Beware the Downgrading of Secure Electronic Mail

Oliver Wiese
Freie Universität Berlin
Berlin, Germany
oliver.wiese@fu-berlin.de

Joscha Lausch
Freie Universität Berlin
Berlin, Germany
joscha.lausch@fu-berlin.de

Jakob Bode
Freie Universität Berlin
Berlin, Germany
jakob.bode@fu-berlin.de

Volker Roth
Freie Universität Berlin
Berlin, Germany
volker.roth@fu-berlin.de

ABSTRACT

Researchers have taken considerable interest in the usability challenges of end-to-end encryption for electronic mail whereas users seem to put little value into the confidentiality of their mail. On the other hand, users should see value in protection against phishing. Designing mail apps so that they help users resist phishing attacks has received less attention, though. A well-known and widely implemented mechanism can be brought to bear on this problem – digital signatures. We investigated contemporary mail apps and found that they make limited use of digital signatures to help users detect phishing mail. Based on the concept of digital signatures we developed and studied an opinionated user interface design that steers users towards safe behaviors when confronted with phishing mail. In a laboratory study with 18 participants we found that the control group was phishable whereas the experimental group remained safe. The laboratory setup has known limitations but turned out to be useful to better understand the challenges in studying e-mail security experimentally. In this paper, we report on our study and lessons learned.

KEYWORDS

Secure electronic mail, phishing, end-to-end-encryption

ACM Reference Format:
Oliver Wiese, Joscha Lausch, Jakob Bode, and Volker Roth. 2018. Beware the Downgrading of Secure Electronic Mail. In *Proceedings of Workshop on Socio-Technical Aspects in Security and Trust (STAST2018).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In a post Snowden era, confidentiality is an essential property of secure communication. App developers are responding to this need and feature end-to-end encryption capabilities in their apps more prominently. Some popular messenger apps build their reputation on end-to-end encryption, for example, Signal, Telegram and Threema. In contrast, end-to-end encryption for electronic mail is widely implemented but is rarely used even though mail is still broadly used.

Beside person-to-person communication, typical use cases of mail are signing up for services, managing login credentials, receiving receipts and coupons, communicating with doctors and other professionals, and business communication [6]. This makes mail an attractive surveillance target of spies and a phishing target of cybercriminals. In the presence of phishing attacks security properties other than confidentiality are becoming more important, that is, integrity, authenticity and non-repudiation. Integrity[1] can be achieved by means of message authentication codes or digital signatures. Authenticity[2] and Non-repudiation[3] can be achieved by means of digital signatures.

Phishing is an *active* attack whereas the confidentiality of mail may be breached by *passive* attacks if encryption features are absent or badly implemented. In 2005, Garfinkel and Miller studied how Outlook users dealt with active attacks. Their design significantly improved the security of users against phishing but was no panacea. We continue this line of investigation in our paper. Specifically, we study how individuals behave when they can reasonably expect to receive signed mail but receive mail without signatures. This models phishing strategies that seek to circumvent the fact that phishers cannot forge signatures in keys they do not possess. This strategy is independent from any particular cryptographic standard or trust model such as *Public Key Infrastructure* (PKI) in S/MIME, *Web of Trust* in PGP, or *Trust On First Use* (TOFU).

In what follows, we detail our threat model, related work and current solutions. We rationalize the design decisions we took, we describe our mail app design and we describe, report and discuss the results of an empirical study we conducted with 18 participants. The study investigated how participants handled our app and how they responded to phishing attempts in a job application scenario. We conclude with remarks on our motivation and what we consider worthwhile future work.

2 THREAT MODEL

Current research on the subject of secure mail focuses on protection against passive eavesdropping, for example, by governmental agencies or mail providers. In addition to this threat, mail users are

STAST2018, December 2018, San Juan, Puerto Rico, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] The content is received as sent.
[2] The receiver can be convinced that the mail was sent by someone holding a specific private key.
[3] The receiver can convince a third party that a given mail was sent by the holder of a specific private key.

exposed to phishing attacks. A phishing attack is an active attack, the adversary impersonates a particular person or organization. We assume that the user communicated with the impersonated person or organization before. We exclude an attacker who has access to the mailbox of the impersonated person/organization or user. In our threat model the adversary can forge the mail address of mail without having access to the impersonated mail account.

We assume that the impersonated person or organization signs outgoing mail and the adversary does not sign the mail. The adversary cannot forge a digital signature or message authentication code (MAC).

3 STATE OF THE ART AND RELATED WORK

3.1 End-to-End Authentication

The headers and bodies of mail are not protected cryptographically by default [19, 23]. Consequently the sender address is not a reliable indicator of the sender. Digital signatures or message authentication codes support a cryptographically sound verification of the sender, when applied properly. Common mail encryption standards such as S/MIME [29] or PGP [13] support digital signatures. The pertinent keys can be associated with a mail address. For example, Apple's Mail.app only indicates that a mail is signed if the mail address in the certificate of a signer corresponds to the sender field. This requires a certificate to work. The certificate must be linked to a local trust anchor. Alternatively, local trust settings can be configured and self signed certificates can be imported to that effect. This, however, erects insurmountable barriers to laymen users.

Besides Apple's Mail.app, iOS Mail, Thunderbird and Outlook support S/MIME by default and PGP with extensions. All applications distinguish between encrypted and signed mail and use separate security indicators for encryption and signatures. Neither of these mail user agents flag unsigned mail or warn users if they receive unsigned mail from a contact who sent signed mail previously. Users have to attend to these potential risks themselves.

3.2 Authentication of Relayed Mail

Besides client-side countermeasures, the Internet Engineering Task Force (IETF) specified a variety of complementary mechanisms that mail service providers can deploy to authenticate the senders of mail.

