"I don't know why I check this..." Investigating Expert Users' Strategies to Detect Email Signature Spoofing Attacks

Peter Mayer, *SECUSO - Security, Usability, Society, Karlsruhe Institute of Technology;* Damian Poddebniak, *Münster University of Applied Sciences;* Konstantin Fischer and Marcus Brinkmann, *Ruhr University Bochum;* Juraj Somorovsky, *Paderborn University;* Angela Sasse, *Ruhr University Bochum;* Sebastian Schinzel, *Münster University of Applied Sciences;* Melanie Volkamer, *SECUSO - Security, Usability, Society, Karlsruhe Institute of Technology*

https://www.usenix.org/conference/soups2022/presentation/mayer

This paper is included in the Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022).

August 8–9, 2022 • Boston, MA, USA

978-1-939133-30-4

“I don’t know why I check this …” – Investigating Expert Users’ Strategies to Detect Email Signature Spoofing Attacks

Peter Mayer[1], Damian Poddebniak[2], Konstantin Fischer[3], Marcus Brinkmann[3], Juraj Somorovsky[4], Angela Sasse[3], Sebastian Schinzel[2], and Melanie Volkamer[1]

[1]SECUSO - Security, Usability, Society, Karlsruhe Institute of Technology
[2]Münster University of Applied Sciences
[3]Ruhr University Bochum
[4]Paderborn University

Abstract

OpenPGP is one of the two major standards for end-to-end email security. Several studies showed that serious usability issues exist with tools implementing this standard. However, a widespread assumption is that expert users can handle these tools and detect signature spoofing attacks. We present a user study investigating expert users’ strategies to detect signature spoofing attacks in Thunderbird. We observed 25 expert users while they classified eight emails as either having a legitimate signature or not. Studying expert users explicitly gives us an upper bound of attack detection rates of all users dealing with PGP signatures. 52% of participants fell for at least one out of four signature spoofing attacks. Overall, participants did not have an established strategy for evaluating email signature legitimacy. We observed our participants apply 23 different types of checks when inspecting signed emails, but only 8 of these checks tended to be useful in identifying the spoofed or invalid signatures. In performing their checks, participants were frequently startled, confused, or annoyed with the user interface, which they found supported them little. All these results paint a clear picture: Even expert users struggle to verify email signatures, usability issues in email security are not limited to novice users, and developers may need proper guidance on implementing email signature GUIs correctly.

1 Introduction

Signatures can provide end-to-end protection of the authenticity and integrity of email messages. Yet, Müller et al. [19] showed that verifying email signatures and displaying the result of the verification in a graphical user interface (GUI) is very challenging. Among others, they described *weak signature forgeries* that mimic GUI elements of a valid signature closely, but not perfectly. Upon close inspection, the user can detect that the GUI elements are fake. For example, they used HTML and CSS to include a green signature validation banner, but the fake banner was positioned incorrectly and did not provide the interactivity of the original. They classify the forgery as *weak* because they argue that vigilant users can detect it. In this work, we examine a subset of their attacks and attempt to answer the following question: *Which strategies do users employ to detect email signature spoofing, and how susceptible do these strategies leave them to these attacks?*

We answer this question by interviewing *expert users* of Thunderbird, who frequently use signatures and are familiar with public-key cryptography, digital signatures, and their email clients. We conducted a user study with participants drawn from attendees of FOSDEM 2020 – a European open source developer conference that also hosted an OpenPGP key signing party. Two pre-studies at the Chaos Communication Camp and Congress in 2019 informed the design of this study.

Our study participants were asked to use Thunderbird and its OpenPGP-plugin Enigmail to inspect eight semantically identical emails. Four of these emails contained a valid signature, and four contained an invalid signature. The invalid signatures were forgeries similar to the weak forgeries in [19]. The participants had to decide whether a signature was legitimate or not. As they were expert users, this gives an upper bound on how well users can detect such attacks.

Our results indicate that even expert users have no effective strategies to detect email signature spoofing attacks, leading to 52% of our participants failing to detect at least one out of four forged email signatures. Our participants’ checks were diverse: They applied 23 different checks when inspecting the attack emails. Of these checks, only 8 tended to be helpful to identify spoofed signatures. Also, the GUI often startled or perplexed the participants.

To counter the lack of effective user strategies to detect email signature spoofing attacks and the resulting suscepti-

USENIX Symposium on Usable Privacy and Security (SOUPS) 2022.
August 7–9, 2022, Boston, MA, United States.

bility to these attacks, email clients should offer guidance to users so they perform the most effective checks and are deterred from making ineffective ones. The GUI should make affordances [21] immediately apparent. We believe a way forward is to follow the results pertaining to supporting developers [18] and offer guidance to developers of email clients in creating GUIs that actively support users in detecting attacks. The core contributions of our research are:

- We give an overview of the checks expert users apply in their strategies to verify email signatures (section 5.1) and assess the usefulness of these checks to detect weak forgery attacks (section 5.2).
- We present the first upper bound baseline regarding expert users' performance in email signature spoofing detection (section 5.3).
- We present an overview of usability issues and how these prevent effective detection of spoofed signatures and instead increase user risk and uncertainty (section 5.5).
- We make the study materials, research artifacts, and evaluation tools available as open-source.[1]

2 Background on Email and OpenPGP

Emails [5] consist of two parts, a header and a body, where the header is a list of (name, value) fields and the body is ASCII text. The header contains the sender address, recipient address, and other metadata, while the body contains the actual content of the message. With MIME [9], emails internally become a tree-structure that can contain not only text but also other data types such as images, attachments, and digital signatures as defined in the OpenPGP standard [3, 7].

Verifying Email Signatures When rendering an email, the client has to clearly communicate each signature's validity, origin, and scope through the GUI to the user. This can be very difficult. Most email clients do not attempt to handle all signed parts at any layer but instead support only a single signed element, omit the scope of the signature, omit information about the signer, or otherwise simplify the process. Such clients require additional security checks.

OpenPGP Signer vs. Email Sender OpenPGP does not require that the signer's identity is identical to the sender in the email header. A secure email client should either enforce that the sender and signer have the same email address, in which case they can omit the signer identity from the signature verification result, or include the signer identity in the result, in which case the user is responsible for checking it.

OpenPGP Key Management Any digital signature could have been generated by anyone at first sight. To make signatures useful in the context of email, the signing key has to be bound to a user identified by an email address. Early OpenPGP implementations favored decentralized key management requiring manual validation, either directly or through the Web of Trust. Today, many users expect automatic key validation, and the most popular solution is the centralized key server *keys.openpgp.org* with 290k keys (Feb. 2022), where the email address is validated by sending a registration link. In addition, various domain-based proposals exist, such as DNS TXT records, DNSSEC/DANE [40], or HTTPS via the Web Key Directory (WKD).

3 Related Work

Human Aspects of Secure Email In their seminal work in 1999, Whitten and Tygar [38] evaluated the usability of PGP 5.0 in the Eudora email client with a cognitive walkthrough and user test (12 novice users). They demonstrated several serious usability issues. Follow-up works by other authors have studied PGP 9.0 in Outlook Express (pilot study with six novice users) [32], PGP support in Mailvelope (20 student participants) [28], and PGP support in Outlook 2016, Thunderbird and Maildroid (12 participants) [23], as well as with Enigmail and Mailvelope (52 non-technical participants) [16]. Due to these usability issues, it was found that while users want to use secure email [25] and find it important [22], adoption of email standards like OpenPGP and S/MIME is low.

Besides usability issues, the key management is often identified as a reason for the low adoption [22, 35]. One proposed mitigation of these key management issues is the automation of the related tasks [2, 11, 27]. For example, Garfinkel et al. [10] propose to accept all keys and only notify the user if the key differs from a previously used one. However, automation can have negative effects, as Ruoti et al. [29] note. Another solution proposed by Roth et al. [24] is rather to use in-person verification than trust certificate authorities. Lerner et al. [15] proposed combining this social approach with automation using Keybase, a service allowing users to link their public keys and social media accounts. Their proposal "Confidante" was well received by the study participants and reduced the time spent on key management while reducing the number of critical errors. Unfortunately, none of these proposals have been adopted, which means that the key management issues remain. The focus of our paper, however, is on the potential usability issues for expert users.

Several researchers have investigated how the usability issues can be addressed. Tolsdorf and Lo Iacocno [37] proposed to use persuasive design to improve the design of secure email GUIs. Ruoti et al. [26] found several ways to increase understanding of email encryption: a short delay and dialogue when encrypting or decrypting emails, a dedicated composer for encrypted emails (separate from the composer for unencrypted emails), and tutorials. Lausch et al. [14] analyzed the usability of novel security indicators in email clients and identified envelopes, torn envelopes, and postcards as promising

[1] https://github.com/SECUSO/email-signature-expert-study

candidates for future designs. However, text indicators might be enough: Stransky et al. [34] found in their comparison of several security indicators that simple text labels such as "encrypted" are as effective. Furthermore, their results indicate that icons can even lead to negative perceptions of the users. Gaw et al. [12] give an example of how a bad design can lead to annoyance. They found that the practice of connecting the encryption status of an email to the urgency status of that email led users to avoid encryption for regular emails.

One may argue that McGregor et al. [17] already studied expert users in the context of secure email communication. However, their focus was on encryption, while our focus is on signed emails and expert users' ability to detect email signature spoofing attacks. In their investigation of the tools used by journalists in the year-long "Panama Papers" project, they found that the tools used were perceived as highly usable and useful by the involved journalists, allowing them to meet confidentiality goals for the entire duration of the project.

Email Signature Spoofing Research indicates that signatures might be at least as desirable for users as encryption. Reuter et al. [22] found that the primary concern in terms of secure email is protection against others impersonating a trustworthy sender. The authenticity provided by digital signatures can fulfill exactly this role.

Müller et al. [19] describe three classes of weak signature spoofing attacks that can be detected by users of email clients: (1) *UI Attacks* are directed at the presentation of signature validation results in the email client. The attacker crafts an email containing an image that mimics a legitimate signature validation. (2) *ID Attacks* are directed at the potential mismatch between the sender and the signer of an email. An attacker creates a legitimately signed email with the attacker's key and then manipulates the email headers such that the signature looks like it was made by the sender instead. (3) *MIME Attacks* are directed at the complex MIME processing in email clients. The attacker gets a legitimately signed email from the victim and then constructs a new email that shows the same signature for a different content.

Other attacks on email signatures include covert content attacks [20], where an attacker attempts to acquire legitimate signatures unbeknownst to the signer, and spoofing attacks at the transport level for DKIM signatures [4]. However, these two types are not in the scope of our research.

4 Methodology

This research aims to investigate the strategies of expert users when deciding whether a signature is legitimate (i.e., a valid signature from the correct sender) and which individual checks these strategies comprise.

4.1 Research Questions

Our investigation is guided by five research questions:

RQ1 [Checks & Strategies]

(a) *Which checks do experts of OpenPGP email signatures in Thunderbird apply to discern legitimate from illegitimate signatures?*
(b) *How does the participants' overall strategy for the application of the checks look like?*

RQ2 [Usefulness of Checks]

(a) *Which checks helped participants to correctly discern legitimate from illegitimate email signatures?*
(b) *Which checks did not help participants to correctly discern legitimate from illegitimate email signatures?*
(c) *Which checks used by the participants pushed participants to incorrect decisions when discerning legitimate from illegitimate email signatures?*

RQ3 [Performance of Participants]

Were experts successful in detecting attacks, i.e., discerning legitimate from illegitimate email signatures?

RQ4 [Predictability of Success]

(a) *Is it possible to predict the outcome of discerning legitimate from illegitimate email signatures based on the outcome of the SA-6 scale?*
(b) *Is it possible to predict the outcome of discerning legitimate from illegitimate email signatures based on the outcome of the RSeBIS scale?*
(c) *Is it possible to predict the outcome of discerning legitimate from illegitimate email signatures based on self-reported expertise with email signatures?*
(d) *Is it possible to predict the outcome of discerning legitimate from illegitimate email signatures based on the self-reported frequency of OpenPGP usage in Thunderbird?*

RQ5 [User Perceptions]

How did participants perceive the process of investigating the legitimacy of message signatures?

