MicroLock: A New Approach to Encrypted Email

Sam A. Markelon and Darian Jennings

Abstract— Digital communication and data exchange is now commonplace in contemporary life. With this comes the heightened need for security and privacy in digital messaging and file sharing systems. There is significant prior work that suggests existing encryption tools, and those specifically layered on top of email, suffer from a lack of adoption due to usability issues among non-technical users. To that end we present MicroLock, a simple standalone encrypted email tool, that hopes to buck this concerning trend and demonstrate overwhelming ease of use as compared to alternative technologies. We highlight the lack of usability of existing tools within the literature, motivate our approach to MicroLock, and present the design of our application. We also present a user study that shows MicroLock to be a promising approach for making encrypted email usable for the common user. We compare these results with studies on existing encrypted email applications.

Index Terms—Encrypted email, usable security, application usability.

1 Introduction

Over the past half century our lives have transformed due to the advent of computers and the internet. We live in a digital age where increasingly our social and professional lives exist on the internet. Our personal information is stored and transmitted online. With this has comes the rise of violations of privacy and digital crimes that span financial theft, internet blackmail, and political censorship.

Cryptography gives the individual a fighting chance against those who wish to do us harm in the digital age by providing us with tools to protect our information. Unfortunately, cryptography is a complex set of tools that is hard for even the experts to use correctly. Paired with the fact that digital applications are often not designed with security in mind, leaves many vulnerable to catastrophic online security failures whose effects can spill out in to the real world. Anyone who has ever had their credit card number stolen online can attest to this.

One such widely used and overlooked tool is email. Email is not secure by default, and even modern email clients like G-Mail leave users open to tracking and spying by adversarial entities. Therefore it is of interest to add encryption to email, particularly when sensitive information is being transmitted using the service. Pretty Good Privacy (PGP) [4] is an application that provides encryption and authentication of data and can be incorporated with various email and messaging applications. PGP requires deep knowledge about cryptographic primitives to use securely. The academic literature has shown that it is basically impossible for non-expert users to use PGP in a secure manner.

In turn, people have tried to make encrypted communication easier. Keybase [12] is a modern update to PGP, that provides a web accessible graphical user interface and various secure messaging and data storage modules. However, as exemplified by the Confidante study [7] Keybase still left many users unable to send an encrypted email and users were unable to form a mental model that matched up with how the system actually functions. Therefore, due these discouraging results we developed MicroLock, a simple and secure messaging tool. We specifically designed MicroLock such that it focuses on the task of allowing users to send encrypted messages without arduous setup or worry about security decisions. We rid our application of technical jargon and distill the operations of encrypting and decryption messages down to two simple operations: *lock* and *unlock*. Moreover, we strip away the complexities of key management by introducing a primitive we call a "secure contact". In this way keys can be imported and exported in a method akin to saving a contact to one's mobile phone or email client.

In this paper we present a survey of the current literature showing the usability issues of current encrypted email solutions. We then present our motivating principles and design of MicroLock, then demonstrate a simple workflow. Further, we present findings from a user study which show encouraging usability results as it pertains to users naturally understanding the process for securely encrypting and decrypting messages. We then compare these results with previous studies from alternative solutions in the space. We lastly discuss the limitations of our work and future work in this line of inquiry.

2 Related Work

The seminal 1999 paper by Whitten and Tygar, *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*, was the first to explore the usability of encryption applications [13]. The results are calamitous. Even though PGP 5.0 has a good interface by standard design heuristics, the software is simply too complex to use in a secure manner by cryptography novices. A cognitive walk through by the authors pointed out many potential critical and irreversible security errors that could arise from using the software. These concerns were affirmed by a user study of non-experts. The majority of the twelve users were not able to complete a simple *encrypt and sign* and *decrypt and verify* task in an hour-and-a-half time allotment. The few that were expressed inconsistencies in their mental model of how the system functioned that could lead to future catastrophic security errors.

A lineage of literature follows that comes to the same conclusion. In [9] a 2015 PGP client (Mailvelope) that integrated with existing email clients was shown to be unable to be used by non-experts in a secure manner. Of the ten pairs of twenty total participants, nine of the pairs were unable to complete a key generation, encryption, and decryption task. The trend continues in [7] with a study of the Confidante encrypted email system, based on Keybase. The qualitative usability study, which focused on the use of the system by lawyers and journalists, found that the design of the system made a number of critical security errors impossible - such as sending a plain text email. However, the complexity of the system lead to mental model errors that made a number participants unable to complete the encryption task at hand. In addition, general confusion on how to use the software, and the role of Keybase accounts versus public keys lead to further user frustrations.

