2017-10-01

Usable Secure Email Through Short-Lived Keys

Tyler Jay Monson
Brigham Young University

Usable Secure Email Through Short-Lived Keys

Tyler Jay Monson

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Kent Seamons, Chair
Daniel Zappala
Jacob Crandall

Department of Computer Science

Brigham Young University

ABSTRACT

Usable Secure Email Through Short-Lived Keys

Tyler Jay Monson
Department of Computer Science, BYU
Master of Science

Participants from recent secure email user studies have expressed a need to use secure email tools only a few times a year. At the same time, Internet users are expressing concerns over the permanence of personal information on the Internet. Support for short-lived keys has the potential to address both of these problems. However, the short-lived keys usability and security space is underdeveloped and unexplored. In this thesis, we present an exploration of the short-lived keys usability and security design space. We implement both a short-lived keys and a long-term keys secure email prototype. With these two prototypes, we conduct a within-subjects user study. Results from our study show that participants believe the short-lived keys prototype is more secure and more trusted. Participants also provide feedback on what they want in a system supporting short-lived keys. They also discuss how concerned they are about the permanence of their information on the Internet and on their devices.

Keywords: Secure email, short-lived keys, usable security

ACKNOWLEDGMENTS

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Despite the rapid growth of instant messaging and Internet chat, email remains an important communication tool in homes and businesses [6]. While messaging applications with end-to-end encryption, such as WhatsApp, have seen widespread adoption among those with little or no computer security knowledge, secure email has yet to see such adoption rates [27]. Without increased adoption of secure email tools, sensitive information in emails will continue to be vulnerable to mail providers, governments, and malicious entities.

The lack of end-to-end encryption in standard email leaves it vulnerable to malicious entities. Even though many major email providers ensure emails are protected by Transport Layer Security (TLS) while in transit to and from mail servers, most email providers do not use encryption to protect emails they store on their mail servers [15]. Storing emails in plaintext creates an opportunity for entities to obtain emails through several avenues. For example, plaintext emails can be obtained through forceful compromise (e.g., hacking) or legal action (e.g., subpoena). While plaintext emails let email providers filter spam, they also present opportunities for data mining, targeted ads, and even stalking [21]. Further, many email users in the United States are unaware their government can exercise provisions in the Electronic Communications Privacy Act (ECPA) to lawfully obtain emails that are at least 180 days old without a subpoena [24].

Applying end-to-end encryption to email is one way to overcome these vulnerabilities. End-to-end encryption secures email messages in transit and at rest. Most research

1

on usable secure email focuses on long-term key management. However, long-term key management has the potential to introduce usability and forward secrecy concerns. In terms of usability concerns, novice security users tend to make mistakes and are prone to feel overwhelmed when required to make long-term key management choices [38, 46]. Long-term keys pose a forward secrecy risk, because many secure messages are protected by one key pair and may be exposed if the private key is compromised. Given these concerns, long-term key management models may be insufficient to meet users' usability needs and concerns about information permanence.

Feedback from users in recent secure email studies has revealed that many users want to use secure email, but only expect to utilize it a few times a year at most [29, 31, 34]. Some users also have fears about information permanence. This arises from uneasiness about the permanent nature of personal information once it enters the Internet [25, 35, 44, 47]. This permanence of personal information can affect friendships, business reputations, job opportunities, and more.

Secure email tools in general support long-term keys, placing increased demands on users to perform necessary key management tasks. For example, users may be required to securely store their keys, or securely move them from one device to another, but may not know how to safely do so. Their inexperience may even lead them to divulge a private key, or trust an unverified public key. In one lab study, a participant divulged their private key while trying to send a secure email [30].
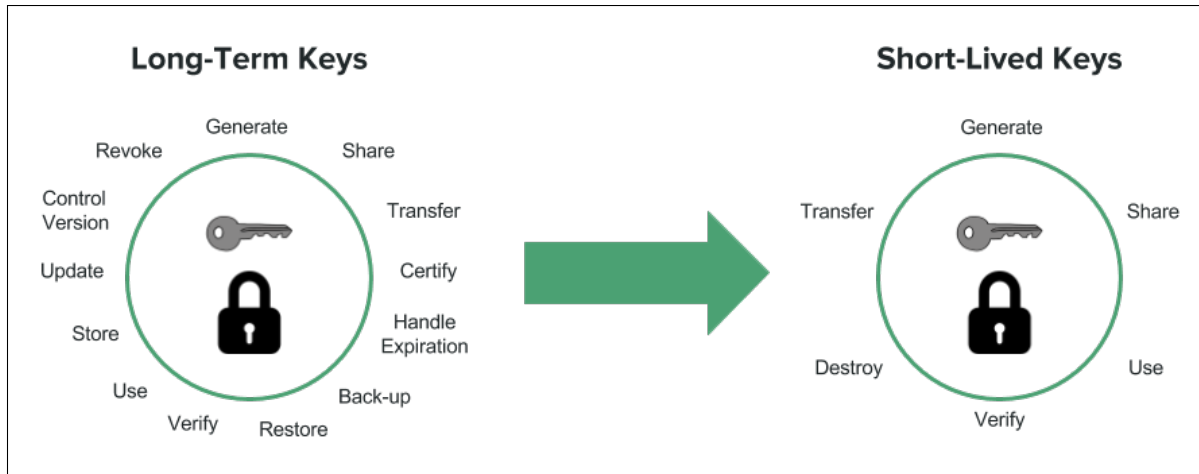
Figure 1.1: Key management operations needed for long-term and short-lived keys.

Traditional secure email tools assume users will generate a single keypair to protect their messages over the course of several years. If the private key is stolen, all encrypted messages could be compromised. Secure email tools that support forward secrecy would alleviate this concern, because compromised private keys would only be able to decrypt a limited number of messages [9].

One approach to addressing these problems is using short-lived keys. With short-lived keys, key pairs expire after several secure messages are exchanged, or after some period of time has passed. A tool using short-lived keys destroys public and private keys as they expire. This means that users no longer need to manage a long-term key pair. Instead, they generate, use, and destroy a key pair every time they send encrypted email. This eliminates the need to store long term keys, as well as the need to transfer keys between devices. Further, a level of forward secrecy is obtained since only a few messages are encrypted with a single private key. In contrast to these benefits, short-lived keys introduce an increased burden in terms of sharing, acquiring, and verifying new public keys.

Short-lived keys have been mentioned in standards documents [11, 14] and discussed in academic papers [8, 12, 37]. While these works introduce the challenges, benefits, and

3

design choices of short-lived keys, there has been no in-depth exploration of the concept. Further, to the best of our knowledge, no secure email tool designed specifically to support short-lived keys has been created or subjected to a user study.

In this thesis, we provide a detailed exploration of the short-lived keys design space, including the usability and security trade-offs for a variety of short-lived keys design choices. A short pilot study was conducted to further refine the results of this exploration. This exploration and pilot study reveal that tools supporting short-lived keys have many design alternatives. The results of this exploration can serve as a guide for designing and implementing a short-lived keys prototype.

Using the MessageGuard framework [33], we implemented two secure email prototypes. Both prototypes utilize an email-based public key exchange. One prototype supports a long-term keys model, while the other supports a short-lived keys model.

To understand the usability of our prototypes and about user perceptions of information permanence, we conducted a within-subjects, paired-participant [31] user study with a total of 24 participant pairs (48 participants total). Each participant in the study completed secure email tasks and information permanence tasks with both prototypes. Participants also participated in a 10–15 minute semi-structured exit interview where they gave feedback on the prototypes, as well as their worries and perceptions of information permanence in terms of messages on their devices and the Internet in general.

As we prepared for our user study, we made the following hypotheses:

- *H1*: Users will believe the short-lived keys prototype is more secure than the long-term keys prototype.

- *H2*: Users will trust the short-lived keys prototype more than the long-term keys prototype.

- *H3*: The short-lived keys prototype will receive a higher Systems Usability Scale score than the long-term keys prototype.

The results of the study show a statistically significant difference between the System Usability Scale scores of the short-lived keys and long-term keys prototypes. These results are complicated by the long-term keys prototype not having functionality for making secure email messages unreadable, while the short-lived keys prototype did. While it is impossible to infer participants preferred short-lived keys over long-term keys, qualitative results indicate that participants did have a strong preference for functionality allowing them to make their secure messages unreadable.

Further analysis of the data gathered from the exit interviews indicate that participants trusted the short-lived keys prototype more than the long-term keys prototype. Further, participants felt the short-lived keys prototype was more secure than the other prototype and they liked its ability to make messages unreadable at the push of a button. Among other things, the qualitative results of this study also indicate that while participants are not too concerned about the permanence of their messages or of their information on the Internet, participants do employ various strategies to limit the sensitive information they disclose online.

The contributions of this thesis are as follows:

An extensive evaluation of the usability and security of short-lived keys. We explore short-lived keys security and usability design choices related to key coverage, key expiration, automation of key destruction, presentation of encrypted messages, secure long-term information storage, using multiple devices with short-lived keys, and key state synchronization.