4.2 Study Design

Two pre-studies informed the design of our main study.

4.2.1 Ethics

While our institutions did not mandate ethical approval for this study, our study fulfills all requirements of our institutions regarding studies with humans. The study procedure and data collection was approved by the data protection authority, also ensuring data minimization. The study had an informed consent form (see appendix A.1), explaining how to withdraw from the study and including a privacy policy. Participants received a debriefing, where the attacks were explained and any remaining concerns or questions of the participants were addressed. Additionally, we provided our contact data to participants in case of further questions or concerns. The video and audio recordings, as well as the questionnaire responses, were encrypted when stored or in transit. Only researchers approved by our data protection authority had access.

4.2.2 First Pre-Study

The goal of the first pre-study was to identify the most widely used email clients and signature standards among our target participant group of expert users. This study was held at the summer camp 2019 of the Chaos Computer Club (CCC) in Germany, attended mainly by IT security enthusiasts and internet activists. Participants were recruited by approaching them on the campground and asked on the spot how often they use OpenPGP and S/MIME signatures and which email clients they use. We collected data from 23 participants. The results showed that Thunderbird with Enigmail was the most popular option, with 12 participants stating to use it regularly. Other options might have yielded insufficient sample sizes in our main study. Therefore, we focused on OpenPGP signatures in Thunderbird with Enigmail for our study.

4.2.3 Selection of Attacks for the Study

Based on the decision to focus on OpenPGP signatures in Thunderbird (68.4.1) with Enigmail (2.1.5), we designed a set of four attack emails with illegitimate signatures for this specific scenario. These four attack emails are loosely based on the "weak forgery" class described in [19]. Our study covered the UI redressing and ID attacks, and we added a new typo-domain case. Due to time constraints, MIME attacks were excluded. Other attacks in [19] were perfect forgeries at the cryptographic API layer and not relevant to our study. In detail, the eight used emails were as follows (cf. figure 1):

Legitimate (4x) Email with a legitimate signature. Enigmail shows a green bar "Good signature from Bob <bob@code-audit.org>". An extended view of this email using all GUI elements is depicted in figure 2.

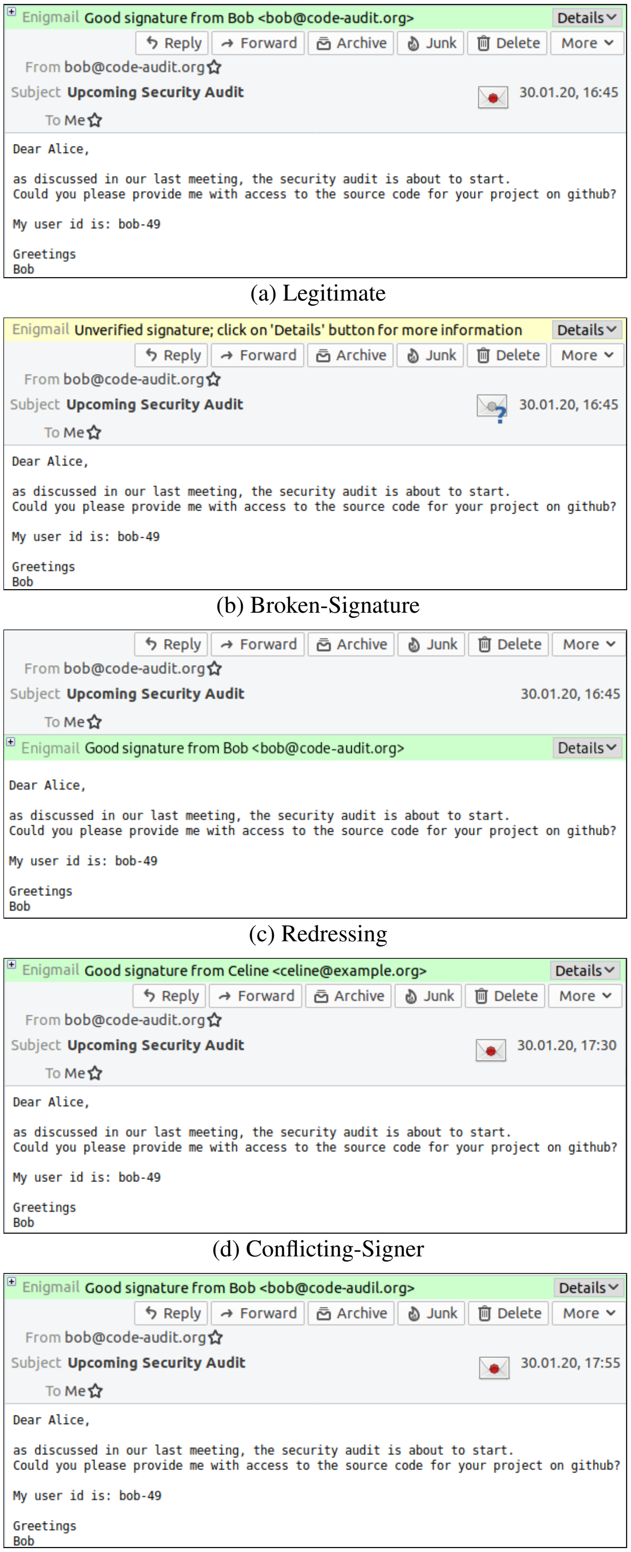
Broken-Signature (1x) Email with a broken signature. Enigmail shows a yellow bar "Unverified signature".

Redressing (1x) Email with an inline image of Enigmail's original green bar at the top of the email body. The simulated bar shows a green bar "Good signature from Bob <bob@code-audit.org>" and scales with the window but does not react to mouse clicks.

Conflicting-Signer (1x) Email signed by a different, easy to spot identity. Enigmail shows a green bar "Good signature from Celine <celine@example.org>".

Conflicting-Signer-Subtle (1x) Email signed by a different, hard to spot identity. Enigmail shows a green bar "Good signature from Bob <bob@code-audil.org>".

These messages are meant to imitate the work of an attacker, who can can send and arbitrarily modify email messages. They can also create new identities and have public keys for these new identities placed as trusted in Alice's keychain (to emulate key validation automation like WKD).



Enigmail Good signature from Bob <bob@code-audit.org>
From bob@code-audit.org
Subject **Upcoming Security Audit** 30.01.20, 16:45
To Me

```
Dear Alice,

as discussed in our last meeting, the security audit is about to start.
Could you please provide me with access to the source code for your project on github?

My user id is: bob-49

Greetings
Bob
```

(a) Legitimate

Enigmail Unverified signature; click on 'Details' button for more information
From bob@code-audit.org
Subject **Upcoming Security Audit** 30.01.20, 16:45
To Me

```
Dear Alice,

as discussed in our last meeting, the security audit is about to start.
Could you please provide me with access to the source code for your project on github?

My user id is: bob-49

Greetings
Bob
```

(b) Broken-Signature

From bob@code-audit.org
Subject **Upcoming Security Audit** 30.01.20, 16:45
To Me
Enigmail Good signature from Bob <bob@code-audit.org>

```
Dear Alice,

as discussed in our last meeting, the security audit is about to start.
Could you please provide me with access to the source code for your project on github?

My user id is: bob-49

Greetings
Bob
```

(c) Redressing

Enigmail Good signature from Celine <celine@example.org>
From bob@code-audit.org
Subject **Upcoming Security Audit** 30.01.20, 17:30
To Me

```
Dear Alice,

as discussed in our last meeting, the security audit is about to start.
Could you please provide me with access to the source code for your project on github?

My user id is: bob-49

Greetings
Bob
```

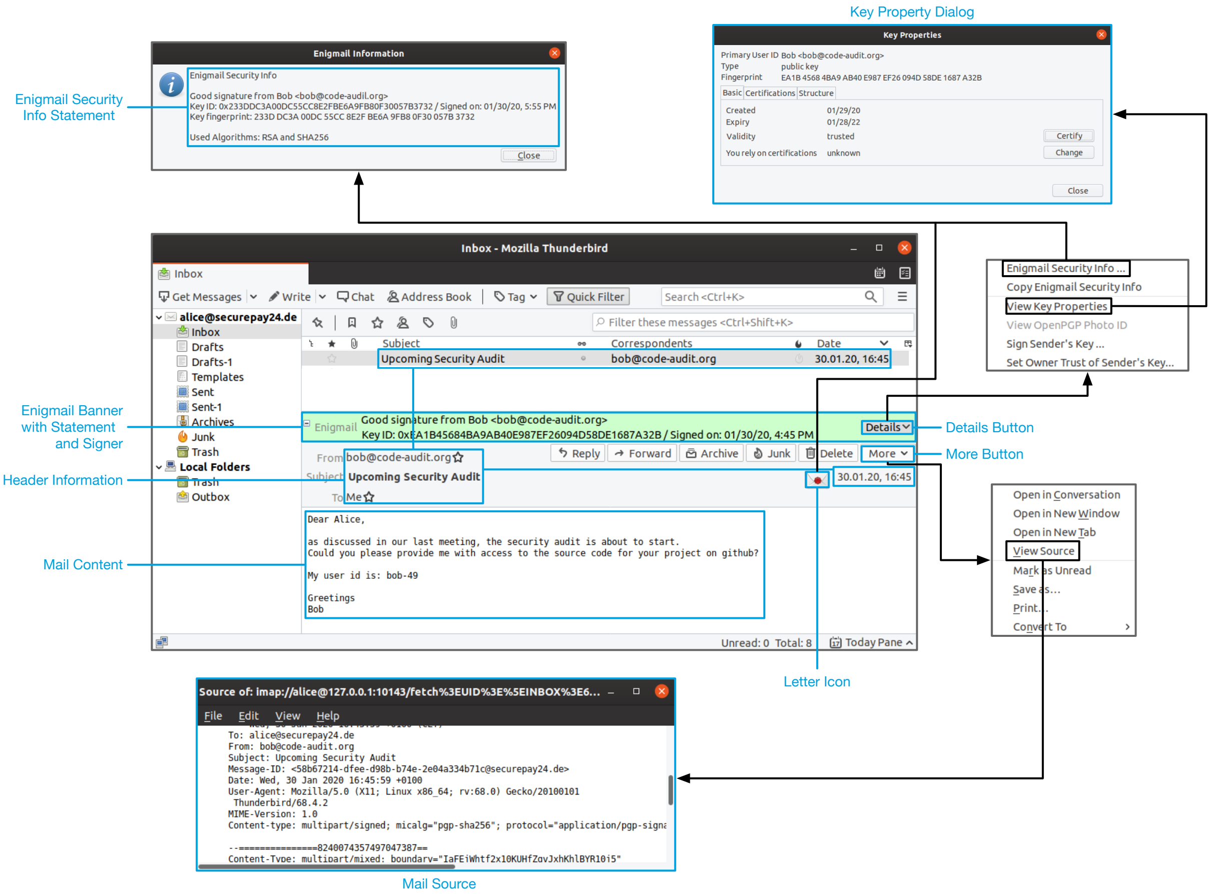
(d) Conflicting-Signer

Enigmail Good signature from Bob <bob@code-audil.org>
From bob@code-audit.org
Subject **Upcoming Security Audit** 30.01.20, 17:55
To Me

```
Dear Alice,

as discussed in our last meeting, the security audit is about to start.
Could you please provide me with access to the source code for your project on github?

My user id is: bob-49

Greetings
Bob
```

(e) Conflicting-Signer-Subtle, note the 'l' instead of 't'

Figure 1: Legitimate email and attack emails as displayed in Thunderbird 68.4.1 using Enigmail 2.1.5.



