoption of IRMAseal, the user interface should be intuitive for the user. Although, Braun

and Oostveen (2019) concluded that a vast majority gave up on PGP with the reason that they had no one else to communicate with, this problem is likely resolved by the fact that IRMAseal does not require the recipient to have any keys set up before the communications.

Clearly, more work is needed to increase the usability of secure email systems and the adoption of these. In this thesis, we want to find out if IRMAseal could be a more usable secure email mechanism in comparison to PGP. Additionally, we will be answering the question if the usability of the current PGP version has been improved in comparison to the earlier version, which has been analysed by Ruoti et al. (2015).

Chapter 4

User Interface Analysis

This chapter contains an analysis of both user interfaces of PGP and IRMAseal. The analysis is going to be carried out by means of an inspection method, namely a cognitive walkthrough, which Whitten and Tygar (1999) also performed in their seminal *Johnny* paper, and Nielsen's heuristics. In the past, many different versions of these techniques have been proposed. In this thesis, we use the cognitive walkthrough proposed by Polson et al. (1992) and the heuristics proposed by Nielsen (1994). Differently from Whitten and Tygar (1999), we combine the cognitive walkthrough with the heuristics. This means that we use the cognitive walkthrough to identify issues, and these will be categorised according to the heuristics in order to understand the issues at hand. Whenever we find design choices adhering to these heuristics, we will elaborate on these as well. Nielsen's heuristics also cover error prevention and recovery. Since these aspects are not all covered in the regular cognitive walkthrough, we introduce the "error walkthrough" where we will move through the system and try to make mistakes where we expect users to make them in order to find out how well the user can recover from these mistakes. Issues arising from this error walkthrough and design choices adhering to Nielsen's heuristics will be categorised according to these heuristics.

First, we explain the cognitive walkthrough of Polson et al. (1992) and the heuristics of Nielsen (1994) followed by applying those concepts to the user interfaces of PGP in Thunderbird[1] and the clickable mock-up of IRMAseal in Outlook[2]. Lastly, we are going to perform the earlier explained error walkthrough on both systems.

[1] `https://www.thunderbird.net/nl/`

[2] `https://www.microsoft.com/nl-nl/microsoft-365/outlook/email-and-calendar-software-microsoft-outlook`

4.1 Cognitive Walkthrough

Polson et al. (1992) proposed their version of the cognitive walkthrough in 1992. It was a new methodology for performing evaluations of user interface designs. The method involves a simulation, performed by a group of reviewers, of the cognitive activities of a user, to ensure that the user can easily learn to perform tasks that the system is intended to support (Polson et al., 1992). They explained that during the walkthrough, the reviewers step through the actions, considering the behaviour of the interface and its effect on the user, and attempting to identify those actions that would be difficult for the average member of the proposed user population to choose or to execute. Polson et al. describes the cognitive walkthrough as two phases: the preparation and the evaluation. In the preparation phase, the reviewers select a set of tasks that are a representative sample of tasks the application supports. Polson et al. continue that for each task in this set, the initial state needs to be described, followed by the sequence of actions used to accomplish the task, and the user's initial goals. Polson et al. explain that in the evaluation phase, the interaction between the user and the interface is analysed in depth. The reviewer looks at three different things: he looks at each user action to determine what goals the user should have leading up to the action, whether the prompts and labels of the interface will induce the user to take the correct action and how the user's goals will change in response to the feedback from the interface after the action is performed.

During our cognitive walkthrough, we are checking for these conditions. Based on our own interpretation, we reformulated these conditions into the following yes/no questions:

1. Does the user have the right goal in mind?
2. Do the labels and prompts of the interface induce the user to take the correct action?
3. Is the user able to see whether his previous action brought him closer to his goal?

Although, Polson et al. state that the cognitive walkthrough is performed by a group of reviewers, we do not have the resources and time to satisfy these requirements, and we will focus on our own findings.

4.2 Heuristics from Nielsen

Nielsen's heuristics originated from the heuristic evaluation proposed by Nielsen and Molich (1990). This is an informal method of usability analysis where a number of evaluators are presented with an interface design

and asked to comment on it (Nielsen and Molich, 1990). In their paper "Heuristic Evaluation of User Interfaces", Nielsen and Molich explain that ideally people would conduct such evaluations according to certain rules, for example like in guidelines. However, back in 1990 when their paper was published, those guidelines consisted over thousand of those rules. This was seen as intimidating by developers. Nielsen and Molich consequently tried to cut the complexity of the rule base by relying on a small set of heuristics. These heuristics are *the nine usability heuristics*, Molich and Nielsen discussed in their paper "Improving a Human-Computer Dialogue". During the past decades, some changes have been made to the original heuristics. Where the original paper introduced nine heuristics, currently there are ten (Nielsen, 1994). We summarize and give an illustration of each of the ten heuristics in an email context:

1. **Visibility of System Status**
   The system should always keep its user informed on its current status. For example, showing a progress bar while sending an email.

2. **Match between System and the Real World**
   The system should use a language understandable by its users (in words or pictures). For example, when you want to create a new email, one can use the metaphor of a physical letter, and use a pen, envelope or letter icon for sending new mails.

3. **User Control and Freedom**
   At any stage of the system, users should be able to cancel unwanted actions. For example, cancelling an email sent by mistake.

4. **Consistency and Standards**
   Users of a system should know that the same icons perform the same actions. For example, clicking the cross in the right upper corner of a program will close any program.

5. **Error Prevention**
   The system should prevent a user from making an error. For example, giving a warning before deleting a user's own key pair.

6. **Recognition Rather Than Recall**
   The user should recognize elements in a system rather than having to remember them. For example, recognition that the recipient's email address needs to be filled in into the 'To' field is easy, but recalling that it is the second text-field is hard.

7. **Flexibility and Efficiency of Use**
   Users should be able to carry out a process in different ways, so that they are able to choose the one they like most. For example, sending

an email can be done via clicking on the 'Send' button or via CTRL + Enter.

8. **Aesthetic and Minimalist Design**
   User interfaces should only contain necessary information helping the user completing their goals.

9. **Recognize, Diagnose, and Recover from Errors**
   The system should present clear and suggestive error messages to the user. For example, notifying when a user wants to send an email without a subject.

10. **Help and Documentation**
    Users should be able to easily search through the helpers guide to find help in completing their task. For example, on a support page.

These ten heuristics from Nielsen (1994) will be used to categorise the issues we find during the cognitive walkthrough in order to be able to understand the issues at hand. Whenever we find design choices adhering to these heuristics, we elaborate on these as well.

4.3 User interface analysis of PGP in Thunderbird

In this section, we are going to perform the user interface analysis of PGP in Thunderbird by means of a cognitive walkthrough in combination with Nielsen's heuristics and the earlier explained error walkthrough.

Scenario

Suppose Jane has asked Johnny to send her his Social Security Number (SSN) via email. She proposes to do this using PGP for better security and already has sent her public key attached to the email. We will perform a cognitive walkthrough on the encryption of the email by Johnny. Our starting point is Johnny, who has just read Jane's email and went to the Thunderbird home screen.

4.3.1 Encrypting an email

From this point we will start the cognitive walkthrough. For Johnny to send his SSN using PGP to Jane, he has to do the following steps:

1. Learn about using PGP on the support page
2. Create his key pair
3. Add Jane's public key

4. Verify Jane's public key
5. Compose the email
6. Enable encryption
7. Send the email

We will take a closer look at each of these steps and analyse the actions needed to perform to complete these steps. Furthermore, we determine if any of the three questions from the cognitive walkthrough are answered with "No" while performing these actions and whenever this is the case, we categorise this problem according to the heuristics of Nielsen (1994). Whenever we find design choices adhering to these heuristics, we will elaborate on these as well.

Learn about using PGP on the support page

Since Johnny is a first time user and has no knowledge about PGP, he needs to go to the support page of Mozilla Thunderbird[3]. This takes the following steps:

1. Search for "PGP in Thunderbird" in a search engine.
2. Click on the Mozilla support link from in the results.
3. Click on the link "I have never used OpenPGP with Thunderbird before: How do I setup OpenPGP?" in the table of contents.
4. Read the contents of this page.

When we take a look at these steps, we can argue that Johnny never will have the goal "Read the user manual" when trying to send an encrypted email, hence question 1 from the cognitive walkthrough will be answered with "No". We can categorise this issue in the category *Consistency and Standards*, because normally when Johnny uses a new program for the first time, he expects himself to be able to use it without looking up the programs' support page. The fact that Johnny can find help easily on this support page is a plus in the category *Help and Documentation*.

Create a key pair

Ideally, Johnny knows from the support page that he has to create a key pair for himself. Creating a key pair requires the following steps:

1. Click on the hamburger icon at the right top of the screen

[3] `https://support.mozilla.org/en-US/kb/openpgp-thunderbird-howto-and-faq#w_i-have-never-used-openpgp-with-thunderbird-before-how-do-i-setup-openpgp`

2. Go to “Account Settings”
3. Move to the “End-to-End Encryption” tab in the “Account Settings”
4. Click on the “Add Key” button
5. Select the option “Create a new OpenPGP Key” and press ‘Continue’ in the newly opened window
6. Click on the “Generate Key” button
7. Click on “Confirm”