The first implementation and usability study of a short-lived keys secure email prototype. To the best of our knowledge, a short-lived keys secure email prototype has never been implemented or tested before.

Quantitative and qualitative results from a user study comparing secure email prototypes with long-term keys and short-lived keys models. In general, we gathered valuable feedback about our two prototypes, with much of the qualitative

feedback clarifying quantitative results. Even though much of this feedback is focused on short-lived keys in secure email, some of it applies to short-lived keys in other contexts and some of it applies to the usability and security of secure email regardless of whether short-lived keys are supported.

Feedback from participants showing they want ephemerality for emails containing sensitive information. Odom et al. [26] showed users desire permanence and ephemerality for different digital possessions. Displaying the importance of email permanence in some cases, Cecchinato et al. show that email users frequently use email archives to retrieve important information [13]. Others have proposed expiration dates for email [43] and ephemerality for email [19]. In contrast to this related work, we have gathered participant data showing users want their emails containing sensitive information to be impermanent and want other emails to be permanent. This feedback is based on participant experience with a secure email prototype that expires and makes secure emails unreadable.

6

<center>**Chapter 2**</center>

<center>**Background**</center>

This chapter contains an introduction to work related to usable secure email, short-lived keys, and user concerns about the permanence of personal information.

2.1 Usability of Secure Email

Formal user studies of secure email started in 1999 when Whitten and Tygar [46] conducted a user study of PGP 5.0. This study revealed that users were unable to successfully send encrypted email due to key management difficulties and misunderstandings of the cryptographic scheme at play. The results demonstrated to the security community that security without usability does not lead to any practical security improvements for users.

Years later, Sheng et al. [38] conducted a pilot study comparing the usability of PGP 9 to the results from Whitten and Tygar's work [46]. They determined that users still struggled with key management. Excessive hiding of cryptographic operations also made users unsure whether the operations were executed. Other results outlined the need for clearer interfaces to better guide user interaction with secure email software.

Garfinkel and Miller [17] conducted a modified version of the Johnny study [46] with a secure email prototype based on S/MIME. Results from the study indicated usability of secure email can be improved by applying automated key generation, key management, and message signing. Usability data also suggested the prototype hid too many cryptographic operations, indicating a balance needs to be struck between what is revealed and what is kept hidden in the cryptography.

<center>7</center>

Ruoti et al. [29] created and compared two versions of Pwm, a secure email prototype utilizing Identity Based Encryption (IBE) [29]. Studies with these prototypes showed users prefer secure email tools that integrate tightly with their webmail platform. Other results showed users felt more secure using a secure email system with less hidden cryptographic operations.

In response to the Johnny study [46], Tong et al. [40] developed a new set of metaphors to help users better understand cryptographic actions. The new metaphors were introduced to users using narratives centered on communication between King George III and his subjects. Even though these lock and key metaphors did not "dramatically outperform" standard PGP metaphors, far less documentation was required to help users understand them.

In 2015, Atwater et al. [2] revisited a study by Ruoti et al. [29] by comparing a standalone secure email prototype called MessageProtector to Mailvelope. Their results indicate users still prefer to use secure email tools integrated into their webmail. They also use their results to develop a set of principles for designing "consumer-friendly end-to-end encryption tools".

Bai et al. [3] conducted a user study evaluating the usability/security tradeoffs users make while choosing a secure messaging tool. After receiving descriptions of the security properties of the tested systems, participants recognized that the less-convenient system was more secure and labeled the security of the more-convenient system as "good enough".

Ruoti et al. [31, 34] conducted several more secure email usability studies with interesting results. One important study introduced a novel user study methodology that uses pairs of participants to better simulate grassroots adoption scenarios of secure email tools [31]. User pairs acted more naturally in the user study scenarios, because they were acquainted with each other before starting the study. In another study, Ruoti et al. [34] leveraged past work [29, 31, 33] to create, adapt, and use MessageGuard to compare secure

email prototypes using three different key management schemes. Using MessageGuard as a base for the three prototypes removed confounding factors from the usability study of the three prototypes. This allowed for the usability of the key management schemes themselves to be more directly compared.

2.2 Short-Lived Keys and Forward Secrecy

The earliest work discussing the utility of short-lived keys is from Schneier and Hall [37] in 1997. The work argues that past, present, and future messages can be exposed if attackers compromise a private key by guessing the key or by guessing the password protecting it. One defense against this vulnerability is minimizing the sensitive messages that are protected by one key pair. This defensive measure provides a form of forward secrecy as more and more key pairs are used to secure different messages. The work also argues for short-lived keys being signed by a long-term key pair to enable the owner of the short-lived key to be identified.

Brown et al. [12] expounded on Schneier and Hall's discussion of short-lived keys. The ideas presented in this work were later included in an Internet Draft titled "Forward Secrecy Extensions for OpenPGP" [11]. In contrast to Schneier and Hall's work, these documents focus more on discussing the logistics and usability of short-lived keys. They argue that short-lived keys require different key management than long-term keys. For example, the timeliness of destroying a private key is crucial in a short-lived key model.

Brown also discusses one potential complication for short-lived keys—worse usability. Brown presents this complication as a cost tradeoff between security and key distribution. This tradeoff is made as short-lived keys trade higher security for more complicated key management. While a more complicated key management method may inflict more usability problems overall, it is argued that these usability problems can be limited by strategically applying automation. For example, new key pairs can be automatically generated in the background and new public keys can be automatically

9

attached to emails as they are sent. The need for key expiration and key revocation warnings are also discussed.

Both documents state that data secured by short-lived keys can be extracted, encrypted, and stored locally using a long-term key. This scenario is necessary if encrypted data is needed after the key pair protecting the data expires. Transmitting data protected by a short-lived key and storing it locally with a long-term key reflects security recommendation 7-3 from NIST's recent document on Trustworthy Email [14].

Finally, Brown et al. [12] also argue that one-time keys, public/private key pairs used only for securing a single message, are the logical conclusion of a short-lived keys. In the context of using one-time keys to secure email, Brown et al. further argue that the nature of email is better suited to an offline key exchange scheme.

Short-lived keys are also discussed in Boneh and Franklin's seminal work detailing Identity Based Encryption (IBE) [8]. In IBE, a user proves their identity to a key server to gain access to a private key generated with a string related to their identity (e.g., email address). Other users can request individual public keys based on identity strings. IBE enables short-lived keys by combining an identity string with a date string. In one scenario, a new private and public key is generated for each user every day using their identity string and the date. This forces users to obtain a new private key every day so they can decrypt new messages. The paper indicates this system would be more feasible to use and maintain in a corporate environment. This work does not discuss the disposal of private keys or any usability concerns related to key management.

Off-the-Record Communication (OTR), outlined by Borisov et al. [9], uses short-lived keys to obtain confidentiality, perfect forward secrecy, and repudiability on instant messaging platforms. A prototype of OTR was implemented for a Linux-based instant messenger, but was not tested in a formal user study. This work outlines some of the challenges related to short-lived keys, such as the challenges inherent in securely synchronizing short-lived keys. Even though OTR is described as impractical for email,

OTR can be used in email if two individuals communicate frequently and don't expect their initial message to be encrypted. Borisov et al. explain this is possible through the use of ring signatures [28].

Topalovic et al. [41] outlined a system for short-lived HTTPS certificates designed to replace OCSP. In this system, certificates are only vulnerable to compromise for several days, because they have short lifespans. It requires web servers to fetch their new short-lived certificates periodically. This necessitates the Certificate Authority (CA) to have an online system that is more vulnerable to attacks and fraudulent activities. While short-lived certificates is a promising idea, it suffers from a complicated deployment process. This system might be helpful in developing a short-lived keys secure email prototype based on S/MIME. Implementing a prototype like this from the ground up wouldn't encounter the same deployment problems encountered with deploying short-lived certificates in the wild.

While some short-lived certificates systems are being discussed in theory, there is one system currently in place for short-lived certificates. Let's Encrypt [1] offers a free service for obtaining, reviewing, and renewing HTTPS certificates. With Let's Encrypt, HTTPS certificates are valid for 3 months before they need to be renewed. Obtaining and renewing certificates with this system is almost completely automated.

More recently, Green and Miers [18] published a paper introducing Puncturable Encryption, a novel approach to "forward secure encryption". Puncturable Encryption allows a user to update their decryption key such that it cannot decrypt messages before a certain date. An interesting property of this system is that it does not require redistribution of keys after the decryption key is updated. This system achieves forward-secure messaging and only adds low overhead in the process. While this work is compelling, it is fairly new and its cryptographic constructions have yet to receive the same security vetting that other systems, such as PGP, have undergone.