Enigmail Security Info

Good signature from Bob <bob@code-audil.org>
Key ID: 0x233DDC3A00DC55CC8E2FBE6A9FB80F30057B3732 / Signed on: 01/30/20, 5:55 PM
Key fingerprint: 233D DC3A 00DC 55CC 8E2F BE6A 9FB8 0F30 057B 3732

Used Algorithms: RSA and SHA256

Primary User ID Bob <bob@code-audit.org>
Type public key
Fingerprint EA1B 4568 4BA9 AB40 E987 EF26 094D 58DE 1687 A32B

Created 01/29/20
Expiry 01/28/22
Validity trusted
You rely on certifications unknown

Upcoming Security Audit — bob@code-audit.org — 30.01.20, 16:45

Good signature from Bob <bob@code-audit.org>
Key ID: 0xEA1B45684BA9AB40E987EF26094D58DE1687A32B / Signed on: 01/30/20, 4:45 PM

From bob@code-audit.org
Subject Upcoming Security Audit
To Me
30.01.20, 16:45

```
Dear Alice,

as discussed in our last meeting, the security audit is about to start.
Could you please provide me with access to the source code for your project on github?

My user id is: bob-49

Greetings
Bob
```

Source of: imap://alice@127.0.0.1:10143/fetch%3EUID%3E%5EINBOX%3E6...

```
To: alice@securepay24.de
From: bob@code-audit.org
Subject: Upcoming Security Audit
Message-ID: <58b67214-dfee-d98b-b74e-2e04a334b71c@securepay24.de>
Date: Wed, 30 Jan 2020 16:45:59 +0100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
 Thunderbird/68.4.2
MIME-Version: 1.0
Content-type: multipart/signed; micalg="pgp-sha256"; protocol="application/pgp-signa

--===============8240074357497047387==
Content-Type: multipart/mixed; boundary="IaFEiWhtf2x10KUHfZgvJxhKhlBYR10i5"
```

Figure 2: Overview of the Thunderbird 68.4.1 and Enigmail 2.1.5 GUI.

Our study aimed to evaluate the email client security indicators. All test emails were constructed such that their legitimacy could be deduced by only using the GUI elements. It was not required to inspect the source code or make further assumptions about the email context to identify the invalid signatures. All email messages had identical headers and text to ensure that our expert users focus on the available GUI elements when evaluating email signature's legitimacy. The GUI elements are depicted in figure 2. Additional technical descriptions of the attacks can be found in appendix C.

To our knowledge, there are no well-established strategies what security checks users should perform and in which order. The following strategy would at least uncover the attacks in this study: First, check that the banner shows a valid signature. Second, check that the banner is the correct indicator in this email client and that the banner is at the right location. Third, check that the signer and the sender are identical.

4.2.4 Study Procedure

Our goal in designing the study procedure was to allow the participants as much freedom as possible and to perform all the checks they normally would and capture their thoughts. Therefore, we decided to use a think-aloud protocol [39] and have the participants perform all study tasks on a prepared study laptop, where they could inspect all emails in a fully functional Thunderbird instance. All instructions and questionnaires were shown in a Firefox browser on this laptop and were implemented as surveys on the SoSciSurvey[2] platform.

Our study consisted of four parts (see figure 3). A Python script automated progression between the parts and started the screen and audio recording at the start of the third part.

Part 1 - Informed Consent and Explanations The participants had to consent to their participation and the analysis of their data (cf. appendix A.1.1). They received the instructions (cf. appendix A.1.2), including that their task would be to assess the legitimacy of email signatures. They were thus fully primed and the detection rates represent upper bounds. We discuss this design decision in section 4.4. To progress to the second part and start the actual decision tasks, the participants had to close the Firefox browser (cf. figure 3).

Part 2 - Introductory Questionnaire Participants had to fill an introductory questionnaire (see appendix A.2). It

[2] https://www.soscisurvey.de/

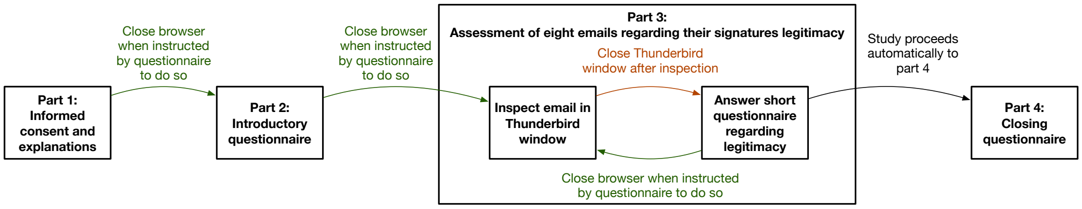

Part 1: Informed consent and explanations
Close browser when instructed by questionnaire to do so
Part 2: Introductory questionnaire
Close browser when instructed by questionnaire to do so
Part 3: Assessment of eight emails regarding their signatures legitimacy
Close Thunderbird window after inspection
Inspect email in Thunderbird window
Answer short questionnaire regarding legitimacy
Close browser when instructed by questionnaire to do so
Study proceeds automatically to part 4
Part 4: Closing questionnaire

Figure 3: Overview of the four parts of our study.

queried whether the participant had participated in one of our previous studies. Those participants were not eligible to proceed. Furthermore, the questionnaire included five questions to measure the participants' self-reported expertise with email encryption and signatures and one question on how often they use OpenPGP encrypted and signed emails in Thunderbird on average. The questionnaire ended with instructions for the third part. Therein, participants were asked to vocalize all thoughts and describe what they are doing during the assessment tasks, beginning from the moment they see the first email up to the point when they have made their decision regarding the legitimacy of the last of the emails. The detailed instructions can be found in appendix A.1.2. Again, participants would proceed to the next part by closing the Firefox browser displaying the questionnaire (cf. figure 3).

Part 3 - Assessment of Emails The participants were instructed to judge if a given email message was legitimately signed by bob@code-audit.org. All participants saw all eight emails listed in section 4.2.3 in a random order. The email messages were shown one at a time. The random order served to minimize any ordering bias. The Thunderbird interface was reset after each email message so that there was only one message in the inbox at any time, and participants could not jump back and forth between messages. During this part of the study, we captured the laptop screen and recorded audio of the participants thinking aloud. If the participants did not say anything, the experimenters reminded the participants to vocalize and explain their thoughts and actions. After inspecting each email message, a short questionnaire popped up, in which the participants could indicate their decision about this message and optionally note any issues they encountered.

Part 4 - Closing Questionnaire The closing questionnaire included the Refined Security Behavior Intentions Scale (RSeBIS [30]) and the Security Attitudes scale (SA-6 [8]).

4.2.5 Second Pre-Study

We performed a second pre-study to pilot the study procedure described above. This second pre-study was held at the Chaos Communication Congress 2019, attended by an audience very similar to the summer camp. Participants were recruited by approaching attendants directly and handing out leaflets. Overall, we performed nine full runs of the OpenPGP study procedure. These runs allowed us to improve the setup and the emails the participants inspected. For example, some participants falsely classified emails as illegitimate due to missing trace headers, i.e., `Received`, or other artifacts that we did not anticipate. These issues were corrected for the final study and steered the participants towards focusing on the graphical security indicators. Also, we addressed a data recording issue preventing full recordings for the think-alouds.

4.2.6 Main Study

We conducted our main study at the Free and Open Source Developer Meeting (FOSDEM) in February 2020 in Brussels.[3] Like the CCC venues, this event is attended by IT specialists, but with a focus on Open Source rather than IT security.

Participants were recruited similarly to the second pre-study by approaching attendees directly and using leaflets. Additionally, FOSDEM 2020 hosted a room for Mozilla with a scheduled talk on Thunderbird development, and one of the co-located events at FOSDEM was a large OpenPGP key signing party. We used both of these opportunities for recruiting. If an attendee was interested in participating in our study, they were asked if they frequently use Thunderbird with Enigmail (OpenPGP) and if they already had participated in our pre-studies. Those that had were excluded. Similarly, participants who stated not to use Thunderbird with OpenPGP frequently were excluded. We then explained the study's goal and the task participants would have to perform.

Overall, we conducted think-aloud sessions with 33 participants. Of these 33 participants, two had to be excluded since their questionnaire data indicated they did not use Thunderbird, three were excluded since they could not be considered expert users (scored lower than 2 on average in our self-reported expertise questions with no individual value larger than 2), two were excluded due to interruptions by third-party attendees, and one was excluded due to missing consent (presumably in error). This left us with 25 valid recordings of think-aloud sessions of expert users classifying signatures.

4.3 Analysis

Qualitative Analysis The think-aloud recordings were transcribed, including the mouse cursor actions and dialogues appearing on screen. Then qualitative analyses were performed

[3] Note that this was before the COVID-19 pandemic started, and in-person studies were still unproblematic: https://archive.fosdem.org/2020/

using an inductive coding approach [36] with two independent coders. The two coders created the codebook based on the research questions (cf. section 4.1) and an initial coding of three transcripts. Both coders coded five more transcripts to ensure inter-rater reliability (IRR). As a measure for IRR, Krippendorff's α was used. The value of $\alpha = 0.71$ indicates a moderate IRR, which is acceptable given our unstructured think-aloud data and the exploratory nature of our study. The remaining 17 transcripts were coded independently, eight by one coder and nine by the other. The coders met to discuss changes or additions to the codebook as they arose from newly coded transcripts. The final codebook contained 69 codes in seven categories (see appendix B for the full codebook).

Quantitative Analysis For the SA-6 and RSeBIS scales and our self-reported expertise questions, the mean of all values for each participant was used in the correlation analyses. For the frequency of use, the answer for each participant was normalized to days per year.

4.4 Limitations

Our participants were sampled from a non-diverse group of people attending FOSDEM in person. As the conference was in Brussels, we expect the participants to be primarily from Belgium and adjacent countries. Therefore, our results might not generalize to other populations. The quantitative results would benefit from a larger sample. Yet, further data collection was prevented by the onset of the COVID-19 pandemic.

Participants were self-selected based on leaflet advertising and word of mouth. We asked participants to not share details of the study with others, but could not control communication between the participants and attendees outside the study. We did not observe any reactions of one participant to another.

Our participants were explicitly tasked with identifying whether a given email signature was legitimate or not. Thus, they were likely to check more thoroughly than under real-world circumstances. Priming our participants in this way was intentional. We wanted to capture our participants' strategies validly even in the first email. We decided that priming our participants to use these strategies throughout the study would be the prudent way to collect this data. Our findings can thus be seen as an upper bound of the expert users' capabilities. Six participants even mentioned at least once during the experiments, after identifying an attack, that they might fall for this in real life: *"But I don't usually do these checks unless I know I am actively being targeted, like right now." -P6.*

For our qualitative analysis, we rely on think-aloud data, which does not guarantee a complete insight into our participants minds and reasoning. We cannot rule out that some checks were not verbalized and are thus missing in our data.

Finally, the delay in our research due to the COVID-19 pandemic has seen Enigmail being integrated into Thunderbird, and as a result, the GUI changed. We describe these differences and discuss their impact on our results in section 6.

#	Checks	n
	Not related to PGP signatures	
1	Header Information	113
2	Mail Content	41
3	Mail is Classified as Junk	1
4	Mail is Encrypted	1
	Related to Redressing Attacks	
5	GUI Behaves Unexpectedly	31
6	Alternative Message Views	8
	Related to Enigmail GUI	
7	Banner Indicator	116
8	Compare Signer to Sender	54
9	Security Info Statement	43
10	Fingerprint	39
11	Letter Icon Status	17
12	Banner Position	15
13	Banner Signer	15
14	Signature Date	14
15	Crypto Algorithms	4
	Related to Key Management	
16	Sender's Key	22
17	Signer's Key is Signed with Own Key	11
18	Key Property Trust Statement	8
19	Key Validity	6
20	Key is in Keyring	5
21	Keyring	5
22	Key Creation Date	3
	Mail Source	
23	Mail Source	86
	Proposed Checks	
1*	Compare Fingerprint to Known One	11
2*	Mail Source	7
3*	Fingerprint	6
4*	Out of Band Verification	4
5*	Recheck with GPG on Command Line	3
6*	Keyring	2
7*	Key Revocation	1
8*	Signature Date	1

Table 1: Overview of the checks applied by our participants and how often they were applied. The checks are grouped regarding the five categories that emerged from the coding and sorted in descending order of their frequencies. The "Proposed Checks" at the bottom of the table are the checks that participants talked about but did not perform.