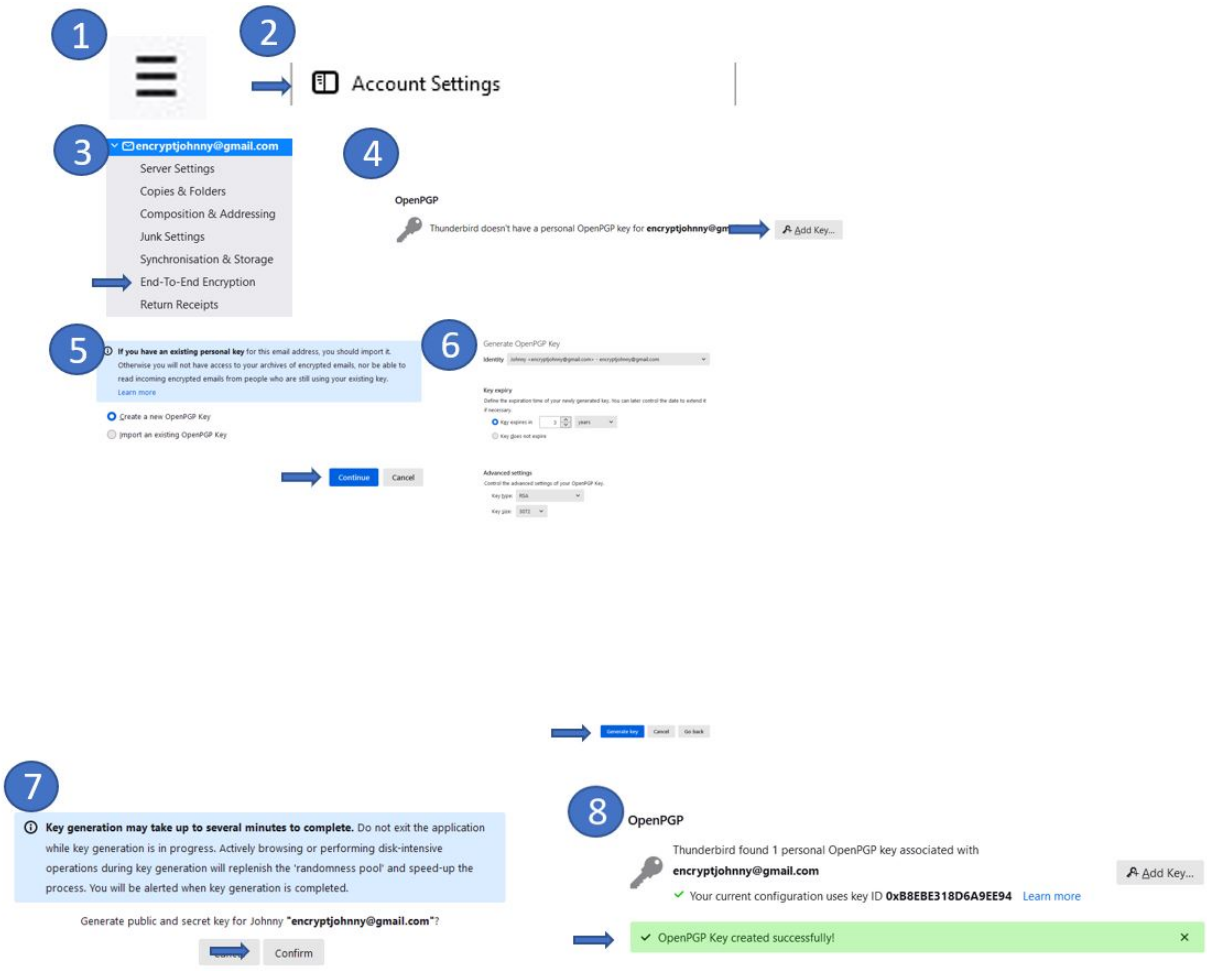

Figure 4.1: Flow of creating a key in PGP

An overview of these steps is shown in Figure 4.1. In step 6, Johnny can choose what type of key he wants to generate. However, Johnny just wants

to create *any* key, which he can use to send an encrypted email. He has no knowledge on the existence of different kinds of keys and needs to guess if the preset settings are correct. The question if the goal of the user matches the goal of the system has to be answered with "No". We can categorise this as an issue in the category *Match between System and the Real World*.

Add Jane's public key

The next step is adding and verifying Jane's public key. Adding Jane's key requires the following steps:

1. Download Jane's key from her email
2. Click on the hamburger icon at the right top of the screen
3. Go to "Account Settings"
4. Move to the "End-to-End Encryption" tab in the "Account Settings"
5. Click on "OpenPGP Key Manager"
6. In the Key Manager, click on 'File'
7. Click on "Import Public Key From File"
8. Select and upload Jane's key from the File Manager.
9. Click on "OK"
10. Click on "OK"

An overview of these steps is shown in Figure 4.2.
In step 6, Johnny wants to add Jane's key to the Key Manager. However, he would expect some sort of an "add" option instead of the "File" button which is supposed to be clicked. We need to answer question 2 of the cognitive walkthrough with "No". We can categorise this issue in the category *Match between System and the Real World*. The next issue can be found in step 9 (step 7 in Figure 4.2). The option "Not accepted (unverified)" is preset, however, Johnny probably has no idea on what this means and will click on "OK" without knowing what this will do. The goal of Johnny does not match the goal of the system, hence question 1 will be answered with "No". We can categorise this issue in the category *Match between System and the Real World*.

There also exists a shortcut for importing the attached key into the Key Manager. This shortcut can be seen in Figure 4.3. Once Johnny right clicks the attachment, the "Import OpenPGP Key" option appears which brings Johnny straight to step 7 from the regular *recipe*. This shortcut is a plus in the category *Flexibility and Efficiency of Use*.

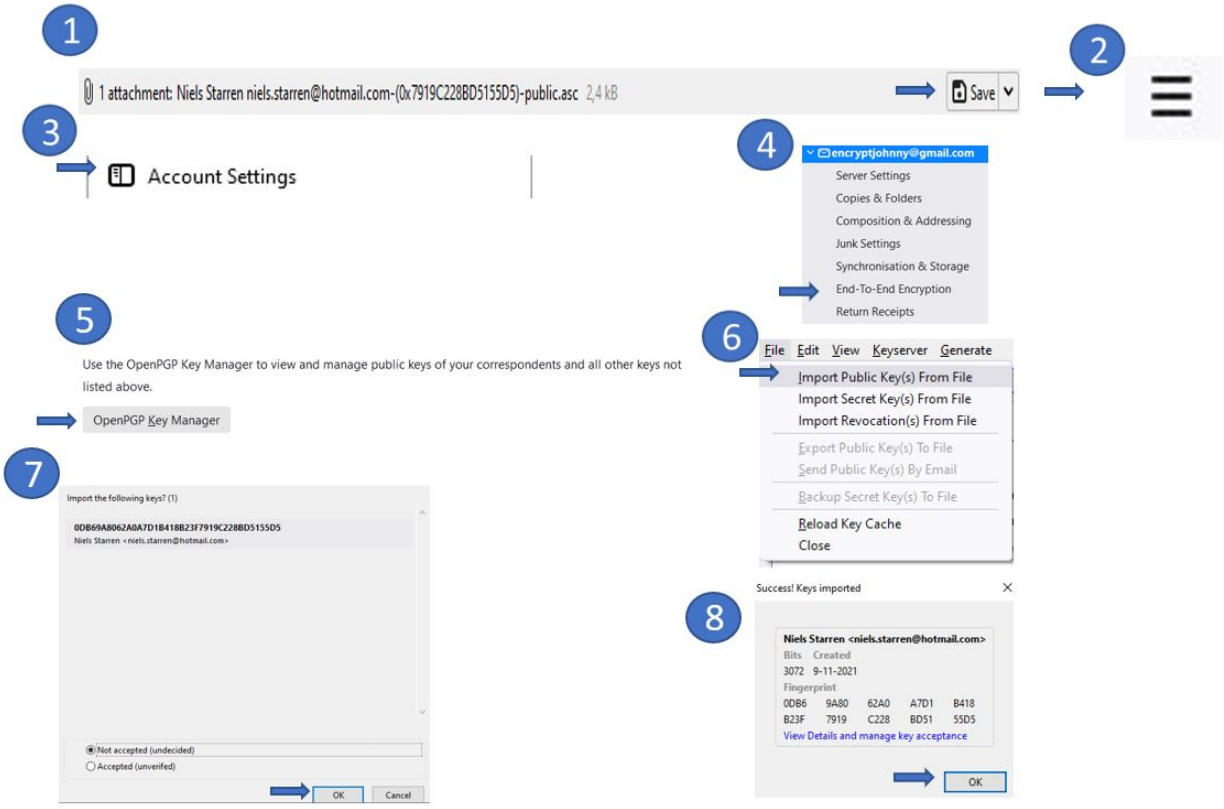

Figure 4.2: Flow of uploading Jane's public key

Figure 4.3: Shortcut for importing an public key attached to an email

Verify Jane's public key

Verification of Jane's key takes the following steps:

1. Get in contact with Jane (face-to-face or via telephone)
2. Check if the fingerprint of the downloaded public key matches Jane's public key fingerprint
3. Click on the hamburger icon at the right top of the screen
4. Go to "Account Settings"

5. Move to the “End-to-End Encryption” tab in the “Account Settings”
6. Click on “OpenPGP Key Manager”
7. Select Jane’s key
8. Click on “Yes, I’ve verified this in person this key has the correct fingerprint”
9. Click on “OK”

In Figure 4.4 we can see an overview of these steps.

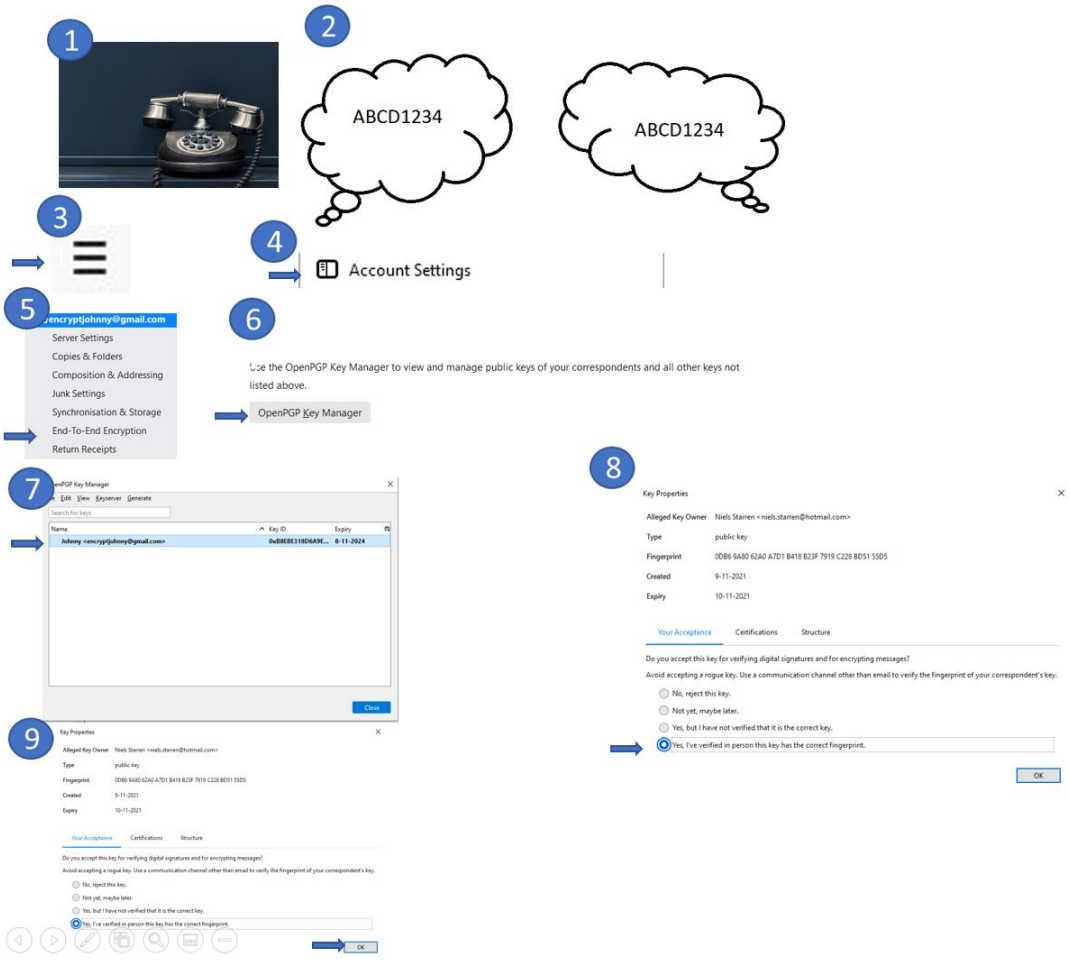

Figure 4.4: Flow of verifying Jane’s public key

The biggest concern in this step is whether Johnny actually knows he has to verify the authenticity of Jane’s key in the first place and how he has to do this in a proper way. Hence, the goal of Johnny does not match the