---

[1]https://letsencrypt.org/

11

2.3 Availability and Permanence of Personal Information

As time passes, more and more sources indicate Internet users are concerned about their information being accessible on the Internet. Ruoti et al. [35] conducted semi-structured interviews to understand how individuals perceive online risks. Several of the participants voiced concerns about the permanence of personal information on the Internet. One participant even stated, "nothing can be forgotten again." Participants also expressed concerns about government entities hacking into and accessing personal information stored on the Internet.

In another study by Munson et al. [25], individuals expressed concerns about modern technology making public records readily available. Work by Wang et al. [44] shows social media users are worried about unintended audiences seeing their posts, which may lead to job loss or relationship complications. Finally, Woodruff [47] conducted a "qualitative study of how users manage their reputations online." The work shows that a damaged reputation not only affects one's career, academic, and social opportunities, but may also inflict emotional and physical harm. In essence, participants in this study indicated that information shared through the Internet inherently becomes "property of the entire world."

While these works display important attitudes and coping strategies Internet users have regarding the permanence of their information on the Internet, they focus on specific forms of information or Internet mediums. Further, they do not specifically explore how worried users are about this subject.

12

Chapter 3

Short-Lived Keys Design Exploration

This chapter explores the design options for short-lived keys. These choices are either not discussed or only briefly mentioned by the current short-lived keys literature. Generally, the current short-lived keys literature only touches the surface of short-lived keys design options, because their discussions are strictly based on theoretical short-lived keys systems. The exploration of the design options discussed in this chapter was based on the goal of designing and producing a highly usable secure email prototype supporting short-lived keys. Working with this goal in mind required many design options to be discovered, recorded, compared, and discussed.

3.1   Initial Design Exploration

Initial exploration of the design space of short-lived keys started with and branched out from the short-lived keys concepts presented by Brown et al. [11, 12], as well as Schneier and Hall [37]. However, due to the lack of related work for short-lived keys in email, many of the design options were generated using previous secure email research and through brainstorming the potential needs of users based on short-lived keys in a secure email context. Some of the most important design option categories produced by this exploration are introduced in the following subsections. Each subsection contains a list of research questions relevant to the design option category discussed in the section.

3.1.1 Key Expiration vs. Key Destruction

This thesis assumes the following definitions for key expiration and key destruction:

Key Expiration: A key is considered expired when it has been marked for destruction, but still exists in some form on the user's device. The public and private portions of asymmetric key pairs both expire at the same time. An expired key may be unmarked for destruction if the owner decides it is still needed.

Key Destruction: The point at which a short-lived key has been completely forgotten, meaning there is no digital or physical record of it. The minimal requirement for an asymmetric key pair achieving destruction is when no physical or digital record of its private key exists. In this case, an asymmetric key pair is considered destroyed even when multiple copies of its public key exist in some way.

3.1.2 Key Coverage

Short-lived keys are not only defined by their expiration parameters, but also by what they protect. For example, a short-lived key A may have a lifespan of 1 week, but may only be allowed to protect 1 message. Another short-lived key B may have the same lifespan, but may be allowed to protect any number of messages sent within its lifespan. Brown et al. [12] state the "logical conclusion" of short-lived keys is one-time keys, where every encrypted message is protected by a different key. While this approach provides obvious forward secrecy benefits not provided by other approaches, this approach may introduce usability challenges not encountered in other approaches. For example, requiring frequent key exchanges may become burdensome to the user in a one-time short-lived keys scheme.

We identified and explored a set of design options related to what short-lived keys can protect (key coverage).

- *Protect N Messages:* Short-lived keys can be defined to protect N messages.

- $N = 1$: Can frequent key exchange usability challenges be overcome? If so, how?

- $N > 1$: Can N be chosen such that reasonable degrees of security and usability are obtained?

- *Protect by Logical Grouping:* Keys can protect a logical grouping of messages, such as a thread of emails, or an instant messaging conversation with one contact.

  - Which logical groupings of messages will users find to be usable?

- *Protect by Time Period:* A short-lived key can be defined to protect all messages received within the lifespan of the key. Design options for key lifespans are further discussed in Section 3.1.3.

- *Combination:* A key can protect messages using a combination of the approaches listed above. For example, a short-lived key could be defined to protect only five messages from a single email thread.

3.1.3 Key Expiration and Lifespan

Short-lived keys can be defined to expire after a certain period of time, or after specific events occur. For example, a key can be set to expire two weeks after it is first generated, or can be set to expire after a message it protects has been read for the first time. While giving a short-lived key a lifespan based on time is not required, it may be important to do so to ensure short-lived keys actually expire. For example, a short-lived key may never expire if it is defined to expire after a certain event because that event may never occur. In this case, it may be safer to define the key to expire after the event, but also expire at a specific time in case the event never occurs. Inherently, design choices made in this category may be dictated by or may dictate design choices from other categories discussed in this exploration.

15

- *Time-Based Expiration:* Short-lived keys are defined to expire on a certain date or after a certain period of time has passed.

  - *How long should a short-lived key go before expiring?*

    * 1-2 days?

    * 1-2 weeks?

    * 1 month?

    * Longer than 1 month?

    * What usability and security trade–offs are made as the expiration time grows longer or shorter?

  - *Expiration Timer Start:* When should the expiration timer start for a short-lived key?

    * When the key is generated?

    * When the public key has been shared?

    * When an encrypted message protected by the key has been opened for the first time?

- *Event-Based Expiration:* Short-lived keys expire when a specific event or set of events are fulfilled.

  - *Examples:*

    * A key expires after a user has read an encrypted message N times.

    * A key expires when a related key has expired/destroyed.

    * A key expires when a message has been received.

  - Are there any cases where event-based expiration does not need time-based expiration as a backup?

- *User Control:* What level of control should users be given over any of these options?

16

3.1.4   Automation of Key Management

In the context of short-lived keys, deciding the level of automation of key management in a secure email tool is an important topic. At first, it seems like an obvious choice to take the burden off users by providing completely automatic key management. Doing this has the potential to prevent users from feeling overwhelmed and making errors. However, complete automation of key management, while helpful, can also lead to some users to lose trust in the software. Short-lived keys introduce further usability concerns related to automated key management. For example, users may become frustrated if they can't read their encrypted email after the tool automatically deletes the expired short-lived key protecting the encrypted message. In this case, giving the user the choice between keeping or destroying their expired key could prevent confusion and frustration.

While the choice of applying complete automation to some key management tasks is more subjective, some key management tasks stand out as obvious choices for the application of complete automation. For example, any encryption keys should automatically be generated when needed. Further, any key data being passed between those participating in the secure information exchange should be passed, extracted, and stored automatically. Other obvious automation applications include, but are not limited to: verifying email DKIM signatures if needed (see Chapter 4), properly erasing short-lived keys from the devices, as well as choosing the appropriate keys for encrypting, signing, and verifying signatures.

One of the more difficult and important design choices related to the automation of key management in a short-lived keys tool is key destruction. As stated above, a short-lived key automatically destroyed by a tool has the potential to frustrate and confuse users. Finding a solution to this problem that gives users both security and usability is an important part of our exploration of this design choice. An important set of questions related to this design decision is:

17

- What level of automation, if any, should be applied to key destruction in a secure email tool using short-lived keys?

    - Should key destruction be completely automatic? This implies a secure email tool will destroy a short-lived key as soon as possible after it has expired. If the software is only run at certain times (for example, a chrome extension only running when a Chrome browser window is open), when will it be possible to destroy the keys?

    - Should key destruction be partially automatic? For example, the secure email tool prompts a user to choose between destroying or keeping a short-lived key once it has expired.

    - Should key destruction be completely manual? In this case, choosing to destroy keys is completely left to the user. They will not be reminded to destroy anything.

    - What are the usability and security trade-offs of these different options?

3.1.5   Presentation of Encrypted Messages

Secure email tools using long-term keys often have an indicator that a message is encrypted. While this is helpful in the context of a long-term keys secure email tool, more visual information may be needed to help users understand the status of their emails encrypted with short-lived keys. The following questions summarize the exploration we did related to the presentation of encrypted messages protected by short-lived keys in a secure email tool.

- Should specialized user interface elements be included to help users know of the expiration of the keys protecting their emails?

- Should a view or folder outside of the normal email inbox be created for these messages?

18

- Should expiration labels be added to views related to messages protected by short-lived keys?

  – Should the text of the label display the short-lived key's expiration date?

  – Should the label display an active countdown to the expiration of the short-lived key?

  – What color(s) should this label be assigned?

  – What color should this label be?

3.1.6 Long-Term Encrypted Information Storage

If a short-lived key is about to expire, users may want to easily, quickly, and securely store the information protected by that key elsewhere for long-term use.

- Should long-term storage of this information be left to the user?

- Should an archive feature be provided for automatic information extraction and secure storage?

- If a secure archive feature is used, how should it protect information?

  – Should they be protected by one master password, or by a unique password for each message?

  – Should a long-term key pair be used for this purpose?