5 Results

In the following, we present the results of our study regarding the five research questions outlined in section 4.1.

5.1 RQ1: Checks & Strategies

5.1.1 Checks Applied by Participants

We identified 23 distinct checks in the transcripts of the 25 think-aloud sessions (cf. table 1). The checks are generally named after the information or GUI element that is inspected by the participant. See figure 2 for an overview of Thunderbird's GUI. Participants applied on average 9.8 distinct checks (median: 10, sd: 2.5) across all emails. Overall, the 23 checks were applied 659 times by our participants, with an average of 3.3 (median: 3, sd: 2.0) checks per email.

From the qualitative coding, five categories of checks emerged: (1) checks based on information not related to OpenPGP signatures, (2) checks based on information related to redressing attacks, (3) checks based on information related to the Enigmail GUI, (4) checks related to key management, and (5) checks based on inspecting the email source code. In the following, we discuss the checks in each category. In addition to the checks applied during the study, participants also proposed additional ones. These proposed checks will be discussed at the end of this section.

Checks Not Related to OpenPGP Signatures These checks are not related to the signatures at all. They are based on inspecting information even found in emails without signatures. Checking the *Header Information* is the most frequently applied check in this category with 113 applications. It includes all the header information displayed in Thunderbird's GUI, e.g., sender, recipient, subject, or date and time. The *Mail Content* was inspected 41 times. The remaining two checks, i.e., whether the *Mail is Classified as Junk* and whether the *Mail is Encrypted* were both applied only once. None of these checks can detect the attacks in our study.

Checks Related to Redressing Attacks The following two checks are not related to email signatures but allow detecting the *Redressing* attack. The most frequently applied check is a reaction when the *UI Behaved Unexpectedly*, which occurred 31 times. Usually, the first encounter with the fake banner prompted this check. Most participants correctly interpreted the unresponsive GUI and adopted these checks for subsequent emails. The second check in this category is inspecting the message in an *Alternative Message View*, such as in plaintext, in simple HTML, or looking at what a reply would contain. This check was applied only eight times.

Checks Related to Enigmail GUI These checks directly relate to the information displayed by the Enigmail GUI. The most frequently applied check (116 times) is checking the *Banner Indicator*, i.e., the statement ("Good signature", "Unverified signature", etc.) and color of the Enigmail banner, which both essentially communicate the same information to the user. This check can detect the *Broken-Signature* attack. The second most frequently used check is to *Compare Signer and Sender* (54 applications). This check can detect two of the four attacks in our study (i.e., *Conflicting-Signer* and *Conflicting-Signer-Subtle*). Another two checks that were somewhat similarly often applied are inspecting the *Security Info Statement* (applied 43 times) and checking the *Fingerprint* (applied 39 times). The *Security Info Statement* displays information similar to the Enigmail banner and allows detection of the same attack. All remaining checks were applied less than 20 times. Of these, checking the *Letter Icon Status* is the most useful, allowing to identify the redressing attack with a first-level GUI element. However, this check is only possible if the user spots that this indicator is missing.

Checks Related to Key Management Some participants ventured beyond Thunderbird to perform checks related to their keychain. However, these checks were not used very frequently. Inspecting the *Sender's Key* (e.g., the existence of subkeys) is the most frequent check (22 applications) and checking whether the *Signer's Key is Signed with Bob's Key* is the second most frequently applied check (11 applications). All other checks were applied less than 10 times.

Checking the Mail Source Another relatively popular check (86 applications) was inspecting the *Mail Source*. While some participants just screened it in general, some inspected specific information, such as `Received` headers.

Proposed Checks Participants proposed several checks which they did not perform. Inspecting fingerprints is the most common theme, with 11 participants stating that outside the study setting they would try to *Compare the Fingerprint to a Known One* and another six participants stating that they might check the *Fingerprint* in more detail (without specifying how they would perform this check). Some of the checks were mentioned as potential further avenues but did not seem to be required at the time, for example: *"So, I see some stuff that I could look at if I was at all suspicious that I probably haven't been looking at before." -P14*. Other checks were not possible in the study, such as an *Out of Band Verification*: *"In this case I would call Bob on the phone." -P10*.

5.1.2 Overall Strategy for Application of Checks

Figure 4 exemplifies how our participants transitioned from one check to the next for the *Redressing* email. While a path with just two checks emerges, when following the transitions with the highest probabilities (participants realize that the *GUI Behaves Unexpectedly* and then inspect the *Mail Source*), figure 4 illustrates how participants did not seem to follow a pre-determined path for every mail. Only three of our 25 participants took this direct path, as expected from the probabilities. Instead, each new email was the start of a treasure hunt, as we watched our participants explore the Enigmail GUI. Consequently, there was a great variety in the order that checks were applied. Among the 42 transitions between checks we observed for the *Redressing* email, only in four instances are the transition probabilities above 50%. Often it seemed that participants were not sure what they were looking for next, as illustrated by P6 when they, after opening a dialog containing details of the signing key, uttered *"I don't know why I check this... "*. This lack of a common strategy is consistent across all attacks (cf. appendix F). There is one exception to this though: if a participant checks the *Mail Source*, it is most frequently the last check they perform. This is, however, contrasted by many checks with a high fan-out and similar probabilities for the subsequent checks.

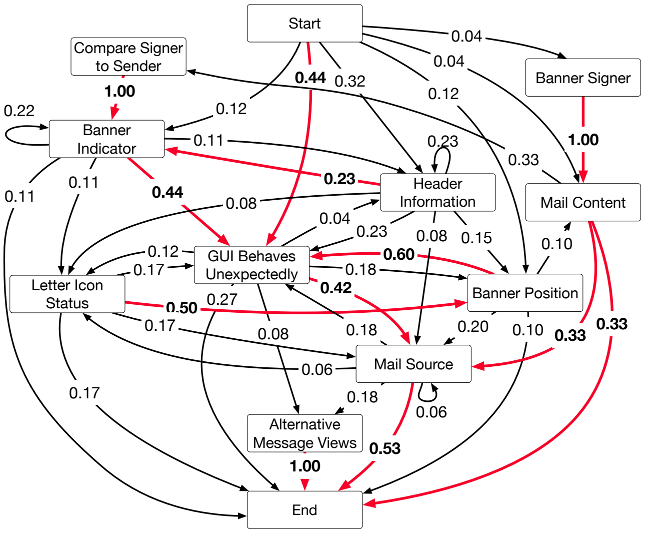


Figure 4: Participants' transition probabilities from one check to another for the *Redressing* email. The most likely transition after each check is drawn in bold and red. Due to rounding, the probabilities for each node might not add up to 100%.

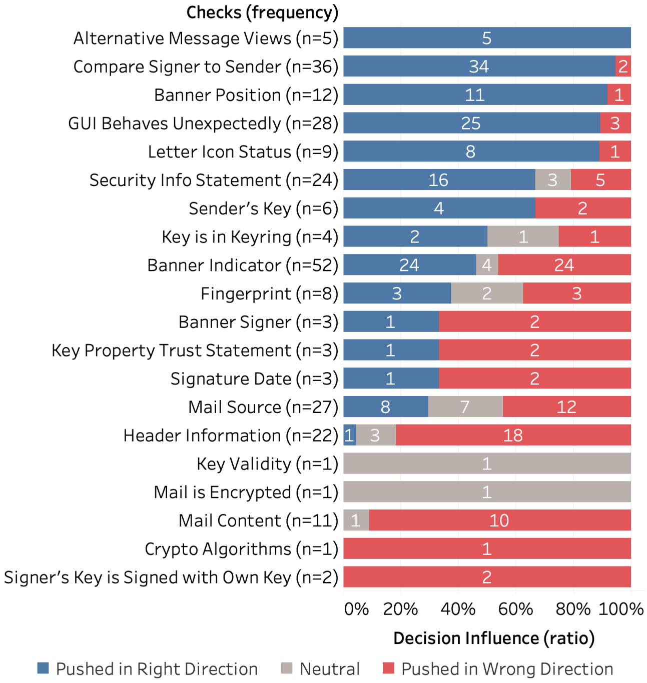


Figure 5: How the checks influenced participants' decision for the emails with illegitimate signatures. Checks from table 1 not appearing here were only applied in the legitimate case.

From inspecting the transition graphs of each email, a few additional relevant observations become apparent. For the *Broken-Signature* email, participants did not seem to trust the *Banner Indicator* check. Instead, when following the path of highest probabilities, participants would also check the *Security Info Statement*. This is noteworthy since the wording regarding the signature's validity differs slightly in these two checks. While the Enigmail banner reads "Unverified signature", the statement in the security info dialogue reads "Bad signature". The latter seems to have had a much stronger impact on the participants' decision. Participants' suspicion might also have been caused by them being primed.

For the *Redressing* attack, both first-level Enigmail GUI elements are present among the checks: The banner and the letter icon. However, while the (spoofed) banner is among the first elements our participants checked, the *Letter Icon Status* is only ever checked after other checks were performed. This illustrates how a missing security indicator poses problems and might not be recognized by the participants, which replicates findings from other domains [6, 31].

For the *Conflicting-Signer* and *Conflicting-Signer-Subtle* email, several participants needed only one check, namely to *Compare Signer to Sender*. For the *Conflicting-Signer* attack this even represents the path with the highest probability (cf. figure 11 in appendix F).

5.2 RQ2: Usefulness of Checks

In order to understand which checks contributed most to participants' detection of the attacks, we coded each of the checks regarding whether it pushed them towards the right decision, towards the wrong decision, or did seemingly not contribute to the decision (neutral). The latter case occurred in particular when participants could not interpret the information they checked, e.g.: *"I don't know what it what it [sic.] means, does it mean the signature does not match the content of the body or does it mean there is no trust part. That's unclear." -P16*. We leave instances where participants did not comment on a certain check out of the analysis to not introduce unnecessary interpretation and bias into our results.

Figure 5 gives an overview of how each check influenced the decision of the participants when inspecting the emails with illegitimate signatures. Unsurprisingly, the checks based on information not related to email signatures are among the least effective. Two of the checks based on information provided by Enigmail or information found in the keyring, i.e., checking the *Crypto Algorithms* and checking whether the *Signer's Key is Signed with Bob's Key*, did not prove useful to the participants either. However, they were rarely used.

The most frequently applied check *Banner Indicator* pushed our participants as often towards a correct decision as it did towards an incorrect decision. This points towards severe issues with this most prominent part of Enigmail's first-level GUI elements. The issues arise when we look at the decisions for the *Redressing*, *Conflicting-Signer*, and *Conflicting-Signer-Subtle* emails. In these attacks, the banner color is green, and the banner contains the text "Good signature". For the *Redressing* email, the *Banner Position* is the better check, but it requires the participants to know how the

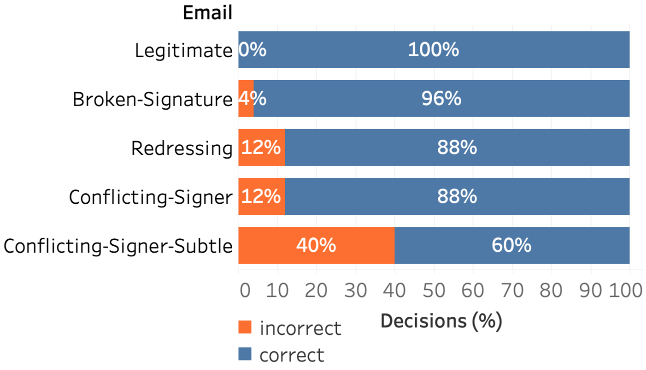


Figure 6: Overview of correct and incorrect classifications for the valid and each of the four attack emails.