goal of the system which means question 1 will be answered with "No". This issue can be categorised as an issue in the category *Recognition Rather Than Recall*, because the system should let Johnny recognise in some way that he has to verify Jane's key instead of letting him recalling it. Whenever Johnny clicks on "OK" in step 9, the window closes but there is no confirmation that the settings were saved successfully. Johnny does not know if this action brought him closer to his goal, so question 3 will be answered with "No". We can categorise this as an issue in *Visibility of System Status*.

Compose the email

Writing the email takes the following steps:

1. Click on the "Write" button
2. Fill in Jane's email address
3. Fill in the subject
4. Fill in the body of the email

These steps are the same as sending an unencrypted email and do not contain any issues.

Enable encryption

When the email is written, Johnny needs to enable encryption. Enabling encryption takes the following steps:

1. Click on the "Security" button.
2. Click on "Require Encryption"

The Figure below shows these steps in image-form. In step 1, the lock icon

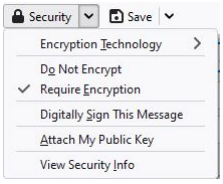

Figure 4.5: Enabling encryption in PGP

next to the "Security" button is explanatory, however, Johnny would expect an "Encryption" button, since this is used for the rest of the system. This means that the label does not induce Johnny to take the correct action, indicating that we need to answer question 2 from the walkthrough with "No". We can categorise this as an issue in the category *Consistency and*

Standards. Furthermore, in step 2, when encryption is enabled (and the "Security" tab has been closed), only a small icon in the lower right corner (shown in Figure 4.6) indicates that encryption is enabled. This could be more emphasised, which means that question 3 of the walkthrough needs to be answered with "No", because Johnny will probably not be able to see whether he has made progress towards his goal. We can categorise this as an issue in the category *Visibility of System Status*. Additionally, when encryption is getting enabled, the system automatically signs the message without explicitly informing Johnny. In this case, the goal of the user does not match the goal of the system, indicating that question 1 of the cognitive walkthrough needs to be answered with "No". Furthermore, Johnny is not notified clear enough about singing turning on automatically, indicating that Johnny is not able to see whether his action brought him closer to his goal. This means that we need to answer question 3 from the walkthrough with "No" as well. These two issues can be categorised as issues in the categories *Match between System and the Real World* and *Visibility of System Status* respectively.

Figure 4.6: Indicator that encryption is enabled in PGP (L) and when encryption and signing is enabled (R)

Send the email

When everything is setup, the mail has been written and encryption has been enabled, the email is ready to be send to Jane. All Johnny has to do for this is clicking on the 'Send' button. When he did this, a window with a progress bar opens. When the bar has been filled completely, the email has been sent, and Johnny returns to the Thunderbird home screen.

4.3.2 Decrypting an email

Assuming Johnny has loaded his key pair in Thunderbird, decrypting an encrypted email is easy. Once Johnny has received an encrypted email in his mailbox, it looks like an "empty" email as shown in Figure 4.7. Whenever Johnny wants to read the email, he has to click on it and the system will automatically decrypt the email for him.

Figure 4.7: Encrypted email (upper) and regular email (lower)

4.3.3 Error Walkthrough

Now, we will perform another walkthrough on the system, but this time we will be trying to make mistakes where they can be expected based on the regular cognitive walkthrough in order to see how the system reacts to this behaviour and how well the user can recover from making these mistakes. Again, we use Nielsen's heuristics to (better) understand the issues at hand.

Not looking up information on the usage of PGP
Let's assume that Johnny skips the entire first step of looking up information on how to use PGP for the first time. We expect that Johnny will not be able to figure out how to encrypt with PGP by himself. Figuring out all steps taken in the cognitive walkthrough without any knowledge of secure email will be quite a challenge, which we expect Johnny will not be able to complete. We can categorise this as an issue in the category *Recognize, Diagnose, and Recover From Errors*.

Trying to encrypt an email without a key pair
Suppose Johnny wants to send an encrypted email to Jane, but he has not created a key pair. When Johnny has composed his email and clicks on the security tab, he sees that the "Enable Encryption" button is not clickable. Unfortunately, the system does not tell Johnny what is happening and why he is not able to encrypt his email. We can classify this as a problem in *Visibility of System Status*. On the other hand, making the encryption button not clickable is a plus in the category *Error Prevention*.

Trying to send an encrypted email to someone without knowing their public key
When Johnny has uploaded his key pair and tries to send an encrypted email to Jane, whose public key is not stored in Johnny's Key Manager, the system will throw an error message saying the email cannot be encrypted since there are issues with the keys of the recipient. When Johnny closes this message, another window opens with more specific feedback on what the problem is, in this case the system shows that there is no key available for the recipient. This is a plus in the category *Recognize, Diagnose and Recover from Errors*.

Trying to send an encrypted email to an unaccepted public key
Just as in the previous two scenarios, when Johnny wants to send an encrypted email to a not accepted key, the system will tell Johnny that there is a problem with the key and on the next window it shows that the key needs to be accepted. This is a plus in the category *Recognize, Diagnose, and Recover from Errors*.

Deleting a key pair

Now, let us assume Johnny tries to delete his key pair from the Key Manager, a warning occurs telling that deleting a key pair is irreversible, and that upcoming messages encrypted with that key will become unreadable. Furthermore, the system gives the option to continue deleting the key pair or to cancel the process, which is a good implementation of *Error Prevention*. This message is shown in Figure 4.8.

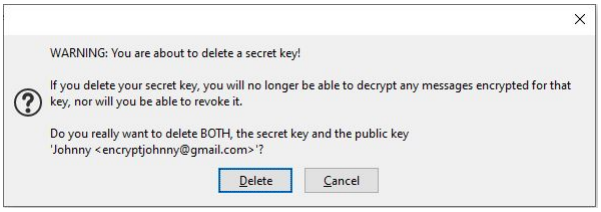

Figure 4.8: Warning message when trying to delete a secret key

Sending to a deleted key pair/ receiving an encrypted email on a deleted key pair

Let us assume Johnny has deleted his key pair for some reason and created a new one. Unfortunately, he did not communicate this with Jane, who sent him an encrypted email but used Johnny's old public key. He will not be able to read the message and instead, he sees an error message containing the text "The secret key that is required to decrypt this message is not available" on his screen as shown in Figure 4.9. Now, Johnny has to contact Jane and tell her he has a new public key and that he was not able to read her previous message. Every time Johnny deletes his key pair, he should inform his contacts about it. On the other hand, Jane should not be allowed to send an encrypted email using a deleted public key from the recipient. This is an issue in the category *Error Prevention*.

Figure 4.9: Error message when receiving an encrypted email on a deleted key pair

Cancelling the sending process of an email

Suppose Johnny has already clicked on the send button, but quickly realises he has forgotten to put something in the body of the (encrypted) mail and

clicks on the cancel button. When he does this, a window appears saying that the message was sent anyway (but it was encrypted), this can be categorised as an issue in the category *User Control and Freedom*.

We have seen a lot of usability issues in our analysis of PGP, just as some design choices adhering to Nielsen's heuristics. Furthermore, we categorised these according to the heuristics. In Appendix A.1, an overview is shown of all the found issues and plus points of the user interface of PGP, categorised according to the heuristics.

4.4 User interface analysis of IRMAseal in Outlook

In this section, we will perform the user interface analysis of IRMAseal in Outlook. Currently, there are two versions of IRMAseal, a working prototype in Thunderbird and a clickable mock-up designed by Merel Brandon. We will analyse the mock-up from Merel Brandon with one exception: the downloading step is based on a download site created solely for the usability study.

Scenario

Again, assume Jane asks Johnny to send her his SSN. However, this time, Johnny is asked to use IRMAseal to protect his sensitive information. We will perform a cognitive walkthrough for the encyption of the email by Johnny and the decryption of the email by Jane. Our starting point is Johnny, who has just read Jane's email and went to the Outlook home screen.

4.4.1 Encrypting an email

In order for Johnny to send an encrypted email using IRMAseal he has to do the following steps:

1. Download and install IRMAseal
2. Compose the email

* Check if encryption is enabled (It is enabled by default)

3. Select the required attributes for Jane to show
4. Fill in the information
5. Send the email

These steps will be analysed during the cognitive walkthrough. Whenever we need to answer one of the questions from the cognitive walkthrough with "No", we will further elaborate on it and categorise the issue according to the heuristics from Nielsen.

Download and install IRMAseal

Downloading IRMAseal takes the following steps:

1. Open a web-browser
2. Navigate to `localhost:8000`
3. Click on the "Download IRMAseal" button
4. In Thunderbird, click on the hamburger icon
5. Click on "Add-ons and Themes"
6. Click on "Install Add-on from File"
7. Select the downloaded file
8. Click on "Add"

We have found no issues here, since all steps are clearly explained on the website.

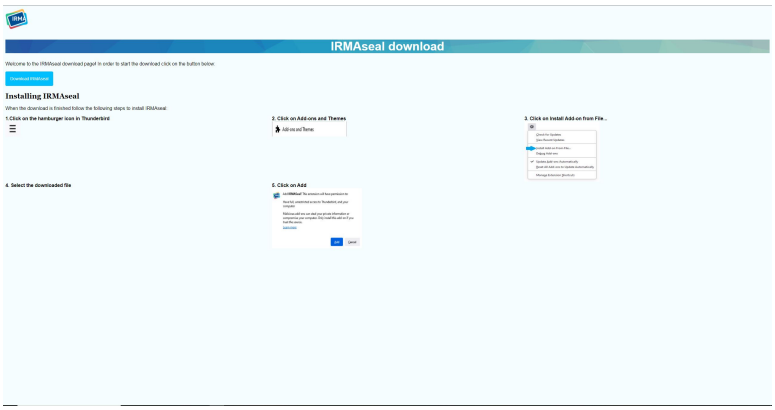

Figure 4.10: The IRMAseal download site (solely used in the usability study)

Compose the email

Composing the email takes the following steps:

1. Click on the "New Email" button

2. Fill in Jane's email address
3. Fill in the subject
4. Fill in the body of the email

These steps are all the same as sending an unencrypted email and do not contain any issues.