3.1.7 Multiple Devices

Another potentially important aspect of designing a usable secure email tool based on short-lived keys are the challenges related to accessing encrypted email on multiple devices. If Alice initially sent an encrypted email from device A, the private key will most likely exist on device A. However, Alice will not be able read that same email on device B, because device B does not have access to the private key stored on device A. In this

19

scenario, Alice may need to be informed which of her devices can actually access the encrypted message.

- Should private short-lived keys be shared between devices?

    - What processes and cryptographic schemes can be employed to do this securely?

    - How does sharing private keys complicate completely destroying short-lived keys?

- If private keys are not shared among user devices, what is the best way to help the user understand which device they need to open the encrypted message on?

    - Can short-lived keys be bound to device descriptions, allowing a short-lived keys tool to notify the user which device they should read from?

3.1.8   Sender/Recipient Key State Synchronization

Another challenge introduced by short-lived keys is synchronizing the keys between participants in an encrypted communication session. This challenge is illustrated by the following scenario and questions. Consider Alice and Bob communicating securely, each using their own short-lived key. Alice decides she wants to end the secure communication session and destroys her short-lived key. Bob, not knowing Alice destroyed her key, sends her additional encrypted information using her public key. Alice can't decrypt the information because she has destroyed her private key.

- What design choices can we make to avoid the problems associated with the above scenario?

- Should Alice's short-lived keys secure email tool automatically inform Bob when she has deleted a key?

- Is there ever a case where the outcome of this scenario is desirable?

- What usability features need to be considered to help Bob understand what it means for Alice to destroy her key?

3.1.9 Design Exploration Discussion

A wide set of implementation choices are available within each design category. Further, the usability and security of these implementation choices may vary based on the context of the short-lived keys application and based on specific user needs. For example, some different implementation choices become apparent assuming an encrypted text message application based on short-lived keys as opposed to a secure email tool using short-lived keys.

3.2 Pilot Study

This section contains details about short-lived keys secure email design choice hypotheses we made after our design exploration. Also included are details about the methodology and results of a pilot study we conducted to gather early feedback on the design choice hypotheses.

3.2.1 Hypotheses

After exploring, recording, and comparing these short-lived keys design options, we made educated guesses on specific design choices in the context of secure email. For example, we hypothesized that users would like each short-lived key to protect a different email thread. This hypothesis was made with the assumption that email threads generally contain related information, where some pieces of information depend on or are related to other information in other thread messages. We further hypothesized that users would be interested in short-lived keys initially starting with an expiration of 2 weeks, then immediately drop to an expiration of 2 days after the first message of the secured thread is read. This hypothesis assumed that some users may not be able to access their email

21

for several days based on their life activities. Further, we assumed users would appreciate the added forward secrecy provided by dropping the expiration period to two days after the first message was opened.

Our exploration of the short-lived keys design choices also led us to develop several more hypotheses about a usable, secure short-lived keys secure email tool. For example, we hypothesized that users would want expiration labels on inbox email headers and email thread headers for messages protected by short-lived keys. We guessed that users would find it helpful to have a timer on these labels that displayed days, hours, minutes, and seconds left until expiration. These labels would actively count down the seconds left until expiration, making it clear to users that messages expire.

In regard to the automation of key destruction, we hypothesized users would want a partially automatic approach with an option to manually destroy short-lived keys as they please. This design decision would remind users to manage expired keys and would allow them to destroy keys without waiting for them to expire first.

We also made some initial hypotheses on the text to be shown on several buttons we expected to add to the user interface of a short-lived keys secure email tool. We hypothesized that potential users would understand the function of a button with the text "Make Unreadable" if the function of the button were to destroy the short-lived key protecting an email thread. Further, we hypothesized that potential users would understand that an email inbox header and an email thread header label with the text "Expired" means that the secure messages would still be accessible, but that action should be taken to make the secure messages permanently inaccessible.

3.2.2   Methodology

The pilot study involved conducting a short (7 - 8 minute), semi-structured interview with random volunteers around the Brigham Young University campus. Participants were asked a set of questions strictly related to a hypothetical short-lived keys secure

email tool. Where appropriate, participants were shown images of user interface mock-ups based on our design choice hypotheses. Instead of introducing the idea of short-lived keys to users during the interview, we strictly referred to messages/threads expiring. Further, instead of expressing the idea of the destruction of a short-lived key, users were told messages/threads could become unreadable or inaccessible.

A total of 16 participants agreed to take part in this pilot study. Several were visitors to the Brigham Young University campus, while most were either BYU students or BYU faculty/employees. The interview outline and UI mockups for the pilot study can be found in Appendix A. No personally identifying information was collected during the study.

3.2.3   Results

When asked about expiration time periods, users gave diverse answers. While some indicated they would feel more comfortable with messages expiring after a day or two, others suggested it would be best to let messages expire after 6 months or even a 1 year. Other users fell into the middle of these two extremes, suggesting 2 weeks or 1 month would be most appropriate. A common theme throughout all this feedback was that expiration timing should depend on the context of the secure communication. While some indicated expiration timing should change based on who they are communicating with, others suggested expiration timing should change based on the kind of information being shared securely.

While most of the pilot study participants in general liked the idea of an expiration label for emails that expire, about half of them did not like the idea of the expiration label actively counting down the seconds until expiration. This countdown design choice was described as "stressful" and "overkill".

On the topic of the automation of short-lived key destruction, users were asked what the secure email prototype should do once an encrypted message expires. They

23

were given the option to have the messages become unreadable automatically, partially automatically, manually, or some combination of automatic and manual action. Most of the participants expressed interest in the partially automatic approach involving a popup asking users what to do with expired messages. Many of these same participants expressed a need for a manual option to make messages unreadable so they wouldn't have to wait days for a message to expire before being given the option to make the message inaccessible. While several participants advocated for a completely manual approach, partially due to their disdain for popups and notifications, not a single participant was interested in the fully automatic approach.

When participants were asked how they would want their encrypted messages bundled in terms of protection and expiration, a majority of participants indicated they would prefer this to work on a thread level. Several participants expressed a desire for the ability to let individual email messages expire and a few even wanted to have the option for protection and expiration to work both on a thread and individual message basis.

In terms of button text, a majority of participants indicated the text "Make Unreadable" would best describe a button that makes encrypted messages permanently inaccessible. While many users expressed support for the text "Expired" on a label for a message that expired, several participants supported the use of text like "Overdue" and "Unresolved".

Finally, pilot study participants were asked about the possible situations the hypothetical short-lived keys secure email prototype could be used for. Several cited personal use situations, such as sending phone numbers, social security numbers, bank statements, etc. On the other hand, many of the participants listed professional or government contexts to use this prototype within. Some of these less personal contexts included sending confidential information within schools, business, government research labs, meeting government guidelines, and national security.

24

<center>Chapter 4</center>

<center>Prototype Design</center>

This chapter contains details regarding the design of the two secure email prototypes we implemented to test in our user study. First, we introduce the threat model we use to guide the design of our prototypes. Next, we introduce MessageGuard, the framework we use to implement these prototypes. After that, we give details on the design of the two prototypes. These details start with an introduction to PGP, the underlying cryptographic system used in our two prototypes. Next, details on the key exchange process used in both prototypes are given. Finally, information about specific design and implementation choices is provided for each of the prototypes.

4.1 Threat Model

Our threat model includes the following three entities:

1. **User:** The user's computer, operating system, Internet browser, and secure email software are considered part of the trusted computing base.

2. **Webmail Provider:** This entity is considered an honest-but-curious party [1]. The webmail provider has access to public keys, but not private keys. Further, this entity will not alter emails and other data. For example, a webmail provider under this condition will not change a public key sent by a user in an email.

---

[1] An honest-but-curious party will gather any information available to them (e.g., Gmail scans email messages), but will not attempt to break the secure email system (e.g., impersonating the user) or collude with other honest-but-curious parties.

<center>25</center>

3. **Adversary:** This entity is free to eavesdrop on, intercept, and alter any communication between users and webmail providers. For instance, an adversary under this condition may change a public key sent with an email. The adversary wins if they can use these resources to access the plaintext contents of the encrypted email body.

Directs attacks against the user, such as credential phishing, malware, viruses, etc., are outside the scope of this threat model. In addition, both an attacker that can compromise the webmail provider or a government that can legally coerce the webmail provider are also outside the scope of this threat model. We do not consider an attacker who can compromise fundamental networking primitives (i.e., TLS, DNS), because someone with this capability can already do far more damage than compromising secure email. In our threat model, the main concern is keeping the user's encrypted information secure. Thus, some information, such as email addresses and email headers, may be visible to the adversary while being transferred between webmail servers.

An adversary has two attack vectors for obtaining the encrypted information sent by a user. In the first vector, an adversary must compromise the public/private keys protecting the information and it must obtain the encrypted message itself. The encrypted message can be obtained by compromising a webmail server, or by eavesdropping on an insecure connection. User private key(s) are only stored on the user's device and may be impossible to obtain given this model's restriction on direct attacks and the definition of the user's secure computing base.