(1) The *Sender Policy Framework* (SPF) standard allows domain administrators to authorize hosts that are allowed to send mail in their name by way of DNS records. Mail transfer agents can request and use this information to verify whether received mail really originated from an authorized host in the sender's domain [18].
(2) The *DomainKeys Identified Mail* (DKIM) standard allows domain administrators to sign outgoing mail. The signing keys are published in DNS records as well [1]. Mail transfer agents can request and use this information to verify the received mail really originated in the sender's domain.
(3) The *Domain-Based Message Authentication, Reporting and Conformance* (DMARC) standard allows domain administrators to publish SPF and DKIM policies, which specify how mail that cannot be authenticated should be handled [22].

These standards are meant to assure that a mail address means what it says. In principle, users should be able to use that information to discern phishing mail from genuine mail for domains they deal with on a regular basis. The situation is not quite as simple in practice for at least two reasons.

First, some mail user agents implement "smart addresses", for example, Apple Mail. This means that only the display part of the mail address is shown to the user. Phishers may leverage this by sending mail from any legitimate domain (with respect to SPF/DKIM) and by forging merely the display name portion of the sender address. The crucial mailbox address is hidden from them.

Second, the deployment of SPF/DKIM/DMARC is less than complete. In 2015, Durumeric et al. [11] reported that the top mail providers enrolled SPF/DKIM/DMARC but only 47% of all Alexa Top Million mail servers enrolled SPF and only 1.1% specified a DMARC policy. Of all policies, 58% were soft-failures where the receiving host is asked to accept a mail but mark it as suspicious, for example, as spam, and another 20.3% had a "no policy" policy. Foster et al. [14] reported that most of the 22 most popular mail providers, for example, hotmail.com, gmail.com, yahoo.com, aol.com, gmx.de, and mail.ru, made a DNS lookup for SPF but only 10 acted upon it. Foster et al. concluded that senders from Yahoo cannot be impersonated when sending mail to Hotmail, GMail and Yahoo accounts "and a handful of other providers." The Federal Trade Commission (FTC) reported in 2017 that 86% of the top businesses with authorized mail hosts are using SPF but only 10% of them have soft-or-strict DMARC policies. [4] Small businesses hardly ever employ DMARC according to a 2018 report of the FTC. This was based on a sample of 11 web hosting providers for smaller businesses. Only one of the providers implemented SPF by default and 10 of them neither integrated a SPF setup during enrollment nor introduced the technology.

In summary, SPF, DKIM and DMARC help against phishing attacks but are not widely adopted or enforced. In what follows, we focus on end-to-end countermeasures rather than provider-based protection.

3.3 Research on Mail Encryption

Garfinkel and Miller proposed and studied a key continuity management design [15] using Outlook as an example. Their design distinguished four states of incoming mail by means of colored borders. Yellow indicated that a key and mail address were encountered for the first time. Green indicated that a mail was signed and the combination of key and mail address had been encountered before. Red indicated that the mail was signed but the key differed from what had been associated with that mail address before. Finally, gray indicated that a mail was unsigned but the sender had sent signed mail before. Garfinkel and Miller found that their design significantly improved the security of users against phishing but was no panacea. In particular, users replied to spoofed mail in the gray state in order to verify whether the sender was able to reply. Users took replies as confirmation of authenticity. Garfinkel and Miller also reported that users tended to be confused about the

[4] See:
https://www.ftc.gov/system/files/attachments/press-releases/online-businesses-could-do-more-protect-their-reputations-prevent-consumers-phishing-schemes/email_authentication_staff_perspective_0.pdf

exact guarantees provided by signing and encryption. However, while their design provided users with information they needed to decide what to do next, their design did not encourage users to take specific actions to resolve suspicions they might have about potentially harmful mail.

In recent studies on private webmail (PWM) based on identity-based encryption, Ruoti et al. investigated encryption of mail but not the authenticity of mail [24]. PWM informs users that only the receiver can read a message (confidentiality). Their threat model focused on protecting the mail body against eavesdropping [26]. Neither integrity nor authenticity played a significant role, phishing and credential theft were not considered in their threat model. In [24], Ruoti et al. contrasted manual and automatic encryption. Study participants had to encrypt and decrypt messages but not verify them. In [25], they compared integrated and depot-based mail approaches and a mix of both. In their study, participants were asked to send a PIN and a SSN via encrypted mail to another participant. In subsequent interviews, encryption was often mentioned but authenticity and integrity were not.

A study of Atwater et al. [4] focused on encryption as well. Their software was based on PGP and a verified key server as a means of finding other public keys. In their application, users were able to disable encryption but it is unclear whether messages were signed or not.

Participants of several lab studies reported that they could not think of situations in which to encrypt [4, 25]. Gaw et al. and Lerner et al. [16, 21] studied mail encryption in the context of specific subgroups of users. Gaw et al. [16] interviewed employees of a non-violent, direct action organization and reported that encryption is used for classified information. Lerner et al. [21] studied mail encryption with a focus on lawyers and journalists. Six of eight lawyers pointed out that attorney-client privilege is an asset. Four mentioned account hacking as a threat, three mentioned mail spoofing, plaintext held by mail providers and government surveillance. Six of seven journalists named protecting sources as an asset but as Lerner et al. discussed this, metadata of mail can already reveal individuals who communicate with a journalist.