Check if encryption is enabled

Checking if encryption is enabled is straightforward and should be enabled by default. As shown in Figure 4.11, the IRMAseal bar is coloured green when encryption is enabled and orange when encryption is not enabled. Additionally, the bar contains explanation in text what the system will do depending on whether encryption is enabled or not. When Johnny has checked whether encryption is enabled, he needs to click on the "Verzenden" (Send) button.

Figure 4.11: The IRMAseal toggle in the Outlook toolbar

The IRMAseal bar is a plus in the categories *Match between System and the Real World* and *Aesthetic and Minimalist Design*, because of its colour scheme depending on encryption being enabled and its minimalistic design, but still containing all the necessary information.

Select the required attributes for Jane to show

Selecting the attributes takes the following steps:

1. Toggle all pieces of information Johnny wants Jane to prove about herself.
2. Click on "Gegevens invullen" (Fill in information)

We find no issues in these steps. The fact that the locks are locked when selected is a nice visual presentation and we can categorise this as a plus in the category *Match between System and the Real World*. Furthermore, there is a progress bar showing the security strength based on the selected options (see Figure 4.12). This is a good implementation of *Visibility of System Status*.

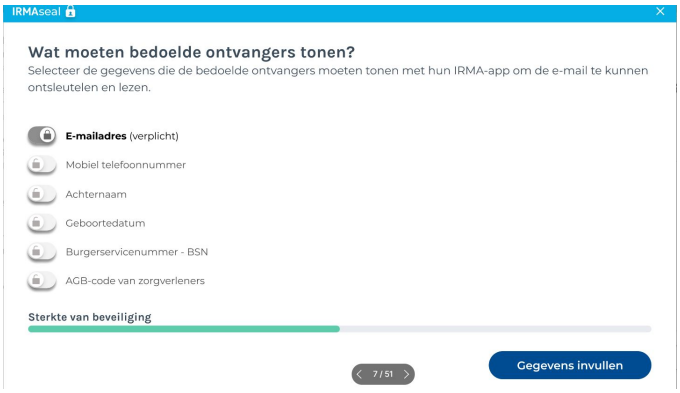

Figure 4.12: The window to select the attributes required from the recipient

Fill in the information

In this case, Johnny only requires Jane to show her email-address attribute. This is the minimal requirement for sending an encrypted email using IRMAseal. Filling in the information based on the selected attributes in the previous step (only the email of Jane) takes the following actions:

1. Check if the email is correctly copied.
2. Click on "Verder" (Next)

We have not found any usability issues in these steps.

Send the email

After the information is filled in, Johnny arrives at the signing screen, as can be seen in Figure 4.13. In order to send the email, Johnny has to perform one step: Click on "Niet ondertekenen" (Do not sign). After this, the email will start sending. In this step, the goal of Johnny does not match the system's and question 1 from the cognitive walkthrough has to be answered with "No". We can categorise this issue as an issue in the category *Recognition Rather Than Recall*, since Johnny should not have to memorise that clicking this button starts the email sending process. During the sending process, there is an option "Ongedaan maken" (Cancel) to cancel the sending process. This is a plus in the category *User Control and Freedom*.

Figure 4.13: The signing screen in IRMAseal

4.4.2 Decrypting an email

In order to decrypt Johnny's email, Jane needs the IRMA app with her email address attribute loaded on it on her phone. We assume that this is indeed the case. Decrypting Johnny's email takes the following steps:

1. Scan the QR using the IRMA app
2. Provide the requested attribute

Since these steps are not performed within IRMAseal but in IRMA, we will only be showing the process and not further discuss these.

Scan the QR using the IRMA app

Scanning the QR code using the IRMA app takes the following steps:

1. On the IRMA app click on "Scan QR"
2. Aim the camera to the QR code on the email.

These steps are shown in Figure 4.14.

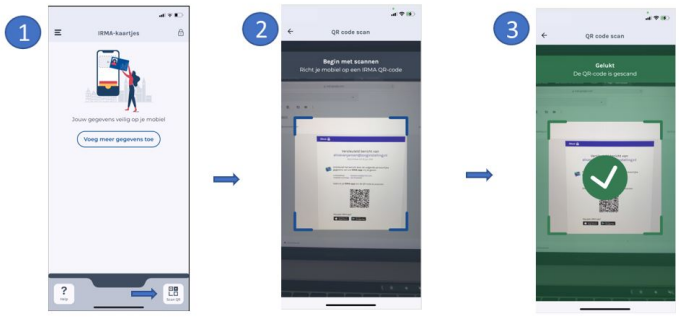

Figure 4.14: Flow of scanning the IRMA QR code

Provide the requested attribute

Providing the requested attribute takes only one step: clicking on the "Ja" (Yes) button as shown in Figure 4.15.

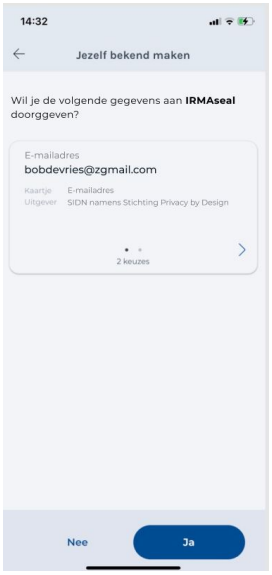

Figure 4.15: Sharing the IRMA attribute

After this step the email will be decrypted and Jane will be able to read Johnny's message.

4.4.3 Error Walkthrough

In this section, we will perform another walkthrough where we are going to try to make mistakes where they can be expected based on the regular cognitive walkthrough in order to check if recovering from these mistakes is possible.

Filling in wrong information

Let us assume Johnny has made a mistake with filling in some required information and wants to correct it, for example, the wrong email address. When this happens, Johnny can close the IRMAseal window, change the email address and click on send without losing the rest of his email. Additionally, it is possible to go back and forth between all screens in IRMAseal to check if everything is filled in correctly. This is a plus in the category *User Control and Freedom*.

Clicking on the "Do not sign button" without knowing what it actually does

Let's assume Johnny made it to the window where he needs to decide if he wants to sign his email. He decides that signing is not necessary and clicks on do not sign. To his surprise, the system starts sending his email. Fortunately, there is a cancel option which takes him back to the same window. This is a plus in the category *User Control and Freedom*.

We have seen quite some design choices that adhere to the ten heuristics. An overview of these choices, categorised according to the heuristics can be found in Appendix A.2.

Chapter 5

Usability Study

This chapter elaborates on the usability study we performed. We will address this study using a within-subjects design. We start this chapter with explaining the study design, followed by discussing the participants and apparatus. Then, we will explain the used procedure before presenting the obtained results.

5.1 Design

The study took three weeks from 2 till 18 December, where nine participants participated. Each session took about ninety minutes, where sixty were scheduled for trying to perform the tasks and the rest for a brief introduction, filling in questionnaires and the post-study interview. Everything needed to be able to perform the tasks, such as a laptop, phone, email account and IRMA app was provided for the participants. For privacy reasons, we did not use any real data of the participants.

The independent variable in this study was the secure email tool used for sending an encrypted email. We have chosen for a within-subjects design, where each participant is exposed to all of the conditions of the experiment as explained by (Field and Hole, 2003). More concretely, half of the participants will start with IRMAseal as secure email tool and the other half with PGP. The dependent variables were the time participants needed to complete all the tasks (within the given time limit), the task success within the time limit, and the SUS scores of both systems.

5.2 Participants

In our study, nine participants participated on a voluntary basis. We recruited them from our private circle. We tried to gather a diverse participant pool, where all participants should be regularly email and internet

users. Knowledge of IRMA was not necessary, but would have been a plus point.

The participants who participated in our study where all male (9; 100%). Most of the participants were between 18 and 24 years old (8; 88.9%) and the rest where 25+ years old (1; 11.1%). Five of the participants had an university background (5; 55.6%) and the other four, a college background (4; 44.4%). Two participants were Computer Science students and therefore had used PGP before (2; 22.2%).

5.3 Apparatus

The apparatus consisted of a laptop with the Thunderbird client installed on it, a notepad used for writing down quotes or events during the study, SUS statements sheets the participants were asked to fill in after using one of the systems (as shown in Appendix A.1), an information letter for the participant, a consent form and a task description describing the tasks we asked the participant to perform. Furthermore, we created the email accounts for Johnny and Jane, which the participant would use during the session, and we created a download site solely for this usability study where participants were supposed to download the IRMAseal application from. The IRMAseal version used is, in contrast to the walkthrough, a working prototype that looks different from the envisioned mock-up (the interface can be found at Figure 5.2). Also, we used the IRMA app on the researcher's phone with the email attributes of Johnny and Jane already loaded on it. As a thanks for participating, we prepared some small gifts for the participants in the form of chocolate or craft beer.

5.4 Procedure

For each participant, we scheduled a separate session. At the start of each session, the participant was asked to read the information document and consent form carefully before agreeing with the conditions. Once the consent form had been signed, the participant received the task description for the experiment. There were two versions of the experiment setup, one where the participant started with IRMAseal and one where the participant started with PGP. These versions were distributed as even as possible among the nine participants. Before the participants started using the secure email mechanisms, we introduced them to IRMA and the Thunderbird email client. We showed them the IRMA app, let them perform a demo[1] to familiarise them with IRMA and to ensure the same starting conditions

[1] `https://privacybydesign.foundation/demo/mail/`

as in the walkthrough, and explained that they could request this app from the researcher whenever they needed it. Furthermore, to introduce the participant to Thunderbird, we let them send an unencrypted email to their *partner*, which happened to be the researcher. To keep the consistency within the thesis, we used the same Johnny and Jane roles in the usability study. Participants were playing the role of Johnny, if they agreed to this, otherwise roles were switched. We played the partner role in all the sessions.

When the introductions were completed, the participant was asked to start using the first secure email mechanism. The scenario involved a request from the participant's partner asking him/her to send his/her (by us provided) SSN using secure email in order to be able to book a flight. Participants had for both PGP and IRMAseal thirty minutes to send an encrypted email.