In the second option, an adversary can intercept emails containing public keys and replace the original public key with their own. The recipient of the adversary's public key might encrypt sensitive information using the adversary's public key instead of the original public key. If an adversary can obtain that encrypted message, they will be able to decrypt it. However, details about DKIM verification in Section 4.4.2 may provide a degree of verification to key exchange data reducing the severity of this problem.

Compared to the traditional PGP threat model, our threat model is rather permissive. However, Ruoti et al. [32] provide several examples of situations in which this kind of threat model is useful.

4.2 MessageGuard

MessageGuard is a platform designed by Ruoti et al. [33] enabling quick prototyping of secure email tools that tightly integrate with Gmail. The platform separates key management functionality from the user interface, allowing designers to maintain many user interface characteristics across prototypes with different encryption and key management schemes. This architecture allows for user interface confounding factors to be reduced as the usability of prototypes with different encryption systems are compared. Ruoti et al. [34] used MessageGuard to rapidly prototype, refine, and compare three secure email prototypes based on IBE, PGP, and password security models.

We used MessageGuard as a platform to develop the two secure email prototypes used in our user study. Using this platform allowed us to focus on developing the specific key management functionality and user interface features of our two prototypes, instead of investing time up front developing our own email software from scratch. Developing our prototypes with MessageGuard also provided the benefits of unencrypted greetings in encrypted emails, tools to create inline tutorials, and a framework for working with data packages included in emails sent with MessageGuard.

4.3 PGP

Both of our prototypes were designed to use Pretty Good Privacy (PGP) [16, 50] as their underlying cryptographic system. PGP is an end-to-end encryption system based on public key cryptography. This system allows users to both encrypt and sign their messages. Historically, PGP has been known to be unusable by novices users [17, 38, 46]. However, Ruoti et al. [34] recently used MessageGuard to implement a secure email

27

prototype supporting a PGP model that relies on a public key server [3] for sharing and distributing public keys. Compared to earlier studies, results from a user study with this prototype show promising usability scores and usability feedback for this prototype.

4.4   Key Exchange Model

While Ruoti et al. [34] used a public key server [3] to exchange public keys, we took another approach for key exchanges in our prototypes. For both prototypes, public keys are exchanged through email messages. An email-based key exchange was chosen over a public key server to allow for an exploration of the usability of an email-based key exchange, as well as a simple exploration of DKIM public key verification. The key exchange process implemented is based on the scenario of Bob sending encrypted information to Alice. It is diagrammed in Figure 4.4 and is described below:

1. Bob generates a public/private key pair if he doesn't have one for this exchange (see the details in Sections 4.5 and 4.6 for more details on how this works for each prototype). He generates a symmetric key K and uses it to encrypt his message. As he sends the encrypted message in an email, he attaches his public key to the email in case Alice wants to send him an encrypted message in the future.

2. After receiving Bob's email, Alice generates a public/private key pair if she doesn't have one for this exchange (see the details in Sections 4.5 and 4.6 for more details on how this is done for each prototype).

3. Alice sends her public key to Bob in an email. The contents of this email, including Alice's public key, can be encrypted for added security since Alice has Bob's public key.

4. Bob encrypts the symmetric key K using Alice's public key and sends her the encrypted symmetric key K in an email.

28

5. Alice uses her private key to decrypt the symmetric key, then uses the symmetric key to decrypt the original message.
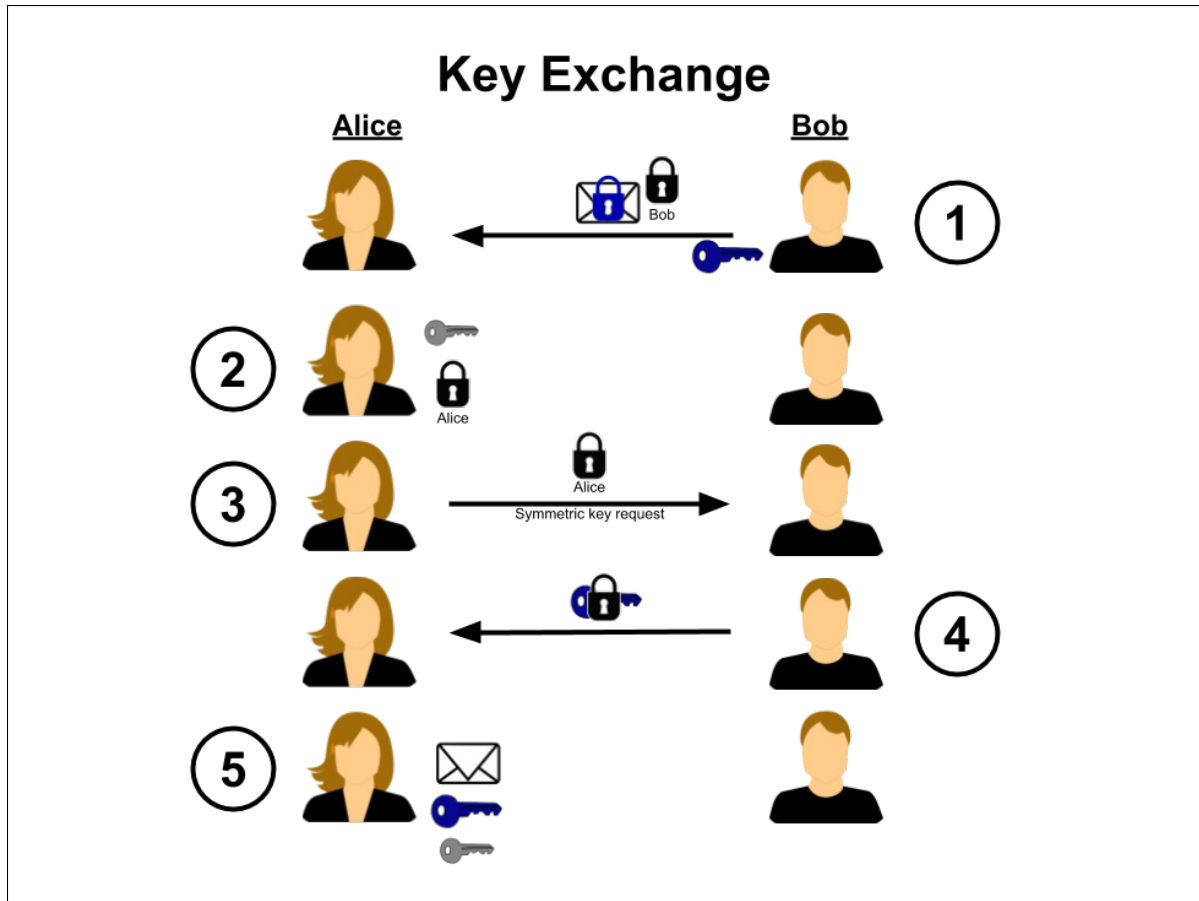
Figure 4.1: Public key exchange model used in both secure email prototypes in this study.

4.4.1 Application of Automation

In an effort to improve the usability of the key exchange and help prevent users from making mistakes, automation was strategically applied to the key exchange process. While automation is present in the five steps of the key exchange shown in Figure 4.4, several manual steps need to be taken by users to complete the key exchange. The steps below provide details on the way automation was applied to these five steps and also include information about manual action users are required to take to continue the key exchange:

Step 1

29

- *Manual Actions:* Bob turns on encryption, composes his message, then clicks "Send Encrypted".

- *Automated Actions:* In the short-lived keys prototype, a new public/private key pair is generated for Bob to be used with the newly created thread. Bob's public key is attached to the email containing the encrypted message.

   Step 2

- *Manual Actions:* Alice must open the email in Gmail after installing the prototype if necessary.

- *Automated Actions:* In the short-lived keys prototype, a new public/private key pair is automatically generated for Alice once she opens Bob's first encrypted email on this thread. Both prototypes automatically store Bob's public key for later use.

   Step 3

- *Manual Actions:* Alice must click the "Send Access Request" button on the overlay of the encrypted message. This step could have been automated, but we decided to make this a manual action to give users the choice to not continue key exchange. This gives Alice more control in the situation that she doesn't want to decrypt the email because she simply doesn't want to, or because she doesn't trust the source of the encrypted email.