In summary, current research on secure mail focuses on protection against eavesdroppers. The authenticity of mail plays a lesser role if any. The problems that arise from omitting signatures when they may be expected was touched only by Garfinkel et al. [15]. This problem exists independently of common concerns such as the choice of key management, specifically the choice between a *Web of Trust,* PKI, TOFU or perhaps the approach taken by *Keybase.*[5]

3.4 Phishing Detection in Browsers

Previous research shows that phishing detection is difficult for end-users without software support. In lab studies, primed participants failed to detect phishing websites [3, 9]. Dhamija et al. [9] studied how well users distinguish between a real and a fake website in a laboratory setting. One phishing website fooled more than 90% of the participants. About one decade later, participants were still vulnerable to phishing in a similar study by Alsharnouby et al. [3]. The researchers evaluated anti-phishing toolbars and concluded that they do not protect users against high-quality phishing attacks. Some users even ignored the toolbars [30, 31]. In a lab study by Egelman et al. [12], active browser warnings and indicators protected users against phishing attacks better than passive warnings. Later research shows that users do not pay much attention to browser warnings and security indicators for SSL [2, 28].

Researchers investigated the reaction of end-users to phishing mails. In a study of Jagatic et al. [17] mail address spoofing increased the success rate of a phishing attack from 16% (unknown sender) to 72% (friend's address). Downs et al. [10] studied different mail response decision strategies and concluded that none of them protect against well-constructed phishing attacks. In a study by Blythe et al. [7] well-presented mail and logos increase the success rate of an attacker. In contrast to digital signatures, text and logos are forgeable.

4 DESIGN RATIONALE

The primary purpose of digital signatures in the context of mail is to provide assurance that a mail is from the holder of a particular private key. Its application to the non-repudiation of a sent mail seems largely irrelevant in practice. We are not aware of a legal dispute that has been settled over the presentation of a signed mail. Mail encryption, on the other hand, is not generally suited to establish authenticity. Still, anecdotal evidence by Garfinkel et al. [15] and Lausch et al. [20] indicates that users are not necessarily aware of that. Keeping signatures and encryption separate in the user interface also opens up the opportunity that users choose to encrypt but not to sign outgoing mail. What they may not realize is that this potentially renders mail malleable in transit. Therefore, it is sensible to treat encryption and signatures as integral parts of secure mail. Consequently, we distinguish three states of received mail:

(1) **Secure:** received mail is encrypted and signed.
(2) **Insecure:** received mail is not encrypted or not signed.
(3) **Corrupt:** cryptographic processing yields an error.

Secure mail is indicated by an icon that symbolizes a closed envelope. Insecure mail is indicated by a postcard. Corrupt mail is indicated by an envelope that is torn open at the upper right corner. Lausch et al. [20] found that these metaphors were understood well by users. They strike a compromise between a padlock metaphor (often used in the context of browser security and encryption) and seals (often used for signatures). The torn envelope worked particularly well for indicating error conditions. Padlocks and seals do not offer such a "third state" that lends itself to easy interpretation. Perhaps most important is that the envelope metaphor is unencumbered whereas users may have developed preoccupation about the meanings of padlocks and seals. We expect that this makes it easier to endow the envelope metaphor with a fresh interpretation. Besides, the name electronic mail alone establishes an analogy to paper mail, which immediately suggests envelopes and postcards as symbols.

When sending mail, we always sign and encrypt together whenever a key of the receiver is known. If a user composes mail for multiple recipients then we use the envelope indicator if and only if a key is known for all recipients. Otherwise, we use the postcard indicator. The rationale is of course that one plaintext mail suffices to leak the contents of a mail irrespective of how many copies are

[5] see: https://keybase.io/

Figure 1: Shows our tutorial in which we introduce envelopes and letters. We focus on security implications and avoid technical details.

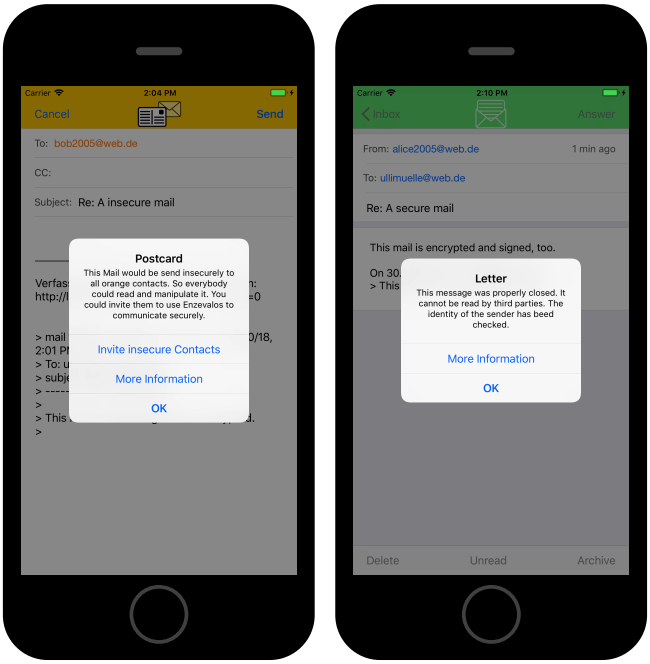


Figure 2: When tapping on icons, brief explanations of the associated security states pop-up.

sent encrypted. When receiving mail, we show a warning if we receive insecure mail from a sender who sent secure mail previously. The warning is designed to remind the user that he or she should not put too much trust into the mail. The warning additionally recommends that the user replies with a secure mail. This covers the case that phishers spoof the mail address of a sender with whom the receiver has an existing relationship, that is, the receiver has received a genuine signed mail from the genuine sender previously. What we report in this paper are the results of our studies of this case.

What we present next are details of the mail app we designed to support our study insofar as they are relevant for the interpretation of our study.