In case the participant was using PGP to complete the task, we as partner sent the request for the SSN to the email-address used by the participant accompanied by our public key necessary for encrypting the email. We gave the participants thirty minutes for trying to complete the task. The participants were allowed to use the internet to look up information on anything they needed. However, they were not allowed to ask the researcher on how to proceed. Whenever the participant successfully sent an encrypted email with PGP, we stopped the timer. If a participant was not able to send an encrypted email within the time limit, we showed him or her the remaining steps to perform. When the encrypted email got sent, we showed the participant in our inbox how to recognise it. We skipped explicit decryption of the email, since we focussed on the in our view most important step in terms of the usability of PGP, which is encryption. Furthermore, we did not expect any problems for decryption as we already explained in the walkthrough. However, we did explain to the participant that decrypting the email only required the participant to open the email.

Sending an encrypted email using IRMAseal required an extra task for the participant: they needed to install IRMAseal before they could use it to send an encrypted email. The same as with PGP, we sent the participants an email with a request for information and proposed to send this encrypted using IRMAseal. Furthermore, we provided a link to the download website, where also the instructions on how to install IRMAseal were shown. When we received the encrypted email from the participant, we stopped the timer and replied with an encrypted email, which the participant had to decrypt. Differently from PGP, we asked the participants to decrypt with IRMAseal, since this takes quite some steps as already discussed in the walkthrough. Doing this required the IRMA app. The app could be requested from the researcher whenever the participant needed it. We did not time the duration of decrypting the email with IRMAseal, since we did not expect the user to

have any issues with it after the earlier completed IRMA demo and we did not have any data on decrypting with PGP to compare these timings with.

When the participant finished trying out a secure email mechanism, we asked the participant to fill in the SUS statements about their experience with the system.

After the participant tried out and filled in the SUS questions for both systems, we asked the participant some questions about their experiences with both systems in a post-study interview. In this interview we discussed both systems, asked the participant's opinion on them and asked if the participant would use any of the two systems in the future. Furthermore, since the participants only used the prototype version of IRMAseal, we showed them the clickable mock-up and asked for their opinion on this. After the post-study interview finished, the session got concluded, and we gave the participants a small gift as a thanks for their participation.

5.5 Results

This section presents the results from our usability study. We present the time it took for participants to send an encrypted email using PGP and IRMAseal and the rate of participants who succeeded in doing this within the time limit. Additionally, we discuss the rate of participants who succeeded in decrypting an encrypted email with IRMAseal. Furthermore, we present the results from the SUS questions answered by the participants after using the systems and determine if these are statistically significant. Then, we discuss the problems encountered by the participants during their process of sending an encrypted email, and we will be taking a look at the post-study interview results.

During this presentation of results participants are referred to as A[1-5]/B[1-4], where A participants started with IRMAseal as encryption tool and B participants with PGP.

5.5.1 Success rate and timings

As already explained, participants had thirty minutes trying to send an encrypted email with both systems. If the participant could not complete the task within the time limit, we stopped the timer and went to the next part of the study.

All nine participants succeeded to send an encrypted email using IRMAseal within this time limit. The average completion time was way faster than the thirty minutes which were scheduled with 3:19 minutes. For PGP, four out of the nine participants were not able to send an encrypted email within the time limit of thirty minutes (44.4%). The five participants (containing two with PGP experience) who were able to send an encrypted email had an average completion time of 11:49 minutes[2]. In Table 5.1 we have shown all the timings for both A and B participants.

	A1	A2	A3	A4	A5	B1	B2	B3	B4
PGP	Failed	Invalid	10:46	04:04	Failed	19:16	Failed	13:12	Failed
IRMAseal	05:14	04:11	02:30	02:47	02:20	05:17	05:53	06:41	02:42

Table 5.1: Completion times of the participants sending an encrypted email for both systems

We explained earlier that participants were not asked to decrypt an email encrypted with PGP, since this only required opening the encrypted email. However, because decryption with IRMAseal does require some steps, participants were asked to decrypt an encrypted email using IRMAseal. We did not measure the time participants needed to complete this task, but we did find that all nine participants were able to decrypt an email with IRMAseal without any problems.

5.5.2 SUS scores

After the participant had the chance to use PGP/IRMAseal for encryption and decryption (in the case of decrypting with PGP, has been told how it works), the participant was asked to fill in the SUS questions about that system. An overview of all ratings can be seen in Table 5.2. When we take a look at the overall score, we can see that PGP has a mean SUS score of 46.1 (for encrypting a message), which maps to the letter F in the adjective rating

[2]One participant was able to send an encrypted email without importing the recipients public key because of a small error in the setup. The completion time of this participant will not be taken account in our results, but we will count this as being able to send an encrypted email.

scale discussed in Chapter 2.2. Furthermore, we can see that IRMAseal has a mean score of 85.2, which maps to the letter B in that same rating scale.

	A1	A2	A3	A4	A5	B1	B2	B3	B4	Mean	Std.Dev
PGP	30	30	45	60	45	72.5	40	55	37.5	46.1	14.15
IRMAseal	65	100	85	80	85	95	95	95	67.5	85.3	12.53

Table 5.2: SUS score for both systems

5.5.3 Statistical Significance

To determine if our obtained results are statistical significant, we have performed a Wilcoxon Signed-Rank Test on our data using SPSS[3][4]. This test is the non-parametric equivalent of the dependent t-test, which allows using non-normal data (Field and Hole, 2003). As we can see in Figure 5.5.3 that a Sig. value in the Shapiro-Wilik test is below 0.05, which indicates that the data deviates from a normal distribution, and we need to use the non-parametric variant of the t-test.

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Treatment	Statistic	df	Sig.	Statistic	df	Sig.
SUSPGP	IRMA	,231	5	,200*	,881	5	,314
	PGP	,257	4	.	,902	4	,442
SUSIRMA	IRMA	,237	5	,200*	,950	5	,740
	PGP	,441	4	.	,630	4	,001

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Figure 5.1: Test of Normality output of SPSS

From the results of the Wilcoxin Signed-Rank Test we get that the SUS score of PGP (Mean= 46.1, Std= 14.1) is significantly less than the SUS score of IRMAseal (Mean= 85.3, Std= 12.5), $T = 2.675, p = .007$ with a large effect size ($\frac{2.675}{\sqrt{18}} = 0.631$).

5.5.4 Problems

During the study, the participants encountered some problems while trying to send an encrypted message using PGP and IRMAseal. With IRMAseal, the main problem was that participants rarely noticed the "encryption-on"

[3] `https://www.ibm.com/products/spss-statistics`

[4] Only on the SUS scores, because the sample size for the PGP timings is way to small as can be seen in Figure 5.5.1

button after installation. They managed to send the email with encryption turned on, but only a few were actually aware of this. For A participants, not seeing the encryption button turned not out to be a problem. After installation of the plug-in, they composed the email and send it without knowing if it would be encrypted. However, the B participants, who already used the Thunderbird user interface for PGP, were struggling. They believed that after the installation of IRMAseal, an extra option in the "Security" tab would appear. After some time, the encryption button was noticed, but participants still found it confusing. Participant B1 asked when noticing the button: "Is that it?". Participant B3 ignored the encryption button at first and after some time, when he found no other way and said: "It will not be the case that just sending will encrypt this", before unaware sending the encrypted email.

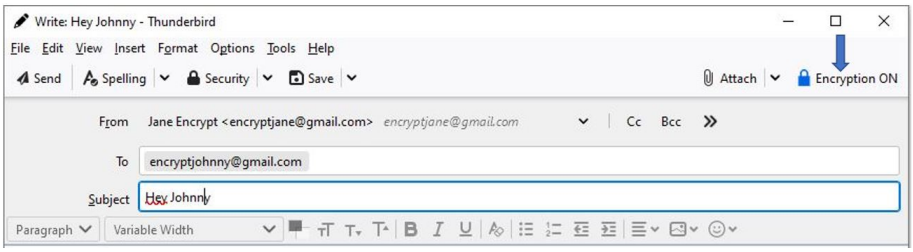

Figure 5.2: The prototype of IRMAseal used in the usability study

While using PGP, the participants encountered a broader range of problems. Especially, A participants sent their SSN to Jane without encrypting it, assuming this would work the same as IRMAseal. Another observation we made was that participants who did not find the official (and latest) version of the Thunderbird support page had more issues starting up. These participants had a hard time figuring out where to start. Participants who found the official support page were able to create their key pair without any problems. Where all participants were able to create their key pair one way or another, the next steps lead to the most problems. Often it happened that after creating the key pair, participants thought this would be enough to encrypt their email. They were able to enable encryption in the "Security" tab, but when they clicked on the send button, an error popped up on the screen saying there was something wrong with the key of the recipient. Some participants found it hard to resolve this problem. Three out of the four participants who were not able to complete the tasks within the thirty minutes got stuck when trying to upload Jane's public key. Some tried to open it in the Adobe Reader or Internet Explorer, and one participant deleted his key pair in order to be able to upload Jane's public key as a personal key. None of these attempts turned out to be successful.

5.5.5 Interview findings

During the study, we observed that the participants had a some trouble using PGP to encrypt their email. Quotes as "I have no clue what I have been doing until know" or "this is really complicated" were not rare. In the post-study interview, we found confirmation that participants found it harder to send an encrypted email with PGP compared to IRMAseal. The use of internet was really necessary for the use of PGP, where IRMAseal was easier and took fewer steps. When answering the question if they would use one of these systems more often in the future and if so, which one, most participants answered with IRMAseal, but not all. Some participants argued that scanning the QR for every encrypted email would become annoying after some time, especially if it would be used on a daily basis. One participant even suggested that it would be nice if the attributes were stored in the add-on after disclosing these once with the app, leading to a reduction in the amount of QR code scans. One participant, who claimed to have used PGP before, still would prefer PGP for sending real sensitive data, because PGP lets the user manage their own keys.

Since we have not been using the envisioned user interface of IRMAseal during the usability study, we walked the participants through it during the interview. We showed them Merel's design of IRMAseal in Outlook and explained what steps are needed to be taken in order to send an encrypted email, and to decrypt an encrypted email using this mock-up. Afterwards, we asked the participants to give their view on the mock-up. The responses were positive. Where the participants were struggling to see if they were making progress towards their goal using the prototype, they found Merel's mock-up more explanatory and more intuitive in comparison to the prototype.

Chapter 6

Discussion

In this chapter, we will be interpreting and explaining the obtained results from our usability study and user interface analysis. Furthermore, we are comparing our results against those from previous studies, discussed in the related work. Additionally, we will discuss the limitations of our research before giving our recommendations for IRMAseal and presenting some ideas for future research.