The secure usability concerns of encryption software extends to end-to-end encrypted messaging applications like WhatsApp and Signal. In particular, users were unaware that by default these applications do not run in an authenticated mode and are subject to malicious man-in-the middle attacks [5].

Conversely, work has been done to try to identify what needs to be done to make encrypted email applications both secure and usable. Of

- *Sam A. Markelon is with the University of Florida. E-mail: smarkelon@ufl.edu.*
- *Darian Jennings is with the University of Florida. E-mail: darian.jennings@ufl.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

highest priority includes integration into existing mail clients, giving context hints during the encrypted email workflow, secure options as defaults, and (seemingly the most important and least addressed) simple key management [10]. On the last point work has specifically been done to investigate what type of key management is most usable for encrypted email applications. The results were largely inconclusive. Of the three examined – password, public-key directories, and identity-based encryption schemes – all have similar system usability scale scores and surprisingly equally divided preference among study participants [8]. However, not considered are the deployment constraints that make password and identity-based encryption schemes untenable in a number of scenarios when designing an encrypted email application for general use.

Looking more broadly at the usable security literature a number of common themes arise. It is suggested that security applications demand too much of the user and require a user to have a mental model that is simply too complex to be functional. Instead it it suggested to wrap complex security operations in a simple task-action model – that is say hit a button and encrypt the message. [11]. Moreover, the key takeaway from a review of field studies of usable security deployments suggests that you cannot bolt on usability to poorly designed security applications. [1] That is, one should not expect to take a system like PGP 5.0 and add contextual pop-up pointers to the user during their workflow and expect to suddenly make the system usable. Further, a systematic review of the usable security literature [6] suggests that the graphical user interface and steps to complete a security tasks must be as simple as possible to reduce cognitive load on the user. Also remarked upon in the review, is that it is essential that users have trust in the system for them to actually want to use it and a need for providing in-application context for complex security operations.

3 MicroLock Application

3.1 Guiding Principles

It is clear that designing an encrypted email application that is easy to understand and use for even novice users is a daunting task. However, the existing literature provides a number of key points that we feel can be leveraged to aid us in our work. Our guiding question is: *Why should a user need to know anything about cryptography to use a secure email application?* We do not make demands of the user to know the underlying technical details of any other software product. Users should not be made to learn any cryptographic jargon or operations to use an encrypted email application. Moreover, it is clear from the literature that we need to lessen the number of steps to complete the simple task of encrypting an email to lighten the users cognitive load and avoid frustration.

Therefore, we use the following principles to design MicroLock. We eliminated technical jargon, hid the complexity of the cryptographic operations from the user, and reduced the number of steps in sending and receiving encrypted emails. We distilled down the functionality to the simple and core paradigms of locking and unlocking a message. Moreover, we made secure cryptographic protocol choices by default so users do not need to worry themselves about options. Finally, we made a simple way to share keys which we coin as "secure contacts". One can export a secure contact as a simple text file, and importing one is akin to adding a new friend's information into a mobile phone of email client.

3.2 Design Overview

We designed MicroLock as a standalone message encryption tool. We serve a local web page using Flask to provide a front-end. We wrote a simple hybrid encryption scheme with the help of the standard cryptography library of Python. The basic scheme encapsulates Fernet authenticated symmetric encryption keys and the encrypted payload using RSA-4096 public keys which users exchange with one another.

We have a login page for a user that loads their public key and password protected private key to use within the system while they are logged in – in our verbiage this is the private part of the secure contact that is stored on a user's system. Alternatively if the user does not yet have a secure contact they can generate a public, private key pair by entering their name, email, and a password to secure their private key via a password-based key derivation function. This effectively generates a secure contact for them; the public part of which they can export and share with those they wish to communicate with. We remind the user that it is essential to choose a strong password and to remember said password.

In this way our system creates and accesses private, public key pairs like any login page for an online service - something that most users are bound to be familiar with. Moreover, we export keys and associated data (name and email address) in a simple text file format. In this way users can upload or paste this file to a personal website, key server, social media site, email, or text message.

Users are able to import contacts from these simple public key files into the MicroLock application contact directory as well as having the option to include their public key in an encrypted message. This gives the user flexibility in not only how to disseminate their public key to others, but also makes our application agnostic to what email or messaging client encrypted messages are passed over.