5 IMPLEMENTATION OF THE MAIL APP

As a basis for empirical studies on end-to-end protection mechanisms for mail we built a functional open source mail app prototype for iOS[6], a deliberate platform choice we made when we applied for our project funding. Since Apple's implementation of Mail on the iPhone does not support plugins or extensions we had to develop our own app. Upon first launch we used a brief tutorial to introduce users of our app to the intended interpretation of the envelope and postcard, as shown in Fig. 1. In addition to our iconic language we used the colors green and yellow to increase the visual distinctiveness of the two cases.

While not being feature-complete yet, the app does support accessing regular IMAP mailboxes over TLS. Encryption and signatures are implemented on top of PGP[7] but most key management functions are missing. PGP keys are stored in the phone's *Keychain* database and mail contacts can be added to the phone's *Contact Book*. Our app maintains a mapping between contacts and keys internally.

Mail correspondence is organized by contact and, within contacts, by key. Hence, a spoofed mail with a fresh key would appear in the contact's conversations but separate from the mail associated with the genuine key. Our app does not yet support updating keys in a forward-secure fashion. In principle, this can be implemented transparently for users. However, the specifics of that approach and whether it is preferable over a more explicit approach still justifies further research in our opinion.

When composing or viewing mail, a colored bar at the top of the screen indicates the mail's security state. The bar is either green with an envelope icon in the middle, or yellow with a postcard icon in the middle. At any time, users can pop up a brief explanation of the security state by tapping on the icon, as shown in Fig. 2. A button within the pop-up offers to take the user to a panel with more information.

Perhaps the most distinctive feature of our app is the warning that shows up at the top of an insecure mail that follows a secure mail of the same alleged sender, see left of Fig. 3. The warning cautions the user not to trust the insecure mail too much because previous mail form the same sender was secure. Additionally, it

[6]The source code of the app is publicly available on the university gitlab server: https://git.imp.fu-berlin.de/enzevalos

[7]We use the open source PGP library ObjectivePGP https://github.com/krzyzanowskim/ObjectivePGP

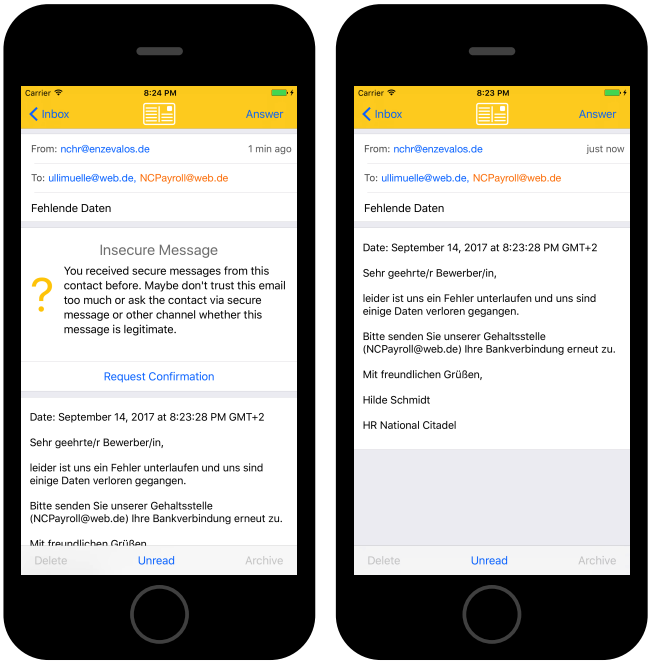


Figure 3: Participants in group A saw a warning (left) while group B only saw the message itself (right). The participants saw a German translation.

recommends that the user contacts the sender of the mail to inquire whether the mail is authentic. The warning includes a button that opens a reply mail automatically. Users only need to tweak the text to their preference and send it. The mail is sent signed and encrypted. The encryption key is the one associated with the mail address of the alleged sender. Of course, phishers will not be able to respond to the inquiry with a secure mail because they lack the necessary private key.

6 USER STUDY

6.1 Method & Study design

We conducted a lab study to test several of the concepts and security indicators of our iOS app. Specifically we wanted to test if a warning with a proposed action would affect the vulnerability of participants to a spear phishing attack. We tested this using an A/B-test. Participants were assigned to one of two groups in an alternating fashion. We also measured the usability of the onboarding and security features of the app using the *System Usability Scale* (SUS). Especially users' perception of the change of state between secure and insecure mail was of interest to us.

Some aspects of the study are modeled after a study conducted by Ruoti et al. [26]. The study participants took part in a simulated application procedure for the fictional bank *National Citadel*. All app usage instructions were given in the form of mail from the bank. Participants also had to perform a secondary task in which they had to securely transmit credit card information to a roommate named Bob.

6.2 Participants

We recruited our participants using flyers distributed on the campus of our university. The flyers solicited participation in a user study for a new iOS communication app. We did not disclose the security context of our study in order to avoid self-selection bias for people who are risk-aware or otherwise interested in security. We tried to select a diverse group in terms of age, gender and education of the applicants. However, we preferred iOS users because they are already accustomed to the usage patterns of the operating system and apps. We expected the results to be more consistent this way. Participants were compensated for their time with 15 €.

We recruited 20 participants for our study. We excluded the data of two participants from our analysis. The first participant encountered a technical problem that we fixed before we continued with our study. A mail that should have been decrypted was not. Hence, the task did not proceed as required. Another subject was an international student who claimed to understand the language in which we conducted the study. In a post-study interview it became clear that his language proficiency was not sufficient to follow the instructions.