6.1 Findings

In this thesis, we are determining if IRMAseal is a more user-friendly alternative for secure email in comparison to PGP. As mentioned earlier in this thesis we would be evaluating both secure email systems using Quesenbery's Es where we selected the four most important ones; *Effective*, *Efficient*, *Error Tolerance* and *Easy to Learn*. We discuss these Es based on the findings from our research. First, we discuss and summarise those findings.

6.1.1 Findings from the user interface analysis

In the user interface analysis we performed a cognitive walkthrough and an error walkthrough were we categorised the issues and design adherences according to Nielsen's heuristics. We can see from the results of the cognitive walkthrough that in general more questions have been answered with "No" for PGP in comparison to IRMAseal. When we first start looking at the results of PGP, we see that mostly the first cognitive walkthrough question has been answered with "No". This means that often the user's goal does not match the goal of the system. Other observations we made in the analysis of PGP were that the vocabulary used in the system does not always match with the vocabulary of the user, and that the system often lacks appropriate feedback in the form of a confirmation message or a progress bar. From the error walkthrough of PGP, we see that it does support error pre-

vention quite well. For example, getting clear error messages when trying to delete a key pair or trying to send an encrypted email without having the recipient's public key is all handled quite well. However, there is one core problem which we expect a regular mail user will not be able to recover from; we expect that when the user does not look up the support page, he or she will not be able to send an encrypted email using PGP. Additionally, PGP does not prevent sending an encrypted email using a deleted public key of the recipient, where it does prevent it for revoked keys. Hence, it is important for the user to inform their contacts whenever he or she deleted his or her public key and created a new one. Otherwise, when the deleted public key is used for encryption, decryption will not be possible and the mail must be sent again with the up-to-date public key of the recipient.

When we take a look at the results of IRMAseal, we see that only one cognitive walkthrough question has been answered with "No". This relates to the fact that the goal of the user did not match the goal of the system when the user clicks on "Do not sign" and the system starts sending the email. Other observations from the heuristic evaluation are all positive, such as good visual feedback and using a simple design. From the error walkthrough, we found that the mistake of clicking on "Do not sign" without knowing that this will start sending the email can easily be cancelled by clicking on the cancel button. Additionally, we have seen that the user is able to navigate back and forth whenever anything needs to be changed.

6.1.2 Findings from the usability study

In the usability study, we measured the time participants needed to send an encrypted email using a PGP and IRMAseal, the rate of participants succeeding in completing this task within the time limit and the rate of participants being able to decrypt an encrypted email with IRMAseal. Additionally, we let the participants fill in the SUS questions in order to calculate SUS scores for both systems. When we compare the times participants needed to send an encrypted email using each system, we can see that sending an encrypted email with IRMAseal took less time than for PGP in all nine cases. Furthermore, we have seen that four out of the nine participants were not able to send an encrypted email using PGP within the given time limit. Additionally, we saw that all nine participants succeeded to perform this task using IRMAseal way before the time limit. When we look at the success rate for decrypting with IRMAseal, we see that all nine participants were able to decrypt an encrypted email using IRMAseal without any problems. The SUS scores given by the participants to both systems indicate that IRMAseal has been experienced as more usable than PGP, with a mean score of 85.3 for IRMAseal and a score of 46.1 for (encrypting with) PGP. These results are also underlined in the post-study interview, where par-

ticipants explained that they found PGP harder to use in comparison to IRMAseal, even with access to the internet. Additionally, most participants indicated that they would prefer using IRMAseal in the future over PGP, whenever they would communicate via secure email.

6.1.3 Linking to the Es

We will use the results of the user interface analysis to discuss the Es *Effective* and *Error Tolerant* for PGP and IRMAseal. The usability study results will be used to discuss *Efficient* and *Easy to Learn*. We will evaluate these Es with the following goals of a secure email user in mind: sending an encrypted email and decrypting an encrypted email.

Starting with *Effective*, which addresses if the software helps the user achieving their goals. We already have seen that this regularly is a problem when sending an encrypted email with PGP. However, we found no issues in decrypting with PGP. When we take a look at encrypting with IRMAseal, we found one similar issue that the software did not help the user achieving their goal, but not as many compared to PGP. For decrypting with IRMAseal, we found no issues. Overall, we can conclude that IRMAseal is more Effective than PGP.

The next E we will discuss is *Efficient*. This means the speed in which work can be done. We already concluded that sending an encrypted email using IRMAseal takes less time than sending an encrypted email using PGP. When we compare the speed in which decrypting can be performed, it is most likely that decrypting with PGP is more efficient.[1] Overall, since the sending process with IRMAseal is much faster in comparison to PGP, we can conclude that IRMAseal is more *Efficient* than PGP.

To determine *Error Tolerant*, which discusses how well users are prevented from making a mistake or how to recover from making one, we will be taking a look at the error walkthrough results. Where we only found two potential mistakes to make in sending an encrypted email with IRMAseal, there were a few more in sending an encrypted email with PGP. When we take a look at how these possible mistakes are prevented, or how the user can recover from making them when sending an encrypted email using IRMAseal, we see that this is integrated nicely into the system. The same holds for sending an encrypted email with PGP, where we find that most possible errors are nicely prevented by the system in such a way that buttons can not be clicked, or error messages are shown on screen. However, we did find that not all errors

[1] Although, we did not time these operations, we expect that automatic decryption by PGP when opening the email is faster than scanning a QR code with the IRMA app and disclosing the right attribute in order to decrypt the email.

can be recovered from. We expect that the user will not be able to recover from not looking up the user manual. When we take a look at decrypting with IRMAseal, we find that there are no possible mistakes to be made by the user, so there is also no need for recovering from these. For decrypting in PGP, we found that PGP does not prevent sending an encrypted email using a deleted recipient's public key, which could lead to users no being able to decrypt their email. Recovering from this mistake would require to communicate the up-to-date recipient's public key and send the encrypted email again, using that updated key. Overall, we can conclude that IRMAseal is more *Error Tolerant* in comparison to PGP, because there are less possible errors to make in IRMAseal and recovering from these few possibilities is nicely integrated into the system.

Lastly, *Easy to Learn* will be discussed with the success rate from the usability study. As already discussed, all nine participants learned how to send an encrypted email using IRMAseal in quite a short time frame. On the other hand, only five out of the nine participants (containing two with PGP experience) succeeded in sending an encrypted email using PGP. When we take a look at decrypting an encrypted email, we saw that all participants were able to decrypt an email with IRMAseal, and we expect this will be the same for PGP, since this only requires to open the encrypted email. Overall, we can conclude that IRMAseal supports *Easy to Learn* better than PGP does.

We have discussed the usability of PGP and IRMAseal in terms of the most relevant Es of Quesenbery. From this discussion, we can conclude that IRMAseal outperformed PGP on all earlier mentioned Es. Additionally, from the SUS scores obtained in our usability study where IRMAseal received a SUS score of 85.3 (corresponding to "Excellent") and PGP received a SUS score of 46.1 (corresponding to "Ok") for the most important step in terms of the usability of PGP, which is sending an encrypted email.

Overall, from these findings, we are able to answer our main research question whether IRMAseal is a more usable alternative in comparison to PGP. We can conclude that IRMAseal is a more usable alternative in comparison to PGP.

6.2 Comparison against previous studies

As we have seen in the related work chapter, there have been performed quite some usability studies on PGP. In the first *Johnny* paper, only four of the twelve participants were able to send an encrypted email using PGP (Whitten and Tygar, 1999). In 2006, Sheng et al. also performed a usability

study on PGP where none of the six novice PGP users were able to send an encrypted email (Sheng et al., 2006). Ruoti et al. found in their usability study on PGP that only one of the ten pairs was able to send an encrypted email using a PGP tool. Also rated their participants PGP with a SUS score of 34.5 as "Poor". All these papers argued that key management was a large issue for the participants in their studies.

When we take a look at the results from our usability study, we see that five out of the nine participants were able to send an encrypted email using PGP. We found that, just as in the earlier usability studies on PGP, key management is a large issue, since all participants who were not able to send an encrypted email using PGP, were having problems with managing their keys. Furthermore, our participants rated PGP with a SUS score of 46.1 as "Ok".

From these findings we are able to answer our sub-question in this thesis whether the usability of the current version of PGP has improved in comparison to the earlier version which has been analysed by Ruoti et al. in 2015. We can conclude that the usability of PGP has been improved since that study, but it is still not usable enough for the masses.

6.3 Limitations

There were a few limitations to our research. First, as mentioned earlier, we unfortunately have not been able to use the envisioned user interface of IRMAseal during the usability study and a small part of the user interface analysis. Where we were mostly able to use the envisioned user interface of IRMAseal during the user interface analysis, we could not use it at all for the usability study since it was not completely implemented yet. This means that the timings and SUS scores for IRMAseal only apply for the prototype version we have been using, and that these only give the direction in which the SUS scores and timings of the final version of IRMAseal can be.

Furthermore, in this thesis we focussed on the first step of secure email communication, which is sending an encrypted email. However, we also discussed decrypting in the user interface analysis and partially in the usability study, where we decided to only let participants decrypt an encrypted email with IRMAseal and explained to them that decrypting with PGP only requires to open the email. For consistency reasons, we should have let the participants decrypt an email with PGP. This means that where the SUS score of IRMAseal takes encrypting and decrypting into account, the SUS score of PGP is limited to encrypting an email, which is in our view the most important step in terms of the usability of PGP.

Another limitation is related to the user interface analysis. As we already have explained, results of the cognitive walkthrough become better when performed by a group of reviewers. However, in this thesis, we were the only ones who performed the walkthrough. Hence, the results of it are only based on our own findings.

Lastly, we would have preferred a larger and more diverse participant pool. Unfortunately, we did not have the resources to make this happen. Additionally, since our experiment setup required physical meetings and the COVID-19 regulations at the moment of performing the study limited our possibilities to arrange these meetings, we were limited to find participants from our private circle.

6.4 Recommendations for IRMAseal

After performing a user interface analysis and a usability on IRMAseal, we have some recommendations for future development. In the user interface analysis, we found that clicking on "Do not sign" triggered the system to send the email. This problem can be resolved by creating another window with a summary of settings and a dedicated 'send' button after clicking on "Do not sign" or by changing the button into "Do not sign and send".

One of our participants suggested that storing the disclosed IRMA attributes in the system would drastically decrease the amount of QR code scans, which participants imagined could become annoying when one receives a large amount of encrypted emails.

Our last recommendation is based on the paper from Braun and Oostveen (2019), who found that a large group of PGP users gave up on PGP because it had no availability on other platforms. We suggest that a mobile version of IRMAseal would help the adoption of the secure email mechanism.