Again, our approach is to avoid complex technical jargon to make the user's mental model as simple as possible. For instance we aim to avoid technical terms like "encrypt with authentication and sign" and "decrypt and verify". We substitute with the more digestible terms "lock" and "unlock". Further we do away with the user needing to worry about keys and simplify a private key as a password protected account, and a public key certificate as a simple text file (secure contact) they can share with others.

3.3 Walkthrough

We will now present a walk through of the use of the prototype of MicroLock that exhibits the principles mentioned above. In the scenarios we have two parties, Sam and Darian. In the following example Sam is going to send Darian a secret message that Darian later decrypts.

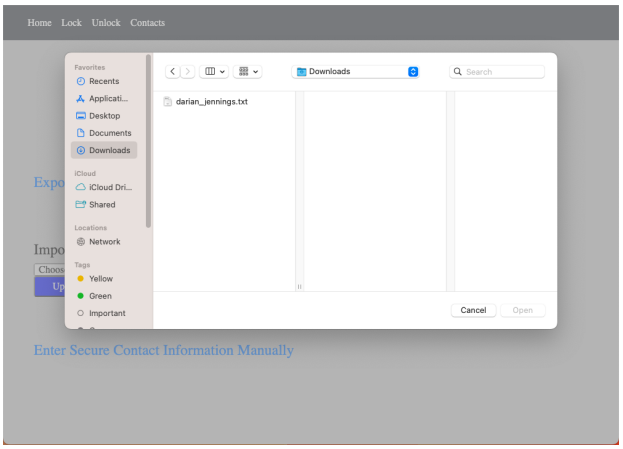

Fig. 1. Sam imports secure contact for Darian.

We assume that Sam has created his account and has logged in to the system. That is he has a public, private key pair and is able to encrypt and decrypt messages. We first serve with a home page that includes instructions for using the MicroLock tool, and provides information about the underlying technology that helps to instill trust in the use. A navbar at the top of the page allows the user to navigate to different pages, each serving a single functionality. After navigating to the contacts page, Sam first imports Darian's secure contact into his contact repository, Figure 1.

In Figure 2 Sam navigates to the lock page (which encrypts, authenticates, and signs the plain-text message) and selects Darian as his recipient. On hitting the lock button, the output is then authenticated and signed cipher-text or as we present it the locked message shown in Figure 3. The locked message can be copied an pasted into the messaging platform of choice and sent to Darian, such that Darian and

Fig. 2. Lock page of MicroLock.

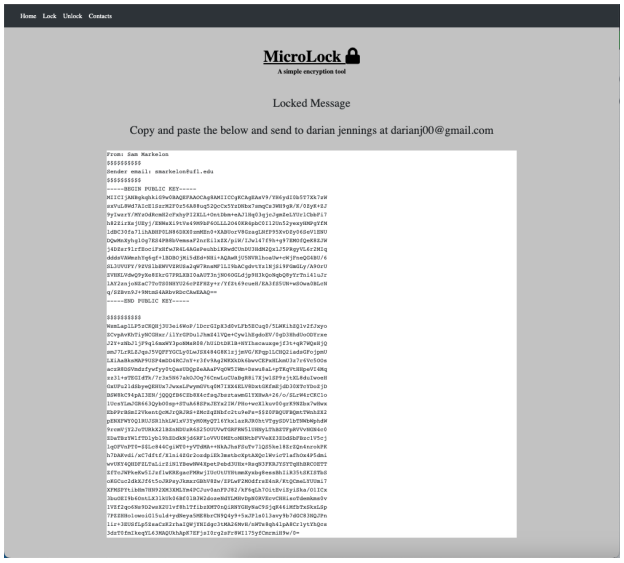

Fig. 3. The locked message.

only Darian (or someone that possesses his private key) can unlock and read the message.

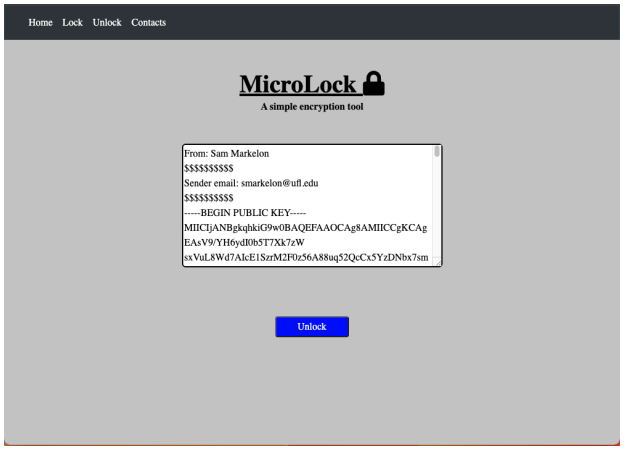

Fig. 4. Unlock page.