Of our participants, 17 were students and one was a research assistant. Six participants had a background in psychology, three in educational sciences and three in economics. Biology (1), english philology (1), study of literature (1) and political science (1) were the other subjects. One participant did not answer that question. We asked the participants to rate their computer proficiency on a scale from 1 (beginner) to 5 (expert). Eleven of them rated themselves as a 4, six of them as a 3 and one of them as a 2. The mean age was 24.9 years. Five participants were male (27.8 %), fewer than the 40.8 % males that are enrolled in the university.

6.3 Procedure

The study took place between the 6th and the 15th of July, 2017, in a room of our university we reserved for our experiments. The entire procedure took between 30 minutes and an hour. Participants wore eye-tracking glasses which were calibrated anew for each participant. The app ran on an iPhone 6s Plus. The primary task was presented to the participants on a piece of paper. All further instructions came in the form of mail from the company. A written introduction to the secondary task was given to the participants after they completed the second task.

- **Task 1** The initial mail, which the participants read in the default iOS mail app, instructed them to set up our app according to company policy to secure the information they had to send the company in order to get a refund for their travel expenses. This task was designed to test the ability of the participants to setup the app without further assistance. Subsequently they had to send their first secure mail.
- **Task 2** The second mail informed the participants that they were accepted for the job. The mail instructed them to inform the accounting department of their bank account number. This mail was the first secure mail they received. This task was designed to test whether the participants were able to send a new secure mail to a new contact. We also wanted to see if they checked the security state of mail.

- **Task 3** The next mail came from Bob, a roommate of the fictional persona Ulli. He asked for credit card details in order to pay a utilities bill. This mail was insecure. This task was designed to show how participants invite a new user to our app. We did not evaluate in detail how the participants described the process to Bob. We continued as soon as it was possible for Bob to find the app in the App Store.
- **Task 4** Bob responds with a secure mail, stating that he installed the app. To this mail the participants were expected to respond with the credit card details.
- **Task 5** The responding mail from Bob was not secure. He asked for the verification number from the backside of the credit card. The previous correspondence was included in the mail as quotation. This task was designed to see how the participants would react to an insecure mail from a known contact of whom they had received a secure mail before. Participants from study group A were shown the warning message under study, see also Fig. 3.
- **Task 6** The last mail appeared to come from the HR department of National Citadel but was insecure. It asked the participants to resend their bank information because there had been a problem. This time the given mail address of the accounting department had a different domain. This task was designed to test the behavior of participants when confronted with a phishing mail. Participants from group A were shown the warning again.

6.4 Results

6.4.1 Usability. Previous studies [4, 24–26] on secure and usable MUAs used the System Usability Scale (SUS) for measuring usability. SUS is broadly used for initial usability evaluations and was proposed by John Brook in 1996 [8]. Users rate five positive and five negative statements about an application on a five point Likert scale. The scores are normalized on a scale from 0 to 100. Our mobile mail application achieved a SUS rating of 75.4 on average with a standard deviation of 11.7. According to the rating of Bangor et al. [5] our app is classified as *good* in terms of usability. The participants of the group with the warning message awarded the app a SUS of 74.7 (sd: 14.4) while the group without warnings awarded a SUS of 76.1 (sd: 9.2).

All participants were able to finish all tasks and were able to compose and read mail. After testing the app and finishing the tasks some participants mentioned that they missed a folder for *draft* and *sent* mail to refer to previous outgoing mail. During the study half of the participants used the Apple Mail.app to check for new mail.

We minimized the information about end-to-end encryption we provided to participants during the tutorial and the introduction. Likely for that reason all except one participant read all four introduction screens. Participants spent about 35 seconds on average on the introduction with a standard deviation of 15 seconds. During the experiment, 13 of 18 participants clicked on the icons (envelope and postcard) for more information and 3.8 times on average. The pop-up provided superficial information, guidelines about end-to-end encryption and an additional button for more information. Nine participants checked the pop-up for more information.

After reading Bob's initial mail, three participants tapped *answer* to compose a reply and then clicked on the postcard icon. In the resulting pop-up, they found the invitation button for insecure contacts and invited Bob. Seven participants investigated the contact information page of Bob and tapped on the invitation button. Three of ten participants modified the general invitation mail and referred in it to the previous mail of Bob. Additionally, one participant composed a separate invitation mail after having sent the general invitation mail. Eight participants composed an invitation mail of their own.

6.4.2 Security. In our experiment, users made the following security decisions when receiving insecure mail. Table 1 summarizes the results. When Bob forgot to secure (encrypt and sign) his response, all participants sent the data securely. Six of nine participants in the group with warnings followed the recommendation and asked Bob to verify the previous mail before sending the data. We observed no similar behavior in the group without warnings.