We had been asked to ask the participants in our study to think about a new name for IRMAseal. The particpants came up with the following names:

- EasyEncrypt
- ISeal

6.5 Future Research

We mentioned earlier that since we used the prototype version of IRMAseal in the usability study, our obtained SUS scores and timings only give a direction in which the SUS scores and timings of IRMAseal can be. In order to obtain those scores and timings, another usability study on IRMAseal with the envisioned user interface needs to be performed.

Furthermore, since the participants only had the chance to encrypt an email with PGP before answering the SUS questions, we recommend that in a follow-up study, decrypting with PGP is included in the usability study for a more accurate SUS score, which can be used for a new usability comparison with IRMAseal and for a new comparison with previous PGP usability studies.

Additionally, since we only performed a small scale usability study where participants were first time users, it would be a good idea to let a larger group use the system and for a longer period. When IRMAseal gets used by a larger group for a longer period of time, there is also a larger basis to study the factors helping or hindering the adoption of IRMAseal.

Chapter 7

Conclusions

In this thesis, we compared the usability of PGP against IRMAseal in order to find out if IRMAseal could be a more user-friendly alternative for PGP. In order to determine the usability of both systems, we performed a user interface analysis and a usability study. From those results we were able to compare both secure email mechanisms in terms of Quesenbery's Es, where we put the most weight on *Effective*, *Efficient*, *Easy to Learn* and *Error Tolerant*, and the SUS scores. The comparison in terms of the Es resulted in IRMAseal outperforming PGP on all earlier mentioned Es. Furthermore, from the SUS scores we saw that IRMAseal received a score 85.3 (corresponding to "Excellent") and that PGP received a score of 46.1 (corresponding to "Ok") for the most important step in terms of the usability of PGP, which is sending an encrypted email. Overall, we can conclude that IRMAseal is a more usable alternative in comparison to PGP. Furthermore, we can conclude that the usability of the current version of PGP has improved in comparison to the earlier version studied by Ruoti et al. in 2015, but it is still not usable enough for regular email users.

For future research, we recommend performing another usability study on IRMAseal using the envisioned user interface. Additionally, for a more accurate comparison with PGP, we recommend including decryption with PGP in the usability study. Lastly, a long term study with a larger participant pool would give more accurate results in terms of the usability of IRMAseal, since our study only focussed on first-time use. Additionally, a long term usability study on IRMAseal would also give the opportunity to study the factors helping or hindering the adoption of IRMAseal in depth.

Bibliography

Aaron Bangor, Philip Kortum, and James Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies*, 4(3):114–123, May 2009. ISSN 1931-3357.

Leon Botros and Daniel Ostkamp. IRMA Meetup 25/06/2021: Leon Botros en Daniel Ostkamp (RU), Versleutelde e-mail met IRMA. URL `https://www.youtube.com/watch?v=B6NDA1Ouq1k&ab_channel=SIDN`.

Sven Braun and Anne-Marie Oostveen. Encryption for the masses? An analysis of PGP key usage. *Mediatization Studies*, 2:69–84, 2019.

John Brooke. SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

Donald Davies. A brief history of cryptography. *Information Security Technical Report*, 2(2):14–17, 1997. ISSN 1363-4127. doi: https://doi.org/10.1016/S1363-4127(97)81323-4. URL `https://www.sciencedirect.com/science/article/pii/S1363412797813234`.

A. Field and G. Hole. *How to Design and Report Experiments*. SAGE Publications, 2003. ISBN 9780761973836.

Simson Garfinkel and Robert Miller. Johnny 2: A user test of key continuity management with S/MIME and Outlook Express. pages 13–24, 01 2005. doi: 10.1145/1073001.1073003.

Eman Hableel, Young-Ji Byon, and Joonsang Beak. Public Key Infrastructure for UAE: A Case Study. In *Proceedings of the 6th International Conference on Security of Information and Networks*, SIN '13, page 336–340, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324984. doi: 10.1145/2523514.2527099. URL `https://doi-org.ru.idm.oclc.org/10.1145/2523514.2527099`.

Harry Halpin. SoK: Why Johnny Can't Fix PGP Standardization. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450388337. doi: 10.1145/3407023.3407083. URL `https://doi.org/10.1145/3407023.3407083`.

Rolf Molich and Jakob Nielsen. Improving a Human-Computer Dialogue. *Commun. ACM*, 33(3):338–348, mar 1990. ISSN 0001-0782. doi: 10.1145/77481.77486. URL https://doi.org/10.1145/77481.77486.

Jakob Nielsen. 10 usability heuristics for user interface design, Apr 1994. URL https://www.nngroup.com/articles/ten-usability-heuristics/.

Jakob Nielsen and Rolf Molich. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, page 249–256, New York, NY, USA, 1990. Association for Computing Machinery. ISBN 0201509326. doi: 10.1145/97243.97281. URL https://doi-org.ru.idm.oclc.org/10.1145/97243.97281.

Peter G Polson, Clayton Lewis, John Rieman, and Cathleen Wharton. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, 36(5):741–773, 1992.

J Ronald Prins and Business Unit Cybercrime. DigiNotar certificate authority breach "operation black tulip". *Fox-IT, November*, page 18, 2011.

Privacy by Design Foundation. IRMA in detail, a. URL https://privacybydesign.foundation/irma-explanation/.

Privacy by Design Foundation, b. URL https://irma.app/.

Whitney Quesenbery. Balancing the 5Es of usability. *Cutter IT Journal*, 17 (2):4–11, 2004.

Karen Renaud, Melanie Volkamer, and Arne Renkema-Padmos. Why doesn't Jane protect her privacy? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 244–262. Springer, 2014.

Scott Ruoti, Jeff Andersen, Daniel Zappala, and Kent Seamons. Why Johnny Still, Still Can't Encrypt: Evaluating the Usability of a Modern PGP Client. 10 2015.

Scott Ruoti, Jeff Andersen, Tyler Monson, Daniel Zappala, and Kent Seamons. A Comparative Usability Study of Key Management in Secure Email. In *Proceedings of the Fourteenth USENIX Conference on Usable Privacy and Security*, SOUPS '18, page 375–394, USA, 2018. USENIX Association. ISBN 9781931971454.

Jörg Schwenk, Marcus Brinkmann, Damian Poddebniak, Jens Müller, Juraj Somorovsky, and Sebastian Schinzel. Mitigation of Attacks on Email End-to-End Encryption. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1647–1664,

New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370899. doi: 10.1145/3372297.3417878. URL https://doi.org/10.1145/3372297.3417878.

Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984. doi: 10.1007/3-540-39568-7_5.

Steve Sheng, Levi Broderick, Colleen Alison Koranda, and Jeremy J Hyland. Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software. In *Symposium On Usable Privacy and Security*, pages 3–4. ACM, 2006.

Christian Stransky, Oliver Wiese, Volker Roth, Yasemin Acar, and Sascha Fahl. 27 years and 81 million opportunities later: Investigating the use of email encryption for an entire university. In *43rd IEEE Symposium on Security and Privacy, IEEE S&P 2022, May 22-26, 2022*. IEEE Computer Society, May 2022.

Alexander Ulrich, Ralph Holz, Peter Hauck, and Georg Carle. Investigating the OpenPGP Web of Trust. pages 489–507, 09 2011. doi: 10.1007/978-3-642-23822-2_27.

Joel Weise. Public key infrastructure overview. *Sun BluePrints OnLine, August*, pages 1–27, 2001.

Alma Whitten and J. D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, SSYM'99, page 14, USA, 1999. USENIX Association.

Appendix A

Appendix

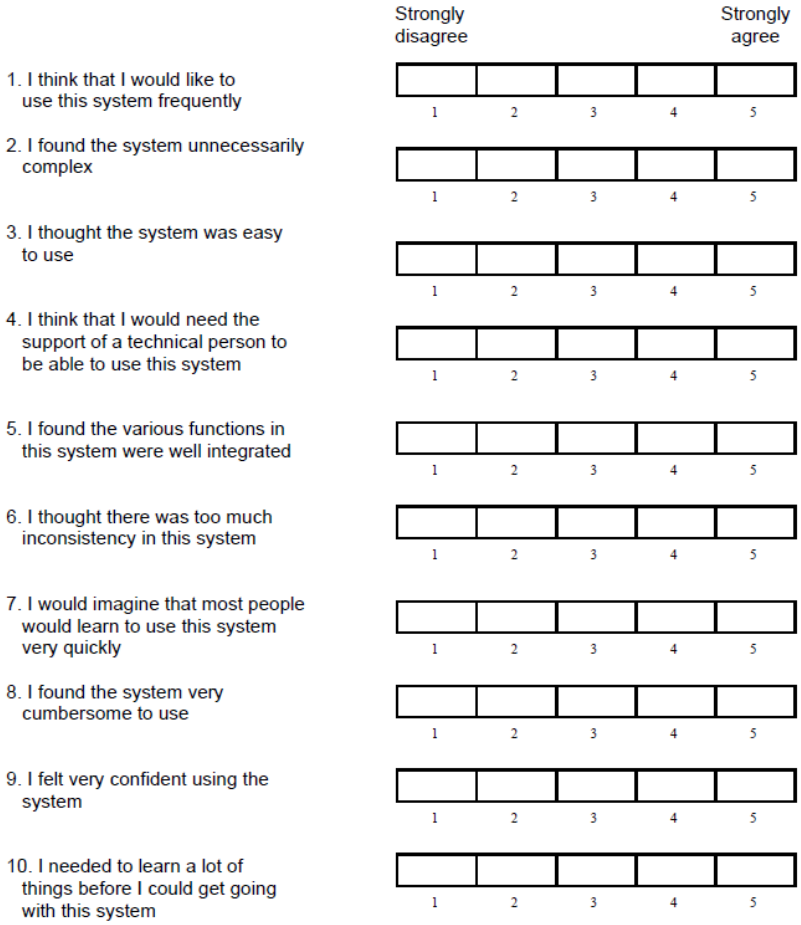

Figure A.1: The ten statements for the System Usability Scale (Brooke, 1995)

Radboud Universiteit

INFORMATIE OVER HET ONDERZOEK

Johnny can encrypt? A usability study of IRMAseal

Inleiding

Wij vragen u om mee te doen aan een bachelorscriptie onderzoek. Meedoen is vrijwillig. Om mee te doen is uw schriftelijke toestemming nodig. Voordat u beslist of u wilt meedoen aan dit onderzoek, krijgt u uitleg over wat het onderzoek inhoudt. Lees deze informatie rustig door en vraag de onderzoeker uitleg als u vragen heeft.