After Darian receives the message over the communication channel

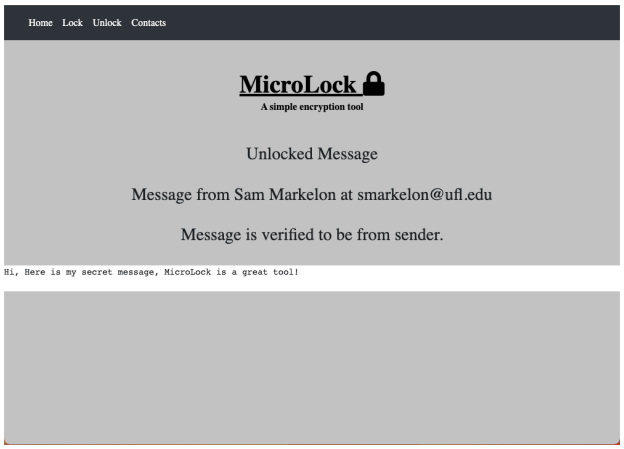

Fig. 5. The unlocked message.

he can paste the locked message into the unlock page of MicroLock as seen in Figure 4. The system unlocks the message by unpacking the sender information, symmetric key and message, followed by running the symmetric decryption algorithm, and verifying the digital signature of the sender. The result is the unlocked message with the status showing that the message was successfully verified from Sam as shown in Figure 5.

This walk through exemplifies the fact that we strip away complex cryptographic processes and distill them down to two simple operations: *lock* and *unlock*. These operations use secure cryptography by default, which in turn limits the number of catastrophic security errors it is possible for a user to make. Moreover, recall that MicroLock runs as a small local applications on the end-user's computer, therefore enabling them the power of choice on which communication platform they send encrypted messages over. Further, importing keys is made simple by having public key certificates exported as simple text files that are easy to import into one's MicroLock contact repository.

4 User Study

After approval from the University of Florida's Institutional Review Board (IRB), we conducted a user study gathering a total of 11 participants. We began with a pre-questionnaire where we collected demographic information, asked about familiarity with cryptography and encryption, as well as there usage of email clients. Then the participants were provided a brief overview of what the MicroLock application is, what it is used for, and how it works. Afterwards they were instructed to complete two different tasks, one was encrypting a message and the other was decrypting an encrypted message. After the participants were done we had them complete the post-questionnaire which were questions from the System Usability Scale (SUS). The SUS is a quick, standard, and reliable tool for measuring the usability of an application [3].

4.1 Study Participants

We recruited 11 participants from the university population. In the Discussion section we refer to these participants as $P1, P2, \ldots, P11$. All were aged 18-25 and were undergraduate students or graduate students. All regularly used email, but none were highly technical in that none of them were able to give an accurate and precise description of public key cryptography.

Two of the participants had used encrypted messaging applications like Signal or Telegram, but none had used encrypted email applications before. Nine people expressed worry about the privacy of their email communication being compromised, and two had been the victims of an online attack.

4.2 Overview of MicroLock

We gave the users a brief overview of the MicroLock application, but did not instill any domain specific knowledge about cryptography. The following is the overview we gave them.

MicroLock is a secure email system that uses state of the art security to secure your communications. The technology behind it is provably secure and thus ensures that only you and your intended recipient will be able to access your data once you lock a message and send it off.

To start you'll need to create a secure contact and share that contact with the parties you intend to communicate with. Your secure contact has two parts: a secret part and a public part. The secret part of your secure contact stays on your computer and is password protected. The public part is able to be shared with others and will allow them to send you secret messages and receive secret messages from you. You can export the public part as a text file, and share it via email or post it on a social media site.

To send a secret message, navigate to the Lock page of MicroLock, write in your text entry, and then hit the lock button. Remember, you will need to have the recipients secure contact to do this. Then you can simply send over the resulting locked message via your messaging platform of choice, by copying and pasting it into said platform and sending it to your desired recipient.

If you receive a secret message you can use MicroLock to unlock it. Simply copy and paste the locked message into the Unlock page of MicroLock, hit the unlock button and you will get out the original message.

4.3 User Task

After this overview, participants were then asked to role-play two secure messaging scenarios – all participants were asked to role-play the exact same scenarios. No sensitive personal data was collected and all simulated personal data was obviously synthetic.