- *Automated Actions:* Once Alice clicks the "Send Access Request" button, the prototype automatically sends an access request email to Bob. Among other data, this email contains Alice's public key.
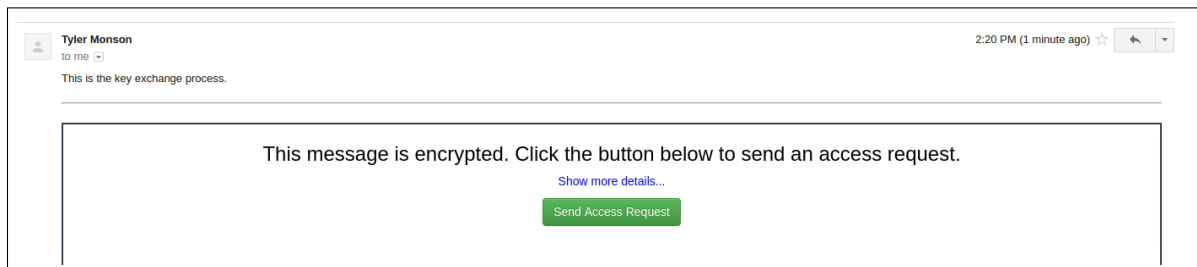
Figure 4.2: When Alice opens the encrypted message from Bob, she has the option to request access to the message. Doing so continues the key exchange.
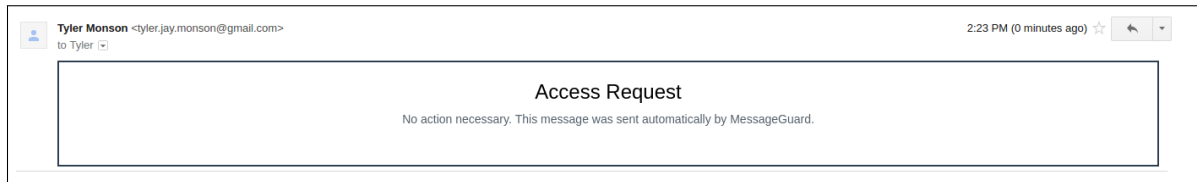
Figure 4.3: If Alice chooses to send an access request, the prototypes automatically send an access request email to Bob. Along with other data, the access request contains Alice's public key.

Step 4

- *Manual Actions:* Bob opens Alice's access request email.

- *Automated Actions:* Once the access request is opened, the prototypes recognize and parse the request. Both prototypes store Alice's public key for later use. Alice's public key is used to encrypt symmetric message key K. The encrypted symmetric key K is automatically sent to Alice in a reply email.
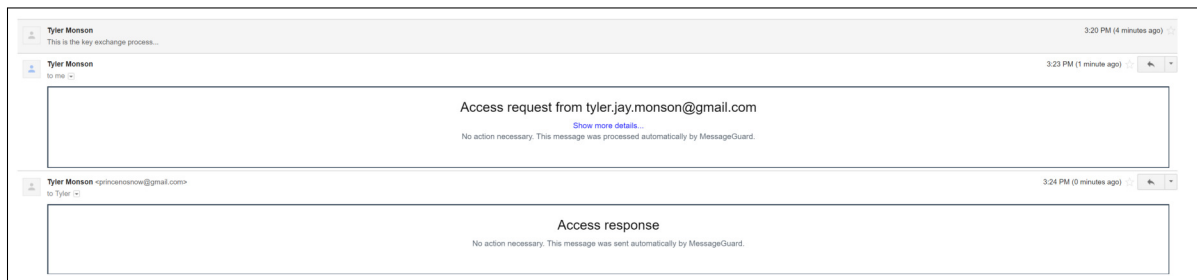
Figure 4.4: When Bob opens the encrypted thread after Alice sends an access request, the prototypes automatically respond with an access response containing symmetric message key K encrypted with Alice's public key.

Step 5

- *Manual Actions:* Alice must open the thread containing the access response that contains the encrypted symmetric key K.

31

- *Automated Actions:* Once the thread is opened, the prototypes automatically recognize the access response, extract and store the encrypted symmetric key K. Once Alice opens the original encrypted message, the prototypes automatically decrypt symmetric key K and use it to decrypt the encrypted message. If the original encrypted message is already showing, the prototype will automatically decrypt it and display the decrypted data.
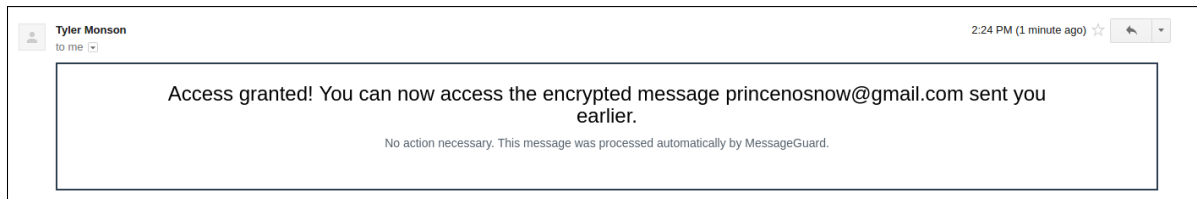
Figure 4.5: Once Alice opens the thread with Bob's access response, the prototypes extract the encrypted symmetric message key and store it.
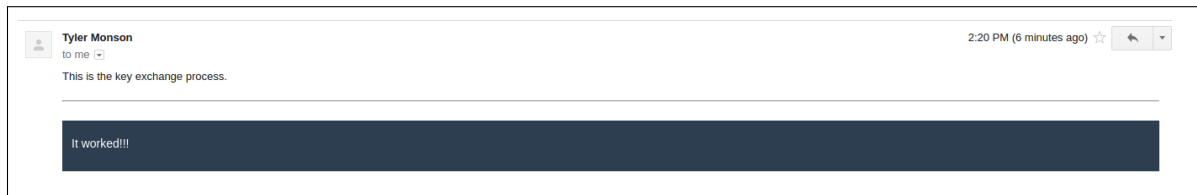
Figure 4.6: Alice can now decrypt symmetric key K and use it to decrypt the original encrypted message.

4.4.2 DKIM Verification

DomainKeys Identified Mail (DKIM) [1, 15] operates on the domain level for authentication and integrity. When Alice sends an email to Bob using Gmail, Gmail servers use a Gmail domain private key to sign the body of Alice's email. When Bob's email provider receives this email, it can request Gmail's domain public key to be used for verifying the signature. While DKIM doesn't have ubiquitous deployment [15], many widely used webmail providers, such as Gmail and Yahoo, utilize it. Although not implemented in the prototypes for this work, DKIM is a viable protocol for gaining a greater degree of

32

verification that any key exchange data, including public keys, have not been spoofed or tampered with.

To be clear, DKIM signatures do not provide verification for email headers, because the signature itself is included as an email header. Instead, a DKIM signature provides verification for the body of an email. This detail makes it clear any key exchange data requiring verification through DKIM must reside in the body of the email. For example, public keys shared as part of a key exchange through email should be part of the body of the email and should not be sent as part of any email header, even the subject header.

A future, more robust implementation of the prototypes could include features that check DKIM signatures, providing greater assurance that key exchange data has not been tampered with. While DKIM doesn't specify what the reciever should do with an email with an invalid DKIM signature, it may be important to give users a warning that the security of the email has been compromised. [15]. Thus, while this feature brings security benefits, it may also introduce usability challenges in terms of properly warning users of the implications of invalid signatures.

4.4.3   Alternate Key Exchange Scenario

A key exchange process for the situation where Alice wants to request sensitive information from Bob was not implemented in either of our prototypes. For simplicity, we assumed users interacting with this prototype could easily send an email requesting this information from their contact. In this request, users would have to provide their own information on how to find and install the prototype. Once Bob installs the prototype, he can send an encrypted message to Alice and they can complete the key exchange as seen in Figure 4.4. Another way this could work with our prototypes is Alice sending an encrypted message to Bob with her request for sensitive information either in the unencrypted greeting or in the encrypted portion of her message.