		RSeBIS	SA-6	TE	FoU	CR
RSeBIS	ρ	1	**.760	.341	.226	−.195
	Sig.	–	< .001	.095	.278	.351
SA-6	ρ	**.760	1	**.586	.304	.014
	Sig.	< .001	–	.002	.139	.946
TE	ρ	.341	**.586	1	*.464	−.201
	Sig.	.095	.002	–	.019	.336
FoU	ρ	.226	.304	*.464	1	−.091
	Sig.	.278	.139	.019	–	.665
CR	ρ	−.195	.014	−.201	−.091	1
	Sig.	.351	.946	.336	.665	–

Table 2: Overview of the investigated Pearson correlations ρ. In all cases $n = 25$. Calculations are 2-tailed. */** marks significance at the .05/.01 level. TE = Technical Expertise, FoU = Frequency of Usage, CR = Ratio of Correct Responses

interface is supposed to look and to recognize the difference. For the other two emails, the "Good signature" statement is insofar misleading as it does not reflect the expectations of the participants as will be further discussed in section 5.5.

When participants *Compare Signer and Sender* it mostly guides them towards the correct decision. Making this comparison as easy as possible would greatly benefit detecting the corresponding attacks. The more reliable but far less frequently used first-level GUI element is the *Letter Icon*. It is placed in the header area, and a click on it leads to the also relatively helpful *Security Info Statement*. Thus it might provide a better template for future designs.

Checking the *Mail Source* also stands out: It pushed more participants towards an incorrect decision than a correct one. Specifically, participants misinterpreted the information they saw or inspected header information that did not help them. Also, this check exhibits the highest number of neutral ratings where participants could not interpret the information they saw, e.g., P17 pondered : *"Another weird segment. I am probably not knowledgeable enough regarding MIME parts."*

5.3 RQ3: Performance of Participants

Figure 6 shows an overview of the correct and incorrect responses for the valid mails as well as each of the attacks. It becomes apparent that our participants were fairly successful in discerning legitimate and illegitimate emails. However, overall 52% of participants failed to detect at least one of the attacks (average 0.76 attacks per participant, $median = 1$, $sd = 0.97$). Thus, these misclassifications are not due to repeated failures by a few participants, but they seem to be fairly evenly distributed among the participants.

When looking at the attacks individually, figure 6 clearly shows that the more intricate the attacks become, the more difficulties even expert users have. The *Conflicting-Signer-Subtle* attack was the most successful, with 40% of participants falling for it. This is likely due to two issues. Firstly, this attack is presumed to take place *after* the corresponding key for *bob@code-audil.org* was imported into the victim's keyring, e.g., by automated key retrieval such as WKD, leading to a green Enigmail banner signaling a valid signature to the victim. Secondly, the discrepancy between signer and sender was minimal, with the two differing in only one letter, which even looks similar at first glance. The effect of a more obvious discrepancy between signer and sender can be seen in our easier *Conflicting-Signer* case: only 12% of participants fall for this attack. Similarly, 12% of participants fall for the *Redressing* attack. The *Broken-Signature* email was still classified as legitimate by one participant because they found the key which was used to originally sign the (subsequently manipulated) email in their keyring.

5.4 RQ4: Predictability of Success

We wanted to investigate whether the participants' *security behavior intention* (RSeBIS scale), *security attitude* (SA-6 scale), *self-reported technical expertise*, or the *frequency of use* (uses of OpenPGP and Thunderbird per year) might be used as predictors of the ratio of correct responses for each participant. This investigation is considered exploratory, with correlations between RSeBIS, SA-6 and the participants' performance deemed not unlikely and the other two constructs being completely exploratory. Yet, from the correlation analysis with Pearson's ρ (cf. table 2), it becomes quickly apparent that none of the measures can serve as a meaningful predictor. In contrast, the measures seem to be predictors for each other, particularly for RSebis and SA-6 as expected [8].

5.5 RQ5: User Perceptions

We observed many participants blaming themselves for any possible errors or slips that might have decreased their success in labeling the messages correctly. E.g., P18 already took it onto themselves to write down the key fingerprint of Bob, but then still said they did not do enough due diligence: *"I* *did write down the SHA256 signature, and they didn't match. Something is fishy. And now I regret that I didn't write down the creation and expiration dates. Insufficient due diligence." -P18* From a usable security perspective, this seems absurd – a tool should never expect the user to write down or compare dates or long strings when it could just do it itself.

P24 was able to point out that in our Conflicting-Signer-Subtle email the signer's address had a typo. However, they decided to label the message as legitimately signed anyway, since they perceived Enigmail's statement "valid signature" to be trustworthy, outweighing their concerns about address inconsistencies: *"I think this email signature is legit [selects "legitimate signature"]. However, the email header was somehow, um... worked with. It's just a guess, because I assume that Enigmail has also correctly verified the signature that is shown as correct. Where the discrepancy with the email address From-field comes from, I just don't know." -P24*. This was, however, the only instance where a participant noticed the address inconsistency but still went on to labeled the message as legitimately signed.

Four participants were upset when they realized that Enigmail was not pointing out that that signer was different from the sender, e.g.: *I would say this one is legit. [Pause] Except that it is signed by Celine... What?! OK. That is quite strange that Thunderbird does not claim anything about, like, it's signed by a different guy than the sender! -P3*. The remaining participants who noticed inconsistencies between signer and sender were mostly confused or insecure and could not pinpoint where the issue was exactly. However, they were able to recognize that *something* was off and thus labeled the message as illegitimate, accordingly.

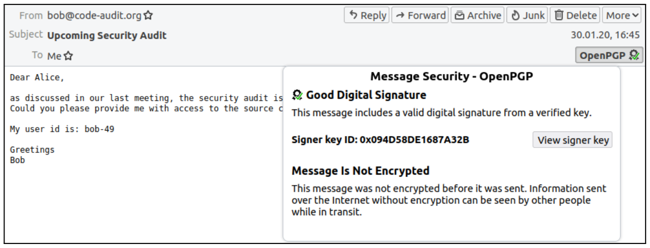
As mentioned in section 5.1.2, our participants did not seem overly determined by neither having a certain, strict click-path nor a set of known indicators to look for. Instead, our participants were often startled, confused, or even annoyed when navigating the GUI elements offered to them by Enigmail and Thunderbird. P23 simply gave up on finding more information on the keys in their key chain after looking for, but not finding it, in three different places: *"[After a very long search through Thunderbird's settings, looking for PGP Keys] Personally this is taking too long for me right now. [closes settings] That's why I cancel this [clicks on Inbox in folder selection] and would claim the email is just not trustworthy and stick to my first impression."*

In summary, even for our sample of expert users, the task of recognizing illegitimate OpenGPG signatures is generally accompanied by haphazardness and uncertainty.

6 Changes in Newest Thunderbird Version

Our study was conducted in 2020 with Thunderbird 68.4.1 and Enigmail 2.1.5. Since then, Thunderbird 78.2.1 has been released with built-in OpenPGP support [33]. Thus, we re-evaluated the presented attacks with the newest version of Thunderbird (91.5.0) and discuss which study results are relevant to the newest version, or for email signatures in general.

Overall Assessment of Changes Figure 7 shows a valid email in Thunderbird 91.5.0. It uses new design elements and the signature validity status is not as prominent as in previous versions (cf. figure 1). The Enigmail banner and the letter icon were replaced with one button in the header area. The button is labeled "OpenPGP," and an icon shows the signature status in green color. Upon pressing the button, a new dialogue appears. It contains a short signature status statement, the signer key ID, and a button that allows inspecting the key and the encryption status. While the newer interface is cleaner and contains just one first-level and one second-level GUI element, we also see negative properties. For example, the colored area in previous versions was much larger (cf. figure 1), which made the email validity more immediately apparent.



From bob@code-audit.org

Subject Upcoming Security Audit

To Me

30.01.20, 16:45

```
Dear Alice,

as discussed in our last meeting, the security audit is
Could you please provide me with access to the source c

My user id is: bob-49

Greetings
Bob
```

Message Security - OpenPGP

Good Digital Signature

This message includes a valid digital signature from a verified key.

Signer key ID: 0x094D58DE1687A32B

Message Is Not Encrypted

This message was not encrypted before it was sent. Information sent over the Internet without encryption can be seen by other people while in transit.

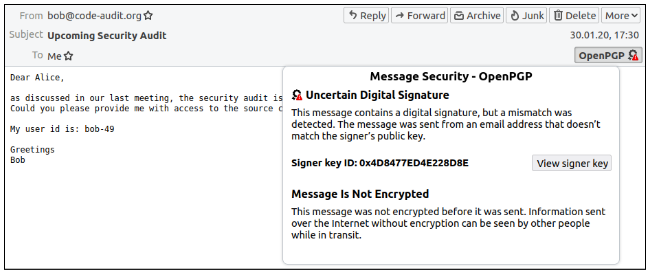
Figure 7: Legitimate email displayed in Thunderbird 91.5.0.

Relevance for Broken-Signature Email (Figure 9) The validity status became clearer. The wording is now "Invalid" email instead of "Unverified" email as it was previously. Also, the red color in the icon signifies the email invalidity. On the negative side, the colored area is much smaller.

Relevance for Redressing Email Our Redressing email has a now obsolete design and would not work in current versions of Thunderbird. Yet, since the validity indicator is not being displayed for unsigned emails in newer Thunderbird versions as well, the base issue remains. More research is needed to determine the viability of such attacks in the new GUI.

Relevance for Conflicting-Signer and Conflicting-Signer-Subtle Email (Figure 8) Detection of conflicting signer and sender got easier in newer Thunderbird versions. A red lock directly shows if the signer is not equal to the sender. Clicking the OpenPGP symbol reports *Uncertain Digital Signature* and allows the user to review the signer's key, which can make the detection of these attacks easier for users.

Yet, a bug allows bypassing this security indicator. By using two `From` headers (a technique from [19]), we were able to have Thunderbird 91.5.0 display a green icon. Our analysis revealed that the first `From` header was used to display the message sender. The second `From` header was used for the message sender validation and for displaying the name of the user in the list of emails. Thus, the second `From` header also



From bob@code-audit.org

Subject **Upcoming Security Audit**

To Me

30.01.20, 17:30

```
Dear Alice,

as discussed in our last meeting, the security audit is
Could you please provide me with access to the source c

My user id is: bob-49

Greetings
Bob
```

Message Security - OpenPGP

Uncertain Digital Signature

This message contains a digital signature, but a mismatch was detected. The message was sent from an email address that doesn't match the signer's public key.

Signer key ID: 0x4D8477ED4E228D8E

Message Is Not Encrypted

This message was not encrypted before it was sent. Information sent over the Internet without encryption can be seen by other people while in transit.

Figure 8: Thunderbird 91.5.0 notifies users about an *uncertain* digital signature when signer and sender differ.

includes a display name of Bob. This attack is only detectable by investigating the message signer in the third-level GUI (after clicking on the OpenPGP icon and the icon *View signer key*). We reported this issue to the Thunderbird developers, who plan a fix in the upcoming release.

Thunderbird 91.5.0 implements a custom key management instead of relying on the local GPG keyring. Thus, possibly insecure configurations where keys are side-loaded and trusted by a mechanism like WKD are less likely. Yet, OpenPGP key validation remains an open problem for Thunderbird users.

7 Discussion

Implications Thunderbird and Enigmail as used in our study have since been replaced with a built-in solution. Yet, our most important result remains unimpaired by this change: Even our expert user participants had no effective strategies to assess whether email signatures were legitimate or not. Instead, participants explored the interface as they went and exhibited much uncertainty about what to check. Some of the most frequently performed checks are of questionable usefulness, e.g., inspecting the *Mail Source*.