Beschrijving en doel van het onderzoek

In dit onderzoek willen we de gebruiksvriendelijkheid van een tool voor veilig te mailen testen. Deze tool wordt momenteel op de Radboud Universiteit ontwikkeld. We willen ook deze nieuwe tool vergelijken met een bestaande oplossing voor veilig mailen. Dit onderzoek is deel van mijn bachelor scriptie. De resultaten van dit onderzoek wordt als feedback gebruikt bij de verdere ontwikkeling van dit programma. Verder worden de resultaten beschreven in mijn bachelor scriptie.

Wat wordt er van u verwacht?

In dit onderzoek gaat u in een sessie van ongeveer 90 minuten twee email *tools* gebruiken om een mail met *gevoelige informatie* veilig (versleuteld) te sturen. U wordt ook gevraagd om een versleutelde mail te ontsleutelen. Na het gebruik van beide tools wordt u verwacht enkele vragen te beantwoorden over uw ervaring met deze tools.
U voert deze taken uit op een laptop van een onderzoeker. U gebruikt telkens een speciaal voor dit onderzoek ingerichte mailaccount, niet uw persoonlijke mailaccount. De *gevoelige informatie* wordt door de onderzoeker beschikbaar gesteld en is daadwerkelijk ook niet gevoelig.

Risico's en ongemak

Wij verwachten geen risico's of ongemakken. Het is goed mogelijk dat U niet in staat zult zijn om sommige taken binnen het tijdslimiet uit te voeren. Ook deze informatie is zeer waardevol voor ons onderzoek. Wij testen niet U maar de gebruiksvriendelijkheid van de tools.

Welke gegevens worden er verzameld?

Wij vragen uw naam op het toestemmingsformulier, uw leeftijdscategorie, geslacht en educatieniveau . Verder zullen wij tijdens het onderzoek notities maken over handelingen die U verricht of opmerkingen die U maakt. Ook noteren wij of het U gelukt was om de taken succesvol af te ronden binnen het tijdslimiet en zoja hoe lang U erover heeft gedaan. Verder wordt U gevraagd enkele vragen te beantwoorden over uw ervaringen met de tools.

Vrijwilligheid

U doet vrijwillig mee aan dit onderzoek. Daarom kunt u op elk moment tijdens het onderzoek uw deelname stopzetten en uw toestemming intrekken. U hoeft niet aan te geven waarom u stopt. Afzien van of stoppen met deelname heeft geen nadelige gevolgen voor u.

Wat gebeurt er met mijn gegevens?

De onderzoeksgegevens die we in dit onderzoek verzamelen, zullen door ons gebruikt worden voor de bachelorscriptie en voor de verdere ontwikkeling van de email tool. We publiceren de onderzoeksresultaten in de bachelor scriptie. Verder zullen we de resultaten delen met de ontwikkelaars van de tool**.**
Persoonsgegevens die verzameld worden, blijven vertrouwelijk. Als we gegevens met andere onderzoekers delen, kunnen deze dus niet tot u herleid worden.

Het door U ondertekende toestemmingsformulier zal gedurende 10 jaar na afronding van het onderzoek bewaard worden. Uw geanonimiseerde onderzoeksgegevens worden bewaard tot ten minste 10 jaar na het afronden van het onderzoek.

U kunt tot vier weken na deelname ook uw onderzoeksgegevens en persoonsgegevens laten verwijderen. Dit kunt u doen door een mail te sturen naar *niels.starren@ru.nl*.

We bewaren alle onderzoeks- en persoonsgegevens op beveiligde wijze volgens de richtlijnen van de Radboud Universiteit.

Heeft u vragen over het onderzoek?
Als u vragen heeft of meer informatie over het onderzoek wilt hebben, kunt u contact opnemen via de contactgegevens onderaan deze brief. Ook kunt U mijn begeleider (Hanna Schraffenberger, hanna.schraffenberger@ru.nl) contacteren.

Toestemmingsverklaring
Als u aan dit onderzoek mee wilt doen, vragen we u de toestemmingsverklaring te ondertekenen. Door uw schriftelijke toestemming geeft u aan dat u de informatie heeft begrepen en instemt met deelname aan het onderzoek.

Met vriendelijke groet,

Niels Starren, niels.starren@ru.nl {telefoonnummer}

Radboud Universiteit

TOESTEMMINGSVERKLARING

voor deelname aan het wetenschappelijke onderzoek: *Johnny can encrypt? A usability Study of IRMAseal*

Ik bevestig hierbij dat:
- Ik naar behoren ben ingelicht over het onderzoek, zowel schriftelijk als mondeling;
- Ik heb de informatiebrief consentV1nederlands.docx gelezen;
- Ik heb de mogelijkheid gehad om vragen te stellen
- Mijn vragen naar behoren zijn beantwoord;
- Ik heb ruimschoots de kans gekregen om goed na te denken over deelname aan deze studie
- Ik vrijwillig deelneem aan deze studie.

Ik begrijp dat:
- Ik het recht heb mijn toestemming te allen tijde in te trekken zonder opgave van redenen en zonder vrees voor nadelige gevolgen, door contact op te nemen met Niels Starren op {telefoonnummer} of niels.starren@ru.nl
- Ik heb het recht om mijn onderzoeksgegevens te laten wissen tot 1 maand na afloop van het onderzoek.
- Ik heb het recht om mijn toestemming voor de (verdere) verwerking van mijn (speciefieke) persoonsgegevens in te trekken;
- Mijn persoonsgegevens worden verwerkt in overeenstemming met de privacyverklaring van de Radboud Universiteit (https://www.ru.nl/english/vaste-onderdelen/privacy-statement-radboud-university/);

Ik ben akkoord dat:
- Mijn persoonlijke en/of onderzoeksgegevens in het kader van dit onderzoek zullen worden verkregen voor wetenschappelijke doeleinden en gedurende 10 jaar beschikbaar zullen zijn.
– Het getekende toestemmingsverklaring voor 10 jaar bewaard zal worden;
- Mijn persoonsgegevens, uitsluitend verzameld voor administratieve redenen, bewaard zullen worden voor maximaal 1 maand na het afronden van het onderzoek;
- Toezichthoudende autoriteiten mogen mijn persoons- en onderzoeksgegevens inzien met het oog op controle voor het onderzoek;

Daarnaast geef ik ook toestemming:
- voor het verwerken van de volgende (bijzondere) persoonsgegevens over mij: leeftijdscategorie, geslacht en educatieniveau

Ik begrijp dat ik, om aan het onderzoek te mogen deelnemen. `JA' moet beantwoorden op alle bovenstaande punten.

Ik ga akkoord met deelname aan deze studie.

Naam:

Handtekening:

Datum:

In te vullen door de onderzoeker:

Ik, de ondergetekende, verklaar dat de hierboven genoemde persoon geschreven en gesproken is geïnformeerd over het bovengenoemde onderzoek.

Naam:
Positie, onderzoeksinstelling:
Handtekening:
Datum:

Task Description V1

Niels Starren

January 15, 2022

In this study you will be playing the role of Johnny. You have received a request from Jane to send your Social Security Number (SSN) for booking a flight. For extra security, Jane advised you to use a secure email mechanism to protect your information.

1 Introduction to IRMA

Your first task is to go to the the following website: `https://privacybydesign.foundation/demo/mail/` On this website you will be performing a demo with sharing your IRMA email-address attribute.

2 Introduction to Thunderbird

Next, we will familiarise you with the Thunderbird email client. Please send an email to Jane, **encryptjane@gmail.com** that you will be looking at sending the requested information in an encrypted email and that he/she can expect the email soon.

3 Secure email using IRMAseal

Look at your email, Jane send you a message! We ask you to send your SSN (**123456**) to Jane using IRMAseal. But first, since IRMAseal has **not** been installed on your device, it remains as a task for you to be completed. When we have received your secured email we will reply with an encrypted email containing a secret message. Your second task is to decrypt the email you received from us. The time limit for this task is **thirty minutes**.

4 Secure email using PGP

You have received an email! Check your mailbox to see what Jane send you! Your next task is the same as before, sending your SSN to Jane, but this time using PGP. PGP comes pre-installed with the Thunderbird email client which

is already installed on the device in front of you. In the case you are stuck, looking up information on the internet is allowed, however the researcher will not be answering questions on the workings of the system. The time limit for this task is **thirty minutes**.

Credentials email accounts and IRMA app:

Johnny:
encryptjohnny@gmail.com

Jane:
encryptjane@gmail.com

IRMA app: request from researcher

Visibility of System Status	**Issue**: • no confirmation when acceptance of key has been changed • only a small picture in the lower right corner when encryption is enabled • only a small picture in the lower right corner when the message is signed • no explanation when encryption can not be enabled for some reason
Match between System and Real World	**Issue**: • vocabulary of the system does not match the user's vocabulary • enabling encryption should not automatically enable signing the message
User Control and Freedom	**Issue**: not being able to cancel the sending process of an encrypted email
Consistency and Standards	**Issue**: • missing an dedicated Insert button when inserting a key • the button for enabling encryption should have been called 'Encryption' • looking up the user manual should not be necessary
Error Prevention	**Issue**: allowing to send an encrypted email to a deleted public key **Plus**: • making the encryption button not clickable when there is no key private key uploaded • clear warnings when the user wants to delete the private key • good explanation of consequences and being able to cancel process of key pair deletion
Recognition Rather Than Recall	**Issue**: no explanation of the need for public key verification
Flexibility and Efficiency of Use	**Plus**: Shortcut for importing a public received as an attachment
Aesthetic and Minimalist Design	
Recognize, Diagnose, and Recover from Errors	**Issue**: unlikely to complete tasks without reading the support page **Plus**: clear messages when there is something wrong with the recipient's key
Help and Documentation	**Plus**: good support page

Table A.1: Overview heuristics from the cognitive walkthrough of PGP in Thunderbird

Visibility of System Status	**Plus**: security strength bar
Match between System and Real World	**Plus**: • the colour scheme of the IRMAseal bar depending on enabling encryption • using the lock metaphor for the attributes needed to be shown
User Control and Freedom	**Plus**: • cancel button during the sending an email process • being able to scroll back and fort to check and if necessary, change information
Consistency and Standards	
Error Prevention	
Recognition Rather Than Recall	**Issue**: clicking on "Niet ondertekenen" (Do not sign), starts the sending process
Flexibility and Efficiency of Use	
Aesthetic and Minimalist Design	**Plus**: encryption enabled bar is simple and contains only necessary information
Recognize, Diagnose, and Recover from Errors	
Help and Documentation	

Table A.2: Overview heuristics from the cognitive walkthrough of IRMAseal in Outlook