Given that the scenario of Alice requesting sensitive information from Bob may be just as likely to occur as Bob initially sending Alice encrypted information, it may be important to streamline this process in future work. For example, functionality can be added to our prototypes allowing Alice to send her public key to Bob in her plaintext email requesting the information. This hypothetical prototype would also attach instructions on how to install the prototype to Alice's initial request email. An example of this key exchange process can be seen in Figure 4.4.3.
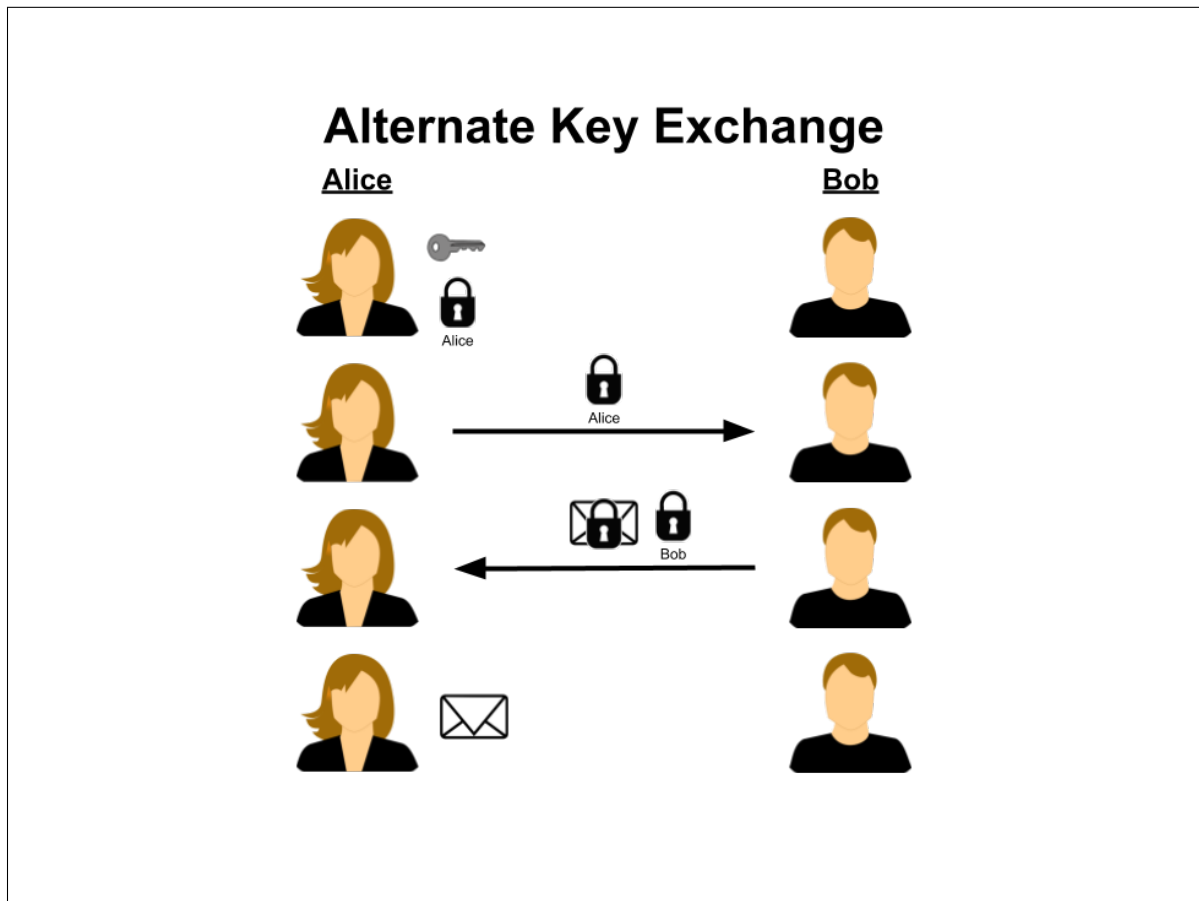
Figure 4.7: An alternate key exchange model based on Alice requesting sensitive information from Bob.

4.5 Long-Term Keys Secure Email Prototype

This prototype was designed based on more traditional long-term key management principles. This section contains details relevant to the design and implementation choices we made as we developed this prototype. Appropriate details are provided to stress the differences between this prototype and the short-lived keys prototype further discussed in Section 4.6.

4.5.1 Key Management

This prototype was designed to use a long-term key management scheme. Essentially, this means that only one public/private key pair is generated for users. It is generated during the setup phase of the tool and is used for all secure communication. For example, Alice, Jane, and Johnny will all use the same public key to encrypt symmetric keys that protect the information they send to Bob. One implication of this long-term key management approach is that users only need to go through our key exchange process described in Section 4.4 the first time they start securely communicating another person. This means that Bob will not send his long-term public key to Alice again if he has already sent it to her before. Thus, if Bob has exchanged encrypted email with Alice before, in the future, he can send her an encrypted message that she can immediately decrypt. A user's long-term key pair is stored in encrypted local storage until the prototype is uninstalled.

4.5.2 Prototype Setup

One important difference between this prototype and the short-lived keys prototype is the setup process. This prototype requires the user to enter the email address they will use with the prototype. Once their email address is entered, their long-term public/private key pair is generated and it is essentially bound to the supplied email address. The prototype is limited to only working with the email account related to the address provided by the

35

user during setup. Future work on this prototype could expand the capabilities of the prototype to work with multiple email accounts.

4.5.3 Making Encrypted Messages Inaccessible

As presented in Section 5.2.2, one of our user study tasks asks user study participants to make their encrypted messages inaccessible. We did not provide any direct functionality in the long-terms keys prototype for doing this. The limitations of our results introduced by excluding this functionality are discussed in the results and other concluding sections of this work. To make encrypted messages inaccessible with this prototype, users need to delete messages, sending them to the trash. From there, they either need to use Gmail's "Delete Forever" button, or let the message sit in the trash folder for 30 days before Gmail automatically deletes it forever.

4.5.4 User Interface

This prototype has inline tutorials similar to those seen in work by Ruoti et al. [31, 32, 34]. User interface elements, such as the green "Encrypted" labels on MessageGuard email thread headers, are nearly identical to those found in the aforementioned works.
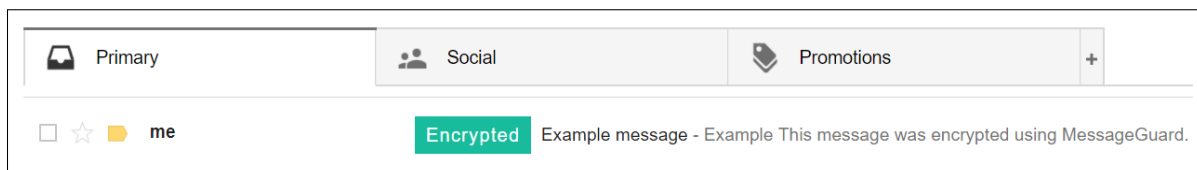
Figure 4.8: Green "Encrypted" labels are placed on the headers of encrypted message threads.

36
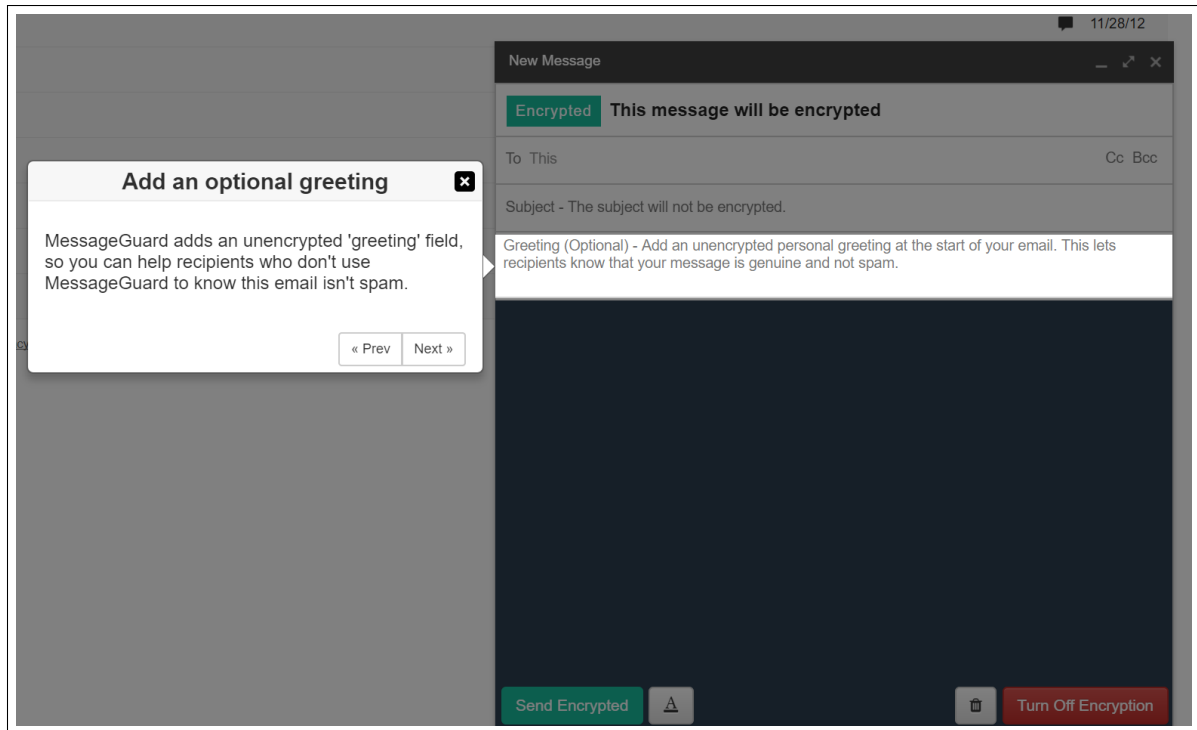
Figure 4.9: Inline tutorials for composing encrypted messages and reading encrypted messages are provided to help users learn more about the prototype.

4.6 Short-Lived Keys Secure Email Prototype

Design and implementation choices we made for our short-lived keys prototype were based on our initial short-lived keys design exploration and pilot study presented in Chapter 3. The design and implementation of this prototype was further refined throughout the design iterations discussed in Section 4.7.

4.6.1 Key Management

Key management for this prototype is based on a short-lived keys model. We applied the following characteristics to our short-lived keys model:

- **Key Generation:** Keys are generated at two different times based on the roles being played in the encrypted information exchange.