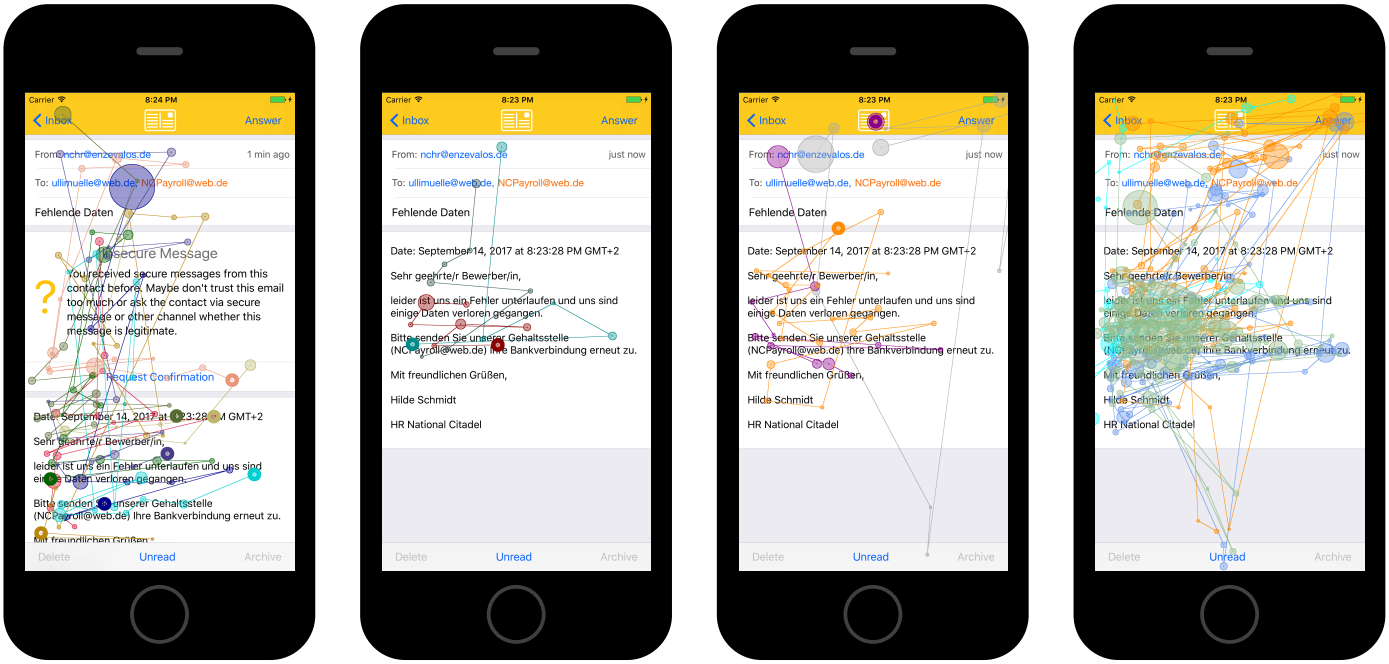
In the case of a phishing attack all participants in the group with warnings asked the sender to verify the previous message. All except one clicked on the button in the warning and sent the message without modification. One participant composed his own message and asked for a verification. All except one waited for a response and looked at the inbox, but one investigated the contact information page and read the phishing mail again. Participants' scan paths indicate that most of them read the warning as well as the message (See figure 4a). But none of them looked at the security indicator. The time to handle the phishing mail was 56.9 seconds on average and the participant spent 17.5*s* reading the phishing mail and warning on average.

In the group without warning, we observed three different reactions with three participants each. Three participants immediately replied and sent the sensitive information to the adversary. Two of them read the postcard pop-up while composing the response and one of them read the postcard pop-up when reading the phishing mail. None of them inspected previous received mail of the sender.

Their scan paths indicate that they carefully read the phishing mail and looked at the security indicator, sender and receiver fields (See figure 4d). The time to handle the phishing mail was 170*s* on average. They spent 32.56 seconds reading the first mail on average and about 135.66 seconds composing the response on average.

Three other participants invited the adversary, similar to the invitation of Bob in a previous task. In two cases, the adversary could not accept the invitation because the mail provider blocked the account. In one case the attacker accepted the invitation and responded with a secure mail. This way, one participant sent the information „securely" to the attacker. All of them read the postcard pop-up while composing a response and used the provided invitation function. One of them inspected the secure and insecure contact information view of the impersonated sender. None of them inspected previously received mail of the sender. Their scan paths indicate that most of them checked the security indicator, sender and receiver fields (see figure 4c). They spent 17.8 seconds reading the phishing mail on average and finished the task after 95.2 seconds on average.

The remaining three participants of the group without warnings were safe and sent an encrypted response to one of the previous



(a) Warning group (b) Participants who stayed safe (c) Participants who invited the phisher (d) Phished participants

Figure 4: Scan paths of participants while reading the phishing mail. Figure 4a shows the scan paths of all participants in this group. The other figures show the scan paths of subgroups of group B (without warning).

mail addresses of the company. Their scan paths indicate that they only read the phishing mail briefly and did not focus on the security indicator (see figure 4b). They spent 241.2 seconds finishing the task on average but only 14.2 seconds on average reading the mail for the first time. In contrast to other participants, they spent 12.9 second reading previous secure mails on average. They read the phishing mail at least two times. Before sending a secure mail they aborted composing an insecure mail.

We found a significant difference ($p = 0.041 < 0.05$) of successful phishing attacks between warning and no-warning group using one-tailed fisher's exact test. All but three participants stated in the survey after the experiment, that they were able to discern between the two security states. Two were not certain. Five participants said they noticed the different color themes besides the security indicators. One said that he maybe noticed it subconsciously. Two said they had seen the different colors, but didn't associate them with the security state.

task	actions	A	B
Bob forgot to encrypt	send data secure	9	9
	send invitation	0	0
	read previous mail before reply	0	0
	ask sender	6	0
Phishing attack	send data insecure	0	4
	send invitation	0	3
	read previous mail before reply	0	6
	send data secure	0	3
	ask sender	9	0

Table 1: Participants' reactions after receiving an insecure mail were a secure one would be expected. In group A a warning called extra attention to the security state of the mail and offered a proposed action.

6.5 Discussion

6.5.1 Usability. Our app received a SUS rating of 75.4 on average. Table 2 compares its SUS to the scores of mail apps studied previously. The SUS ratings of the warning group and the no-warning group differ but the standard deviations suggest that the measured difference is not indicative of a difference between the conditions. Post-condition interviews we conducted with the participants suggest that the usability hurdles our participants encountered were not caused by the security features but by missing mail management functions. This speaks in favor of extending an existing mail app rather than developing a new one from scratch because existing apps likely have a more complete feature set and users are likely already familiar with the interface.