In the first scenario we had the participant assume the role of John Smith and provided them with the credentials to log in to John's account. The task was to send an encrypted message to their tax advisor Jane Smith. We further informed the used that we had already imported the contact for Jane Smith into John's contact repository and provided them with the following prompt: *Hello my name is John Smith, here is my social security number, so that you can look into my taxes for this past year: 123-45-6789.* Once the user was able to lock the message, we then simulated an email client and told them to paste the resultant locked message into the draft email box and act like they were sending the secret message via email.

We then notified the user they received a response back from Jane in the same simulated email client and had them attempt to unlock Jane's message and read it out.

In line with previous work in the space (and highlighting the current low-bar that is task completion) we had the simple binary measure of whether the user was able to complete both tasks or not.

5 Results

5.1 Comparison to Alternatives

Encouragingly, all of our participants (11/11) were able to complete the tasks that we outlined in the user study. While not precisely measured, all were able to complete the tasks within a few minutes (less than 5 minutes total). We can compare this to 4/12 participants who were able to complete an encryption and decryption task in the original PGP 5.0 usability study [13] and 1/11 participants who were able to complete an encryption and decryption task in the Mailvelope study [9]. Both these studies used very similar tasks descriptions as our study. However in each of these previous studies participants were given a one-and-one-half hour time limit to complete the task. This is obviously something that we did run up against.

We apply two separate one-tailed Fisher's exactness test to see if users were more likely to be able to complete the encryption/decryption task using MicroLock as compared to the alternatives (one for PGP 5.0 and one for Mailvelope). A Fisher's exactness test is useful for measuring the significance of the a contingency between two classifications, particularly when you have a small sample size. In this case we compare the contingency of users of MicroLock versus alternative applications in being able to complete the task versus not complete the task. An example contingency table for MicroLock versus PGP 5.0 is shown in Table 1.

While, this may not be entirely proper as the tasks for all studies were similar but not exactly the same, and we did not personally conduct the previous studies. However, at $\alpha = 0.05$ and under the null hypothesis that users were equally likely to be able to complete a simple encryption and decryption task with MicroLock and the alternative tool we see using this test we have evidence to suggest that users of MicroLock are betterable perform an encryption/decryption task than users of alternative secure email tools. That is, we have statistically significant evidence that users perform better using MicroLock than PGP 5.0 ($p = 0.0013$) and statistically significant evidence that users perform better using MicroLock than Mailvelope ($p = 0.00003$).

5.2 SUS Score

Further, from the user study post-questionnaire we computed a SUS Score for MicroLock equal to a 76. Which translates to an *acceptable* or a grade of a *B* on the standard interpretation scale [2]. This is in comparison to Mailvelope which scored an *F* or *not acceptable* on the standard interpretation scale with a SUS score of 34.5.

A the majority of complaints received from participants in our user study were due to a lack of integration into existing email platforms and application layout rather than the core security functionality and workflow. While we acknowledge these as legitimate concerns, and certainly something we will integrate into future work tasks, we are highly pleased that our decision to abstract away almost all of the highly technical cryptographic terms and operations seemed to be of great success in having users deem our application as usable.

6 Discussion

We will summarize a number of user comments and contextualize them with regards to our approach.

Workflow All participants started by reading the instructions on the homepage. We provided instructions to help the user and instill trust in the system as recommended from prior work in the usable security literature. After reading the instruction many participants proceeded right to the lock tool to complete the first task, however participants P3, P4, and P5 went to the contacts page first. They explained that they expected to see a list of contacts they already had imported, and to be able to click on said existing contacts to draft them a secret message. We identify this as a familiar interaction paradigm with-in mobile phone systems and will update MicroLock to replicate this functionality.

Application Design Participants P4 and P7 had trouble finding the navigation bar at first, and remarked that it should be larger and more noticeable. Also noted among participants P6 and P7 was the desire to streamline the process for sending messages so we do not need to move from application to application. On the other hand, P2 liked that this process gave them flexibly in which application they choose to send the message through. P1 desired the application to be integrated with iMessage – their communication platform of choice. With these remarks in mind we will work to integrate the application with a number of email and messaging clients, in addtion to overhauling the graphical user interface to be in line with contemporary standards.

Cryptographic Functionality All users seemed to grasp the lock and unlock paradigms rather easily. We did note a positive learning effect in which all participants took a bit of time to lock their first message, but were almost immediately able to unlock a message in the second task. P3 likened the locking and unlocking of a message to a language translator translating a word from one language to another and back to the original language. This abstraction is helpful in thinking about how non-technical users view the encryption and decryption process.