However, the users are not to blame here. We saw users baffled by the GUI or overwhelmed by the complexity of the necessary checks. The GUI needs to give meaningful support to the user when they need to perform these complex checks with obvious affordances [21] that invite to perform useful checks and deter from performing unuseful ones. This becomes particularly apparent when sender and signer differ. This discrepancy is not highlighted in the Conflicting-Signer and Conflicting-Signer-Subtle emails. This was perplexing for users, and we agree with this assessment. The problem seems to be that the signature is technically valid (i.e., no manipulation of the email), but the email context carries the additional expectation that it is only legitimate if it was signed by the sender. Honoring these expectations is what developers should strive for, and supporting developers in achieving this task by mapping out these expectations in an easily digestible way is the future work ahead of us as research community. Our work also highlights that email client GUIs need trustworthy zones where security status indicators can reside to impact the viability of *Redressing* attacks. Future work is needed to formulate proper guidelines in this respect.

Also, our research supports the results of earlier studies. Most checks relating to key management, e.g., checking the *Key Validity*, leave at least as many participants in uncertainty or lead them to incorrect decisions as they helped participants. Similarly, the signature GUI seems geared towards checking for simple manipulations, not more sophisticated attacks. In both cases (key management and usability issues), our work extends the existing research, which reports on the usability issues surrounding encryption and digital signatures. Yet, the "upper bound" detection rates in our results due to the priming of our participants underlines the severity of these issues.

Recommendations We believe that the proper long-term solution for end-to-end email security is shifting the ecosystem from *indicating secure messages* to *warning about (potentially) insecure messages*. This is important, because the absence of security indicators is often overlooked by users [6, 31]. The security of systems should not rely on users checking for the presence of indicators. However, to avoid warning fatigue [1], this shift can only happen after end-to-end secured email has become the default for email communication. In this chicken-and-egg problem, it is up to current tools to help adoption by implementing these security features as usable as possible.

Overcoming the complexity of checking a signature's legitimacy before hand-off to the user plays a key role. We believe an approach based on allowlists of secure MIME structures, as described in [13] to classify emails is key to achieving this. In particular harnessing the power of crowd-sourcing to maintain and extend such allowlists seems like a desirable approach. Based on these allowlists, we envision that email clients automate as many checks as possible and that interfaces distinguish four cases: (a) the *legitimate case*, where the signed email's structure is in the allowlist and the signature is validly signed by the sender; (b) the *illegitimate case*, where the signed email's structure is in the allowlist, but the signature is not valid or not from the sender; (c) the *check case*, where the signed email's structure is not in the allowlist and the GUI has to support the user in performing useful checks; and (d) the *unsigned case*, where the email is not signed.

Coloration to distinguish the cases may support users. Yet, the colors should be chosen to be accessible by users with colorblindness.[4] Also, the fourth case should not be skipped as is currently the case for most email clients. Such "missing indicators" rely on the user realizing that the indicator is not there, which has been proven to be problematic. This would introduce a source of conflicting information for Redressing attacks and thus make them easier to spot for users.

[4] E.g. https://davidmathlogic.com/colorblind/ can be used to choose suitable colors.

Acknowledgements

We thank the organizers of the Free and Open Source Developer Meeting 2020 for providing an interview location on the premises of the event and allowing us to recruit participants from among the attendees. We also thank Imke Ines Klatt for her feedback on the referee instructions, recruiting, and execution of participant sessions during the 36c3. We thank Tobias Kappert and Christian Dresen for the execution of the first pre-study at the Chaos Communication Camp 2019, Martin Grothe for his support during the interview process at FOSDEM, Jonathan von Niessen for his security analyses of signature validation in Thunderbird, as well as Christiane Rosa, Marie-Claire Thiery, and Fabian Ballreich for their help in transcribing the think-alouds.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA – 390781972. Marcus Brinkmann was partially supported by the German Federal Ministry of Economics and Technology (BMWi) project "Industrie 4.0 Recht-Testbed" (13I40V002C). Damian Poddebniak was supported by a research grant of the Münster University of Applied Sciences. This research was further supported by funding from the topic Engineering Secure Systems, subtopic 46.23.01 Methods for Engineering Secure Systems, of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

References

[1] Devdatta Akhawe and Adrienne Porter Felt. Alice in warningland: A Large-Scale field study of browser security warning effectiveness. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 257–272, Washington, D.C., August 2013. USENIX Association.

[2] Wei Bai, Doowon Kim, Moses Namara, Yichen Qian, Patrick Gage Kelley, and Michelle L. Mazurek. Balancing Security and Usability in Encrypted Email. *IEEE Internet Computing*, 21(3):30–38, 2017.

[3] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer. OpenPGP Message Format. RFC 4880 (Proposed Standard), November 2007. Updated by RFC 5581.

[4] Jianjun Chen, Vern Paxson, and Jian Jiang. Composition kills: A case study of email sender authentication. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2183–2199. USENIX Association, August 2020.

[5] D. Crocker. STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES. RFC 822 (Internet Standard), August 1982. Obsoleted by RFC 2822, updated by RFCs 1123, 2156, 1327, 1138, 1148.

[6] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Conference on Human Factors in Computing Systems, pages 581–590, 2006.

[7] M. Elkins, D. Del Torto, R. Levien, and T. Roessler. MIME Security with OpenPGP. RFC 3156 (Proposed Standard), August 2001.

[8] Cori Faklaris, Laura A. Dabbish, and Jason I. Hong. A Self-Report Measure of End-User Security Attitudes (SA-6). In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, Santa Clara, CA, August 2019. USENIX Association.

[9] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045 (Draft Standard), November 1996. Updated by RFCs 2184, 2231, 5335, 6532.

[10] Simson L Garfinkel, David Margrave, Jeffrey I Schiller, Erik Nordlander, and Robert C Miller. How to make secure email easier to use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Conference on Human Factors in Computing Systems, pages 701–710, 2005.

[11] Simson L. Garfinkel and Robert C. Miller. Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express. In *Proceedings of the 2005 Symposium on Usable Privacy and Security*, SOUPS '05, page 13–24, New York, NY, USA, 2005. Association for Computing Machinery.

[12] Shirley Gaw, Edward W. Felten, and Patricia Fernandez-Kelly. Secrecy, flagging, and paranoia: adoption criteria in encrypted email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Conference on Human Factors in Computing Systems, pages 591–600, 2006.

[13] Daniel Gillmor. Guidance on end-to-end e-mail security. `https://datatracker.ietf.org/doc/draft-ietf-lamps-e2e-mail-guidance/`, 2022. Online; accessed 13 February 2022.

[14] Joscha Lausch, Oliver Wiese, and Volker Roth. What is a secure email? In *European Workshop on Usable Security (EuroUSEC)*, 2017.

[15] Ada (Adam) Lerner, Eric Zeng, and Franziska Roesner. Confidante: Usable Encrypted Email. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE European Symposium on Security and Privacy, pages 385–400, 2017.

[16] Juan Ramón Ponce Mauriés, Kat Krol, Simon Parkin, Ruba Abu-Salma, and M. Angela Sasse. Dead on Arrival: Recovering from Fatal Flaws in Email Encryption Tools. In *The LASER Workshop: Learning from Authoritative Security Experiment Results (LASER 2017)*, The LASER Workshop, pages 49—57. USENIX Association, 2015.

[17] Susan E. McGregor, Elizabeth Anne Watkins, Mahdi Nasrullah Al-Ameen, Kelly Caine, and Franziska Roesner. When the Weakest Link is Strong: Secure Collaboration in the Case of the Panama Papers. In *Proceedings of the 26th USENIX Security Symposium*, USENIX Security Symposium, pages 505—522. USENIX Association, 2017.

[18] Azadeh Mokhberi and Konstantin Beznosov. Sok: Human, organizational, and technological dimensions of developers' challenges in engineering secure software. In *European Symposium on Usable Security 2021*, page 59–75. Association for Computing Machinery, 2021.

[19] Jens Müller, Marcus Brinkmann, Damian Poddebniak, Hanno Böck, Sebastian Schinzel, Juraj Somorovsky, and Jörg Schwenk. "Johnny, you are fired!" – Spoofing OpenPGP and S/MIME Signatures in Emails. In *28th USENIX Security Symposium, USENIX Security 2019.*, 2019.

[20] Jens Müller, Marcus Brinkmann, Damian Poddebniak, Sebastian Schinzel, and Jörg Schwenk. Re: What's Up Johnny? – Covert Content Attacks on Email End-to-End Encryption. https://arxiv.org/ftp/arxiv/papers/1904/1904.07550.pdf, 2019.

[21] Don Norman. Affordances and design. *Unpublished article, available online at: http://www. jnd. org/dn. mss/affordances-and-design. html*, 2004.

[22] Adrian Reuter, Ahmed Abdelmaksoud, Karima Boudaoud, and Marco Winckler. Usability of End-to-End Encryption in E-Mail Communication. *Frontiers in Big Data*, 4:568284, 2021.

[23] Adrian Reuter, Karima Boudaoud, Marco Winckler, Ahmed Abdelmaksoud, and Wadie Lemrazzeq. Secure email - a usability study. In Matthew Bernhard, Andrea Bracciali, L. Jean Camp, Shin'ichiro Matsuo, Alana Maurushat, Peter B. Rønne, and Massimiliano Sala, editors, *Financial Cryptography and Data Security*, pages 36–46, Cham, 2020. Springer International Publishing.

[24] Volker Roth, Tobias Straub, and Kai Richter. Security and usability engineering with particular attention to electronic mail. *International Journal of Human-Computer Studies*, 63(1-2):51–73, 2005.

[25] Scott Ruoti, Jeff Andersen, Scott Heidbrink, Mark O'Neill, Elham Vaziripour, Justin Wu, Daniel Zappala, and Kent Seamons. "We're on the Same Page": A Usability Study of Secure Email Using Pairs of Novice Users. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, Conference on Human Factors in Computing Systems, pages 4298–4308. ACM, 2016.

[26] Scott Ruoti, Jeff Andersen, Travis Hendershot, Daniel Zappala, and Kent Seamons. Private Webmail 2.0: Simple and Easy-to-Use Secure Email. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, Annual Symposium on User Interface Software and Technology, pages 461–472. ACM, 2016.

[27] Scott Ruoti, Jeff Andersen, Tyler Monson, Daniel Zappala, and Kent Seamons. A Comparative Usability Study of Key Management in Secure Email. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 375–394, Baltimore, MD, August 2018. USENIX Association.

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

[29] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy van der Horst, and Kent Seamons. Confused Johnny: when automatic encryption leads to confusion and mistakes. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, Proceedings of the Ninth Symposium on Usable Privacy and Security - SOUPS '13, pages 69–88, Ottawa, 2013. USENIX Association.

[30] Yukiko Sawaya, Mahmood Sharif, Nicolas Christin, Ayumu Kubota, Akihiro Nakarai, and Akira Yamada. Self-Confidence Trumps Knowledge: A Cross-Cultural Study of Security Behavior. International Conference on Human Factors in Computing Systems, pages 2202 – 2214, 2017.

[31] Stuart E Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. The Emperor's New Security Indicators. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, IEEE Symposium on Security and Privacy, pages 51 – 65, 2007.

[32] 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 *Poster session of the Second Symposium On Usable Privacy and Security*, 2006.

[33] Ryan Sipes. Openpgp in thunderbird 78. https://blog.thunderbird.net/2020/09/openpgp-in-thunderbird-78/, 09 2020. Online; accessed 18 February 2022.

[34] Christian Stransky, Dominik Wermke, Johanna Schrader, Nicolas Huaman, Yasemin Acar, Anna Lena Fehlhaber, Miranda Wei, Blase Ur, and Sascha Fahl. On the Limited Impact of Visualizing Encryption: Perceptions of E2E Messaging Security. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, Symposium on Usable Privacy and Security, pages 437—454. USENIX Association, 2021.

[35] 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 *Proc. 43rd IEEE Symposium on Security and Privacy (SP'22)*, Symposium on Security and Privacy. IEEE, 2022.