Five participants did not realize that our app is a regular MUA. They believed that all mail sent and received with it were secure, comparable to messenger apps such as *Signal* or *WhatsApp*. One participant thought all mails were secure because our app ran on an iPhone and everything from Apple was secure.

scheme	type	SUS score mean	std
PWM 2.0 [26]	gmail web plugin	80	9.2
Our Prototype	**mobile mail client**	**75.4**	**11.7**
integrated MP [4]	browser plugin	75	14
standalone MP [4]	browser plugin	74	13
PWM [25]	gmail web plugin	72.7	16.5
Virtru [25]	gmail web plugin	72.3	13.7
Tutanota [25]	browser plugin	52.2	17.8

Table 2: Our mail client (bold) compared to previous studied secure mail clients.

On opening of the app for the first time, four pages tutored users on the possible security states and their properties. The participants spent on average 35 seconds (SD 15 seconds) on the tutorial. Even though the eye tracker data showed that all but one read the text, many could not actually remember the information. Two said that it would be useful to be able to get back to the information later.

6.5.2 Security. Participants in the warning group followed the recommendation and asked the assumed sender to verify the message. It seems that they were confident afterwards that the task was finished and they waited for a response.

Participants in the other group became unsure how to react. Most of them were bewildered by the phishing mail. They read the mail again, checked the security indicator, viewed previous mail (secure mail and the initial insecure one) or checked the contact information page. After considerable time, compared to the warning group, they fell back to learned or previous behavior. In the best case, they answered only to the known mail address and were safe. The invitation of the attacker may be considered behavior they learned from having invited Bob earlier. The analysis of the eye-tracker data showed how the participants looked for a solution to their problem: They would have to send an insecure mail containing sensitive information. They did not actually realize that the request for information was not legitimate.

This indicates that they did not interpret the situation correctly and tried to keep the information safe during transport despite the lack of trust in the receiver. The last group gave the information immediately to the phisher. They either did not grasp the situation or had no notion of how to solve the problem. Oddly, no participant in this group asked the sender to verify the previous message. One reason could be that the participants stuck to previously learned behavior and were unable to develop a new strategy in this new situation.

6.6 Limitations

Our study clearly suffers from the limitations that are usually associated with short-term laboratory studies in university settings. For example, habituation to warnings is a known effect in practice that we could not address properly because of the short duration of the experiment. Therefore, we cannot tell whether users of our app would eventually habituate to our warnings.

Furthermore, participants may misunderstand warnings as instructions of the experimenter. In order to counteract this effect we made the first warning occur in a false positive case so that participants have an opportunity to learn that warnings are merely warnings and not instructions on how to respond in a fashion that pleases the experimenter.

We recruited participants on campus and hence the participants reflect only a limited portion of the population. There remains a potential self-selection bias and a clear (albeit intended) bias towards iOS users. Most importantly, the number of participants was small. Therefore, we cannot expect our results to generalize to a larger population. Since we both designed and studied the application that was front and center of the study task we expected a risk of unconsciously influencing study participants. We tried to counter this risk by scripting the interactions we gave to participants and by providing them information and instructions in written form whenever possible.

Participants in the no-warning group took more time to react to phishing mail. The reason might have been that they were unsure how to accomplish the security goals of the task. On the other hand, they might have been merely unsure about how to proceed with the task itself since there were no specific instructions on what to do in the case of a suspected phishing (a subtle yet important difference).

A typical challenge of studying security mechanisms is that participants know or assume that any threats or risks to which they are exposed in a study context are not real. Consequently, participants may accept greater risks in simulated settings than they do in a real setting [27]. We have not come to a conclusion yet how to best address this limitation in agreement with basic ethical principles. Participants knew that they took no personal risk in our experiment and they needed to imagine to be in that particular situation. Furthermore, we did not simulate realistic mail traffic and realistic response times. This of course limits the ecological validity of our experiment and should be considered in future experiments.

We focused on a specific phishing scenario. Additionally, more complex cases need to be investigated, for example, multiple keys and mail addresses per contact as well as rolling keys over in a fashion that is forward-secure. Only then will we have a more complete picture of the kinds of trade-offs between usability and security we can achieve.

Our study uses a scenario in which it is plausible that users can inquire with assumed senders whether a mail received previously is indeed authentic. In other scenarios, for example, the PayPal scenario we mentioned in our introduction, this seems less feasible. On the other hand, it is implausible that a company such as PayPal sends unprotected mail subsequent to adopting signed mail.

7 CONCLUSIONS AND FUTURE WORK

Making end-to-end encryption of electronic mail easy to use to the point where it becomes ubiquitous has been a goal of researchers and activists for a long time. And yet the goal seems elusive. Despite Snowden's revelations, apparently few users see sufficient value in encrypting their mail to make the necessary effort. Approaches that reduce the complexity of widely available encryption tools do not seem to resonate with the vendors who would need to embrace them. At the same time, messenger apps lead the way to end-to-end encryption, leaving mail behind. Perhaps protection against

phishing in areas where mail is important would be suitable to entice vendors and users to embrace signatures as a protection mechanism and end-to-end encryption could be deployed along with it.

In order to investigate whether signatures can be effective against phishing we conducted a study with 18 participants. We studied whether recommending safe actions in response to suspicious mail helps users to steer clear of being phished. Specifically, we looked at the case where phishers spoof plaintext mail when a genuine sender would be capable of using signatures to authenticate mail (compared to no recommendation). Users responded positively to our recommendations and avoided being phished in all cases. The app we designed to provide end-to-end protection was rated quite usable despite a lack of mail management features. This gives us hope that opportunities exist to improve mail experience to the point where mail can compete with messenger apps again for personal communication. One of the greatest strengths of mail is that it is a provider-independent standard whereas popular messengers are proprietary walled gardens.