P5 thought *lock* and *unlock* was too technical and would just prefer the terms *send* and *recieve*. However, P5 was able to complete the

	MicroLock	PGP 5.0	Marginal Row Totals
Completed	11	4	15
Did Not Complete	0	8	8
Marginal Column Totals	11	12	23 (Grand Total)

Table 1. The contingency table for task performance of MicroLock versus PGP 5.0.

task without problem and many participants – P2,P6,P8,P9 – remarked that the operations of locking and unlocking was straightforward. This suggests that we have abstracted the underlying cryptographic functionality to such a degree that even non-experts are able to comfortably encrypt and decrypt messages with this simple task-action paradigm we have adopted.

7 Limitations & Future Work

Our participant sample size, n = 11, was relatively small. While in line with the standard usable security methodology for examining the usability of encryption applications we would like to expand the size of our study future work. Moreover, we would like to expand the scope of the user study we conduct by running a with-in subject study such that we can directly compare user performance when using MicroLock versus alternatives. That is we will have users complete the same set of tasks using multiple different encrypted email applications and measure their performance using each application. In this way we can do away with any possible confounding variables when comparing performance between MicroLock and alternatives.

Moreover, our study excluded the use of clients generating personal MicroLock accounts and their own email clients in the user task. Ideally we want participants to write and encrypt a message using our system, send off that encrypted message via an email client of their choice, and then receive an encrypted response of which they would use our system to decrypt. Due to the limitation of not being able to have users install MicroLock on their own devices or having access to a bevy of test computers coupled with time pressures we were unable to do this. In the future, we would like to seek further IRB approval to modify the task to match the above, and carry out said study.

Lastly, we would like to update our application beyond its current iteration to better integrate into common email and messaging clients. Further, we will update the graphical user interface to make it conform to more modern standards and follow some of the recommendations from our study participants.

8 Conclusion

Prior work shows that encryption tools have poor adoption and usability from both technical and non-technical users. MicroLock serves as a promising step in the right direction. Our approach of eliminating technical jargon, hiding complexity, reducing the number of steps, and making secure choices by default proved to be critical design components that led to our superb success rate. Future work will be done to improve the system and conduct a larger user study focused on comparison to leading alternatives.

Acknowledgments

We give special thanks to our study participants who provided valuable insight and suggestions for our project. We also give thanks to Dr. Ragan for his support and feedback throughout this research project.

References

[1] D. Balfanz, G. Durfee, D. K. Smetters, and R. E. Grinter. In search of usable security: Five lessons from the field. *IEEE Security & Privacy*, 2(5):19–24, 2004.

[2] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[3] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[4] S. Garfinkel. *Chapter 1: PGP Overview*, p. 1–31. O'Reilly and Associates, 1995.

[5] A. Herzberg and H. Leibowitz. Can johnny finally encrypt? evaluating e2e-encryption in popular im applications. In *Proceedings of the 6th Workshop on Socio-Technical Aspects in Security and Trust*, pp. 17–28, 2016.

[6] M. Lennartsson, J. Kävrestad, and M. Nohlberg. Exploring the meaning of usable security–a literature review. *Information & Computer Security*, 29(4):647–663, 2021.

[7] A. Lerner, E. Zeng, and F. Roesner. Confidante: Usable encrypted email: A case study with lawyers and journalists. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 385–400. IEEE, 2017.

[8] S. Ruoti, J. Andersen, T. Monson, D. Zappala, and K. E. Seamons. A comparative usability study of key management in secure email. In *SOUPS@ USENIX Security Symposium*, pp. 375–394, 2018.

[9] S. Ruoti, J. Andersen, D. Zappala, and K. Seamons. Why johnny still, still can't encrypt: Evaluating the usability of a modern pgp client. *arXiv preprint arXiv:1510.08555*, 2015.

[10] S. Ruoti and K. Seamons. Johnny's journey toward usable secure email. *IEEE Security & Privacy*, 17(6):72–76, 2019.

[11] M. A. Sasse and I. Flechais. Usable security: Why do we need it? how do we get it? O'Reilly, 2005.

[12] A. Spooner and S. Coates. *Chapter 1: Your Account and Chapter 2: Chat*, p. 1.0–2.4. Keybase.

[13] A. Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *USENIX security symposium*, vol. 348, pp. 169–184, 1999.