[36] David R. Thomas. A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*, 27(2):237–246, 2006.

[37] Jan Tolsdorf and Luigi Lo Iacono. Vision: Shred If Insecure – Persuasive Message Design as a Lesson and Alternative to Previous Approaches to Usable Secure Email Interfaces. In *Proceedings of the 5th European Workshop on Usable Security (EuroUSEC 2020)*, European Workshop on Usable Security, pages 172–177, 2020.

[38] 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.

[39] Christopher D Wickens, Sallie E Gordon, Yili Liu, and J Lee. *An introduction to human factors engineering*, volume 2. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[40] P. Wouters. DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP. RFC 7929 (Experimental), August 2016.

A Procedure

Original instructions as presented to all participants of the study. Text enclosed by "<" and ">" denotes comments that were not contained in the original material.

A.1 Informed Consent and Explanation

A.1.1 Informed Consent (Page 1)

Dear participant, thank you for taking part in this study! Your participation will take only about 20-30 minutes of your time, but will help us tremendously in understanding the usage of email encryption and signatures.

We are researchers from the University of Applied Science Münster and Karlsruhe Institute of Technology. Our goal in this study is to investigate the usage and perception of e-mail encryption and signatures. This study comprises the following steps:

1. Informed consent (this page)
2. Instructions and introductory questions
3. Assessment of eight (8) emails regarding their signatures' legitimacy
4. Closing questionnaire

We will ask you to vocalize your thoughts while looking at the emails. In order to get a better understanding of your perceptions of the emails, both, your voice and your interaction on the screen will be recorded. The basis for the collection and analysis of the data is our data privacy policy [DE/EN].

Your participation is voluntary. If you wish to withdraw your participation before, during or after the completion of the survey, you can do so. If withdrawing, all data recorded up until this point will be discarded and deleted. For withdrawal from the survey once you have completed it, you will need to provide the participant code that you see below. Please write it down on the piece of paper provided to you.

Participant code: <randomly generated>

Please check the box below to indicate your agreement to participate in the study.

□ I am at least 18 years old and agree to participate in the study under the conditions as stated above.

A.1.2 Explanation (Page 2)

Dear participant, thank you for agreeing to participate in this study! To complete this study, you have to progress through four parts which are described in the following.

Note: Throughout the study, you will be asked to close program windows in mutiple instances. This is an essential part of the study and represents having completed the respective task. Therefore, be sure to close the program windows only, once you have completed the respective task. In particular, when answering questionnaires only close the windows, once the questionnaire instructs you to do so or your answers might be lost.

Part 1: Informed consent and explanation This part is the one you currently see. It comprised your agreement to participate in this study (previous page) and explains the tasks comprised in this study (this page).

Part 2: Introductory questionnaire In this part, the task is to fill a questionnaire with questions about your usage of email encryption and signatures as well as questions regarding your IT background.

Part 3: Assessment of eight emails regarding their signature's legitimacy This part comprises the tasks of rating the legitimacy of the signatures of eight emails. Each rating task encompasses the following two steps:

1. Each email will be opened in a dedicated Thunderbird window, allowing you to check the legitimacy of the signature. Once you have decided whether or not the signature is legitimate, you will have to close the Thunderbird window.

2. Once the Thunderbird window is closed, a short questionnaire in which you have to indicate whether the signature is legitimate or not will be opened automatically. Having completed the questionnaire for the respective email, you will be instructed to close the respective browser window in order to advance to the next email. However, after completing the questionnaire of the 8th (last) email, you will automatically continue with part 4 of the study, without the need to close the window.

These two steps are repeated for each of the eight emails. Note that each email should be rated in isolation, i.e. the emails do not reference each other and should be rated on its own.

In order to get a better understanding of your perceptions of the emails, we will ask you to vocalize your thoughts and to explain your actions while looking at the emails. Talk out loud constantly, telling everything you are thinking beginning from the moment you see the first email up to the point when you have made your decision regarding the legitimacy of the last of the emails. Please try to not plan out what you are going to say and do not try to explain your thoughts. Just act as if you were alone in the room and talking to yourself. Your voice and your interaction on the screen will be recorded. The recording of the screen and your voice will start automatically at the beginning of this part.

Part 4: Closing questionnaire

This part comprises a final questionnaire.

This procedure with all four parts is illustrated in the following: <Inline image of procedure, see Figure 3.>

A.2 Introductory Questionnaire

1. Have you participated in a study on email encryption and signatures at either the Chaos Communication Camp 2019 or the Chaos Communication Congress 2019?

   ○ Yes ○ No

2. Please describe how you usually check if an email you received is legitimate or was sent by a scammer.
   Please do not include any sensitive information about other people in your answer.

   <Multiline free text form>

3. Are there any additional checks you would perform on all incoming emails if you knew you were at risk of being specifically targeted?
   Please do not include any sensitive information about other people in your answer.

   <Multiline free text form>

4. Please indicate to what extent the following statements apply to you. <Likert items from (1) "does absolutely not apply to me" to (5) "absolutely applies to me">

   - I use email encryption and signatures regularly
   - I am confident in my ability to use email encryption and signatures (PGP, S/MIME, etc.)
   - I feel confident in being able to explain how to operate the email encryption and signature scheme I use (PGP, S/MIME, etc.) to others
   - When encountering problems handling encrypted or signed emails I usually know what the problem is
   - I believe I would recognize emails with invalid signatures

5. I handle PGP encrypted and signed emails in Thunderbird on average about <dropdown>

   ○ once
   ○ twice
   ○ three times
   ○ four times
   ○ five times
   ○ more than five times
   ○ I don't handle PGP encrypted / signed emails

   per <dropdown>

   ○ hour
   ○ day
   ○ week
   ○ month
   ○ year
   ○ I don't handle PGP encrypted / signed emails

<new page>

For this part of the study, please assume the following:

- You are Alice, a software developer at SecurePay24.
- Your email address is alice@securepay24.de.
- Your company has authorised an external security audit of the software you are currently working on.
- The security audit is performed by Code Audit Inc.
- You know Bob, the contact person at Code Audit Inc., from a conference call meeting.
- Bob's email address is bob@code-audit.org.
- You have exchanged keys with Bob, i.e. you have his public key in your keychain.

Please click “Next” to continue with the study.

<new page>

In the next step of this study you will see an email opened in Thunderbird. Inspect it to determine whether its signature is legitimate or not. After having inspected the email, please close the Thunderbird window to proceed with the study. Remember: all emails are independent of each other and should be rated in isolation.

In this part of the study, please vocalize your thoughts and explain your actions while looking at the emails. Talk out loud constantly telling everything you are thinking, beginning from the moment you see the first email up to the point when you have made your decision regarding the legitimacy of the last of the emails. Please try to not plan out what you are going to say and do not try to explain your thoughts. Just act as if you were alone in the room and talking to yourself.

Please close this browser window now to proceed to the email. This will also start the recording of your voice and the interaction on screen.

A.3 Assessment of Eight (8) Emails Regarding Their Signatures’ Legitimacy

1. Is the signature of the previously inspected email legitimate?
   ○ Yes, the signature is legitimate
   ○ No, the signature is not legitimate
2. Is there anything else you want to tell us with respect to the email you saw?
   Note here e.g. if you have closed the windows prematurely (i.e. before finishing inspecting the email).
   <Multiline free text form.>

A.4 Closing Questionnaire

A.4.1 SA6

3. On a scale of “Strongly Disagree” to “Strongly Agree”, rate your level of agreement with the following statements. <Likert items from (1) “Strongly Disagree” to (5) “Strongly Agree”>

- I am extremely knowledgeable about all the steps needed to keep my online data and accounts safe.
- I am extremely motivated to take all the steps needed to keep my online data and accounts safe.
- I often am interested in articles about security threats.
- I seek out opportunities to learn about security measures that are relevant to me.
- Generally, I diligently follow a routine about security practices.
- I always pay attention to experts’ advice about the steps I need to take to keep my online data and accounts safe.

A.4.2 RSebis

4. To what extent do following statements apply to you? <Likert items from (1) “Never” to (5) “Always”>

- I use a PIN or passcode to unlock my mobile phone.
- I include special characters in my password even if it’s not required.
- When browsing websites, I mouseover links to see where they go, before clicking them.
- If I discover a security problem, I fix or report it rather than assuming somebody else will.
- When I’m prompted about a software update, I install it right away.
- I use different passwords for different accounts that I have.
- I set my computer screen to automatically lock if I don’t use it for a prolonged period of time.
- I try to make sure that the programs I use are up-to-date.
- When I create a new online account, I try to use a password that goes beyond the site’s minimum requirements.
- I manually lock my computer screen when I step away from it.
- I change my passwords even if it is not needed.
- I use a password/passcode to unlock my laptop or tablet.
- I know what website I’m visiting by looking at the URL bar, rather than by the website’s look and feel.
- I verify that information will be sent securely (e.g., SSL, “https://”, a lock icon) before I submit it to websites.
- I verify that my anti-virus software has been regularly updating itself.
- When someone sends me a link, I open it only after verifying where it goes.

A.4.3 Debriefing

Thank you for participating in this study!

The study is now finished, please contact the experimenter to receive a debriefing and ask any potential questions you might have.

B Code Book

USED CHECKS: *Alternative Message Views, Banner Indicator, Banner Position, Banner Signer, Compare Signer to Sender, Crypto Algorithms, Fingerprint, GUI Behaves Unexpectedly, Header Information, Key Creation Date, Key is in Keyring, Key Property Trust Statement, Key Validity, Keyring, Letter Icon Status, Mail Content, Mail is Classified as Junk, Mail is Encrypted, Mail Source, Security Info Statement, Sender's Key, Signature Date, Signer's Key is Signed with Own Key*;
PROPOSED CHECKS: *Compare Fingerprint to Known One, Fingerprint, Key Revocation, Keyring, Mail Source, Recheck with GPG on Command Line, Out of Band Verification, Signature Date*;
USEFULNESS: *Indeterminate, Neutral, Right Direction, Wrong Direction*;
DECISION: *False Illegitimate, False Legitimate, True Illegitimate, True Legitimate*;
PERCEPTION: *Email is encrypted, I might fall for this in real life, I might fall for this in a study, No distinction of Thunderbird and Enigmail, Not sure why Thunderbird trusts signature, Uncertainty leads to mistrust*;
PROBLEM: *Bad GUI design, Does not know what to do, GUI target too small, Misleading GUI, Unable to locate desired option, Unhelpful information*;
VALIDITY: *Check possible due to study setting, Check potentially failed due to study setting, Checks intensively*

C Email Test Cases

Any email consist of a set of headers and a payload. The test emails had the following headers: `Received`, `To`, `From`, `Subject`, `Message-ID`, `Date`, `User-Agent`, `MIME-Version`, and `Content-Type`. From these headers, only `To`, `From`, `Subject`, and `Date`, are used by Thunderbird in the graphical UI. Other headers are only available by additional configuration or when viewing the raw email source.

In the eight provided test cases, only the `Content-Type` could differ, in order to use different body payloads. In other words, only the email body is relevant to discern legitimate from illegitimate emails in our study. All (irrelevant) headers were set to "sane" defaults, such that no participant focused (nor were misguided) by missing or incorrect headers.

The following email serves as a template for all email test cases:

```
Received: ...                             // irrelevant
To: alice@securepay24.de
From: bob@code-audit.org
Subject: Upcoming Security Audit
Message-ID: ...                           // irrelevant
Date: Wed, 30 Jan 2020 16:45:59 +0100
User-Agent: ...                           // irrelevant
MIME-Version: ...                         // irrelevant
Content-Type: {}

{}
```

Although the actual payload differed, the Thunderbird UI always showed the following text in its main window:

```
Dear Alice,

as discussed in our last meeting, the security audit is
about to start.
Could you please provide me with access to the source
code for your project on github?

My user id is: bob-49

Greetings
Bob
```

C.1 Email Test Case: Legitimate

A legitimately signed email. The root `Content-Type` is `multipart/signed` and the payload was correctly signed with the key of `bob@code-audit.org`.