More research is necessary to investigate how users might behave in various edge cases we have not covered in our study, and how users can be supported in these edge cases. However, we are convinced that if messengers can succeed with end-to-end protection then so can good old mail.

8 ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments and Isabella Bargilly for her help with proof reading. The first three authors were funded by the Bundesministerium für Bildung und Forschung (Federal Ministry of Education and Research, Germany) under grant number 16KIS0360K (Enzevalos).

REFERENCES

[1] Eric P. Allman, Jon Callas, Jim Fenton, Miles Libbey, Michael Thomas, and Mark Delany. DomainKeys Identified Mail (DKIM) Signatures. RFC 4871, May 2007. URL https://rfc-editor.org/rfc/rfc4871.txt.

[2] Hazim Almuhimedi, Adrienne Porter Felt, Robert W. Reeder, and Sunny Consolvo. Your Reputation Precedes You: History, Reputation, and the Chrome Malware Warning. In *Symposium on Usable Privacy and Security (SOUPS 2014)*, SOUPS '14, pages 113–128. USENIX Association, 2014. ISBN 978-1-931971-13-3.

[3] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, 82(Supplement C):69 – 82, 2015. ISSN 1071-5819.

[4] Erinn Atwater, Cecylia Bocovich, Urs Hengartner, Ed Lank, and Ian Goldberg. Leading Johnny to Water: Designing for Usability and Trust. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 69–88, Ottawa, 2015. USENIX Association. ISBN 978-1-931971-249.

[5] Aaron Bangor, Philip Kortum, and James Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3):114–123, 2009.

[6] Frank Bentley, Nediyana Daskalova, and Nazanin Andalibi. "If a Person is Emailing You, It Just Doesn'T Make Sense": Exploring Changing Consumer Behaviors in Email. In *Proc. Conference on Human Factors in Computing Systems*, CHI, pages 85–95. ACM, 2017. ISBN 978-1-4503-4655-9.

[7] Mark Blythe, Helen Petrie, and John A. Clark. F for Fake: Four Studies on How We Fall for Phish. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3469–3478, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9.

[8] John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996. ISSN 1097-0193.

[9] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why Phishing Works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 581–590, New York, NY, USA, 2006. ACM.

[10] Julie S. Downs, Mandy B. Holbrook, and Lorrie Faith Cranor. Decision Strategies and Susceptibility to Phishing. In *Proceedings of the Second Symposium on Usable Privacy and Security*, SOUPS '06, pages 79–90, New York, NY, USA, 2006. ACM. ISBN 1-59593-448-0.

[11] Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, Michael Bailey, and J. Alex Halderman. Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security. In *Proc. Internet Measurement Conference*, IMC, pages 27–39. ACM, 2015. ISBN 978-1-4503-3848-6.

[12] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. You'Ve Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1065–1074, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1.

[13] Hal Finney, Lutz Donnerhacke, Jon Callas, Rodney L. Thayer, and David Shaw. OpenPGP Message Format. RFC 4880, November 2007. URL https://rfc-editor.org/rfc/rfc4880.txt.

[14] Ian D. Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage, and Kirill Levchenko. Security by Any Other Name: On the Effectiveness of Provider Based Email Security. In *Proc. Conference on Computer and Communications Security*, CCS, pages 450–464. ACM, 2015. ISBN 978-1-4503-3832-5.

[15] 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*, pages 13–24. ACM, 2005.

[16] 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*, CHI '06, pages 591–600, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7.

[17] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social Phishing. *Commun. ACM*, 50(10):94–100, October 2007. ISSN 0001-0782.

[18] D. Scott Kitterman. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208, April 2014. URL https://rfc-editor.org/rfc/rfc7208.txt.

[19] Dr. John C. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008. URL https://rfc-editor.org/rfc/rfc5321.txt.

[20] Joscha Lausch, Oliver Wiese, and Volker Roth. What is a secure email? *EuroUSEC '17*, 2017.

[21] Adam Lerner, Eric Zeng, and Franziska Roesner. Confidante: Usable Encrypted Email: A Case Study with Lawyers and Journalists. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*, pages 385–400, 2017.

[22] Franck Martin, Eliot Lear, Tim Draegen, Elizabeth Zwicky, and Kurt Andersen. Interoperability Issues between Domain-Based Message Authentication, Reporting, and Conformance (DMARC) and Indirect Email Flows. RFC 7960, September 2016. URL https://rfc-editor.org/rfc/rfc7960.txt.

[23] Pete Resnick. Internet Message Format. RFC 5322, October 2008. URL https://rfc-editor.org/rfc/rfc5322.txt.

[24] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy van der Horst, and Kent Seamons. Confused Johnny: When Automatic Encryption Leads to Confusion and Mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 5:1–5:12, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2319-2.

[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*, CHI '16, pages 4298–4308, New York, NY, USA, 2016. ACM.

[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*, UIST '16, pages 461–472, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4189-9.

[27] Stuart E. Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. The emperor's new security indicators. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 51–65, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2848-1. doi: 10.1109/SP.2007.35. URL https://doi.org/10.1109/SP.2007.35.

[28] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, pages 399–416, Berkeley, CA, USA, 2009. USENIX Association.

[29] Sean Turner and Blake C. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. RFC 5751, January 2010. URL https://rfc-editor.org/rfc/rfc5751.txt.

[30] Min Wu, Robert C. Miller, and Simson L. Garfinkel. Do Security Toolbars Actually Prevent Phishing Attacks? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 601–610, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7.

[31] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong. Phinding phish: Evaluating anti-phishing tools. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007)*. Internet Society, 2007.