C.2 Email Test Case: Broken Signature

This test recreates an email with a broken signature. It only differs from the legitimate email by a non-functional change to the MIME boundary. In effect, Thunderbird is not able to correctly verify the signature anymore, but the signer is still `bob@code-audit.org`.

C.3 Email Test Case: UI Redressing

An email without a cryptographic signature at all. HTML and CSS were used to mimic Enigmail's "green bar." The bar is not clickable, but otherwise a pixel-perfect copy of the original Enigmail bar. It resizes when Thunderbird is resized. However, the position of the bar differs from Enigmail. In Enigmail versions below 2.0.8 the bar was below Thunderbird's header area, and this placement is used in this test mail. However, Enigmail has since changed the position of the green bar to be above Thunderbird's header area. The source code was obfuscated to hide the HTML and CSS elements (via base64), and the MIME boundary was set to `--PGP SIGNED MESSAGE---` to pretend that OpenPGP was used in some form. Due to the required images, the source code was substantially longer compared to the legitimate email.

Reasoning We obfuscated the source code to redirect our participants to focus on the Enigmail elements, since prior participants at 36c3 classified the email as illegitimate as soon as they saw the HTML source code.

C.4 Email Test Case: Sender is not Signer

This email is equal to the legitimate email except for the OpenPGP signature. Here, the email was signed with the key of `celine@example.org`, instead of `bob@code-audit.org`. However, the `From` header still states that the email is from `bob@code-audit.org`. This test is motivated by the fact that OpenPGP signatures are traditionally not bound to the

From header. In S/MIME, this check is conducted by the email client, and we anticipated this discrepancy as a potential source of confusion.

C.5 Email Test Case: Sender is not Signer 2

This email is equal to the signer-vs-sender email but the signer uses a typo-domain bob@code-audil.org, instead of bob@code-audit.org (note the ‘l’ instead of ‘t’).

This test is motivated by the fact, that newer technologies such as Autocrypt, WKD, etc. may automatically import OpenPGP keys into the local key ring. Here, we test the phase *after* automatic inclusion. In other words, the key of bob@code-audil.org is trusted. To prevent that a participant spots this key in a prior test, this key is only contained in the key ring during the period of this test.

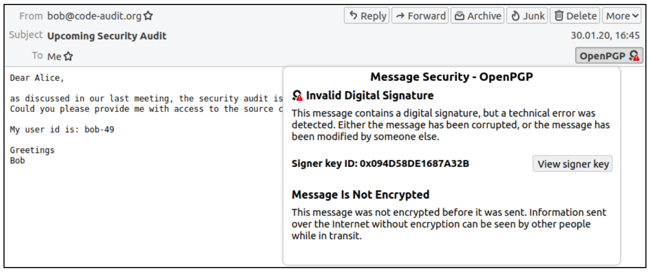
D GnuPG Key Ring

The GnuPG keyring contained the following trusted keys:

- Alice <alice@securepay24.de>
- Bob <bob@code-audit.org>
- Celine <celine@example.org>
- David <david@example.org>
- Ezra <ezra@code-audit.org>
- Farah <farah@example.org>
- Garrett <garrett@code-audit.org>
- Hoy <hoy@example.org>
- Iva <iva@example.org>
- Joon <joon@code-audit.org>
- Kemina <kemina@example.org>

Additionally, during runtime of the Sender is not Signer2 case, Bob <bob@code-audil.org> (note the “l”) was added as a trusted key to the key ring.

E Screenshot of the new Thunderbird interface for the Broken-Signature case



From bob@code-audit.org
Subject Upcoming Security Audit
To Me
30.01.20, 16:45

```
Dear Alice,

as discussed in our last meeting, the security audit is
Could you please provide me with access to the source c

My user id is: bob-49

Greetings
Bob
```

Message Security - OpenPGP

Invalid Digital Signature

This message contains a digital signature, but a technical error was detected. Either the message has been corrupted, or the message has been modified by someone else.

Signer key ID: 0x094D58DE1687A32B

Message Is Not Encrypted

This message was not encrypted before it was sent. Information sent over the Internet without encryption can be seen by other people while in transit.

Figure 9: *Broken-Signature* email in Thunderbird 91.5.0.

F Additional Transition Graphs

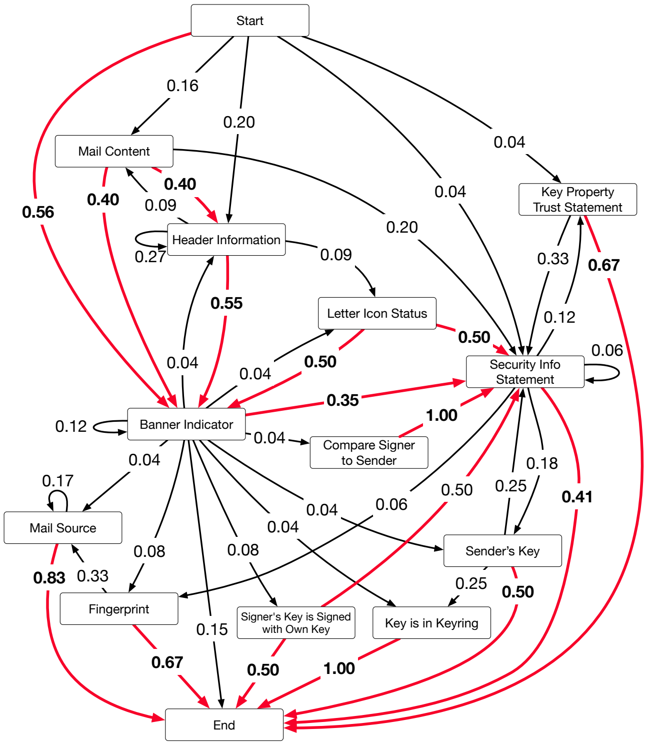


Figure 10: Overview of our participants’ transition probabilities from one check to another for the *Broken-Signature* email. The most likely transition after performing each of the checks is marked in red. Due to rounding the probabilities for each node might not add up to 100%.

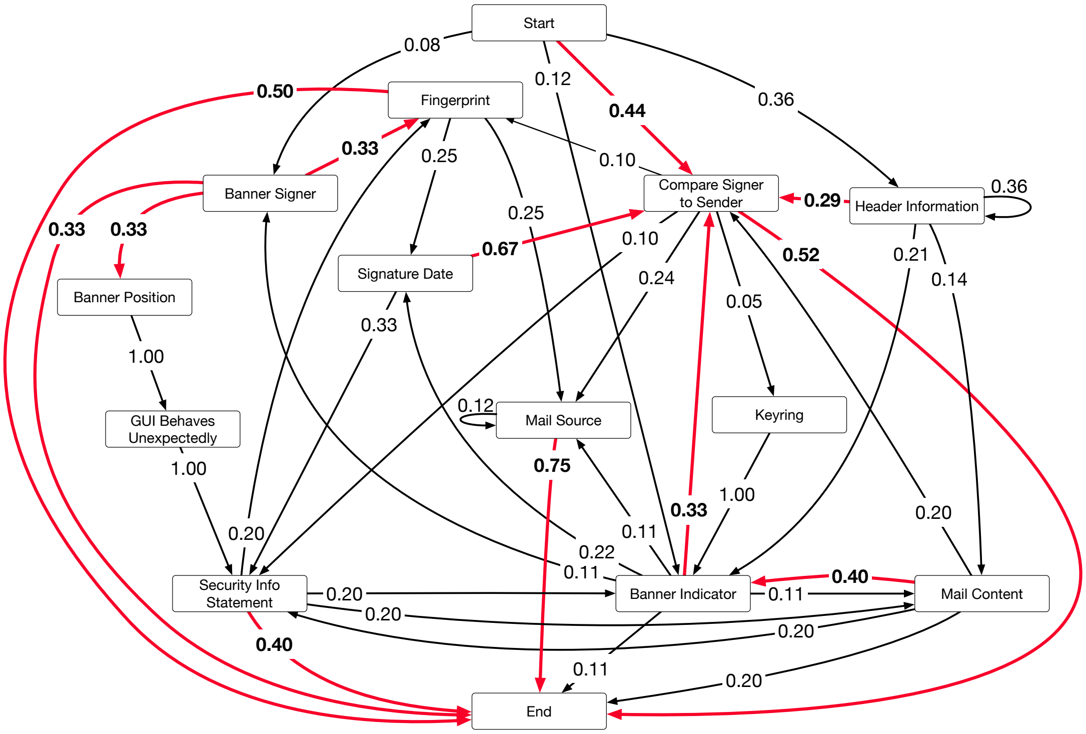


Figure 11: Overview of our participants' transition probabilities from one check to another for the *Conflicting-Signer* email. The most likely transition after performing each of the checks is marked in red. Due to rounding the probabilities for each node might not add up to 100%.

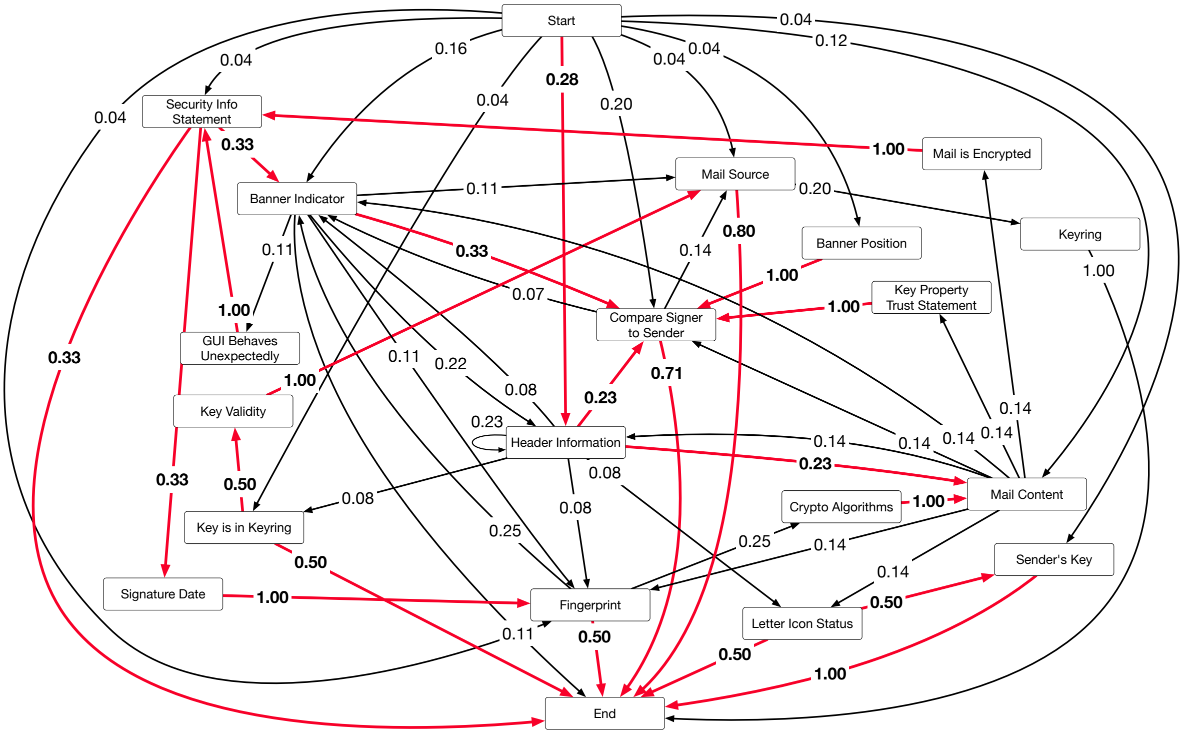


Figure 12: Overview of our participants' transition probabilities from one check to another for the *Conflicting-Signer-Subtle* email. The most likely transition after performing each of the checks is marked in red. Due to rounding the probabilities for each node might not add up to 100%.