- *Initial Sender:* If Bob is initially sending an encrypted message to Alice, his short-lived key pair is automatically generated in the background while he is composing his message. Doing this, as opposed to generating the key pair as Bob clicks "Send Encrypted", is a short-lived keys usability suggestion made by Brown et al. [12]. If Bob decides to cancel composing his message, the newly generated short-lived key is immediately destroyed.

- *Initial Receiver:* As Alice for the first time opens a new encrypted thread started by Bob, a new short-lived key is automatically generated in the background and bound to that thread. Even if she doesn't request access to the thread, her newly generated key will exist until it is destroyed.

- **Key Lifespan:** Short-lived keys in this model have an initial lifespan of 30 days. After this time period, users can choose to extend the lifespan of a key for 2 days. As the key continues to expire, users can continue to extend its lifespan by 2 days if they decide against destroying it.

- **Key Coverage:** One short-lived key pair protects all encrypted messages on one email thread. If Alice has two encrypted threads open with Bob, both Alice and Bob use 2 key pairs (4 key pairs total between the two of them) to communicate with each other.

- **Automation of Key Destruction:** Key destruction in this prototype uses a combination of manual and partially automated key destruction features (see Chapter 3). Users can choose to destroy their short-lived keys at any time, but are also reminded to manage their keys through a popup once the keys have expired.

Unlike the long-terms keys prototype detailed in Section 4.5, this prototype requires a full key exchange every time two users start a new encrypted thread. This requirement is based on the short-lived key model characteristics presented above.

38

This prototype uses a popup to remind users to manage expired short-lived keys. Figure 4.10 shows an example of the popup that users encounter once one or more of their keys expire. Users are forced to choose one of the two options for each thread listed before the popup disappears. They cannot exit out of the popup by any means besides managing each of the threads presented. This popup will only show up on Gmail windows and will only appear when there exist keys that have expired, but have not been destroyed or had their lifespan extended.
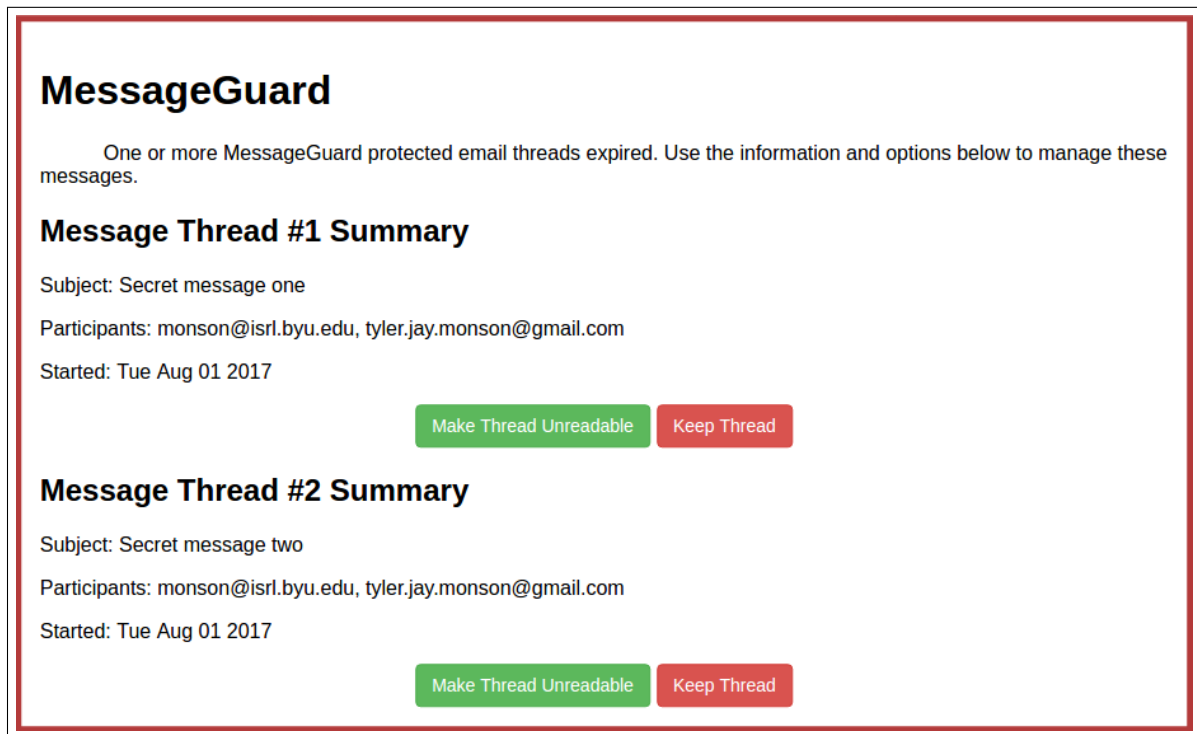
Figure 4.10: Users a reminded to manage their expired keys through a popup like this.

4.6.2 Prototype Setup

The setup for this prototype does not ask for a user's email address, because a long-term key pair is not generated during the setup phase. Instead, as seen in Section 4.6.1, short-lived keys are generated automatically in the background as they are needed. This on-the-fly key generation allows the prototype to work with multiple email accounts at the same time.

4.6.3   Making Encrypted Messages Inaccessible

With this prototype, users have multiple options in terms of making their encrypted messages inaccessible. They can still delete their messages from their email inbox and delete them forever as seen with the long-term keys prototype. However, taking this approach will not destroy their keys. If a user deleted an encrypted message in this way, the key pair protecting the message will still exist in local storage until it expires. At this point, the prototype will prompt the user to pick between destroying or retaining the key.

Another option users have for making their encrypted messages inaccessible is using this prototype's "Make Unreadable" button (see Figure 4.11), which is available at all times unless the messages are manually deleted. While this option does not delete the messages themselves, it destroys the short-lived keys protecting the encrypted messages, making it impossible to decrypt them. Users can create the same effect by using the "Make Thread Unreadable" button that shows up for every thread listed on the expired messages management popup. Once a short-lived key pair is destroyed, encrypted messages will be overlayed to show they can no longer be accessed (see Figure 4.12).
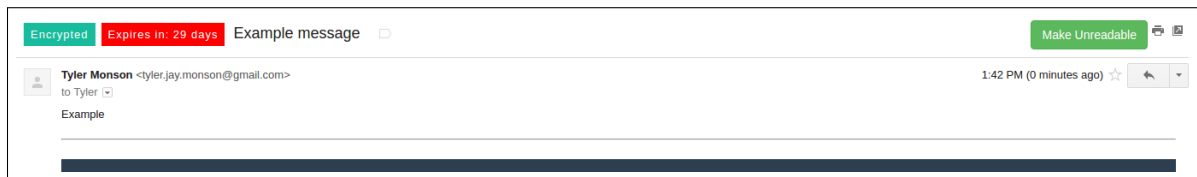
Figure 4.11: Users can use the "Make Unreadable" button to revoke their access to read their encrypted threads at any time.
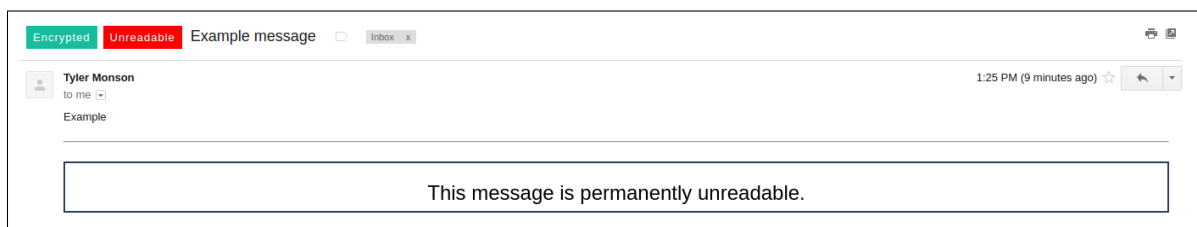
Figure 4.12: After using the "Make Unreadable" button, messages on encrypted threads are permanently unreadable.

40

4.6.4   User Interface

To help users understand their messages can expire and to help them keep track of the expiration state of their messages, red expiration countdown labels are provided next to the green "Encrypted" labels. The text of the labels shows users the amount of time left before their thread of encrypted messages expire. As time passes, these labels automatically update. They are accurate down to a minute. As seen in Figures 4.15 and 4.16, expiration label text changes from a countdown to a description as the state of the keys protecting the labeled thread change. If a short-lived key is expired, but not destroyed, the text of the related label will display "Expired". Likewise, if a short-lived key has been destroyed, the text of the related label will read "Unreadable".
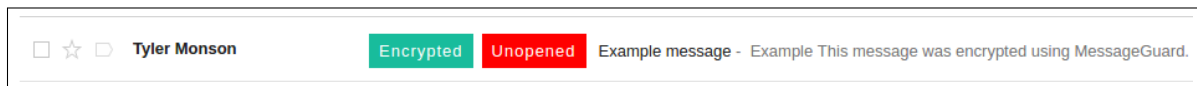
Figure 4.13: Labels for encrypted threads that haven't been opened at all will display "Unopened".
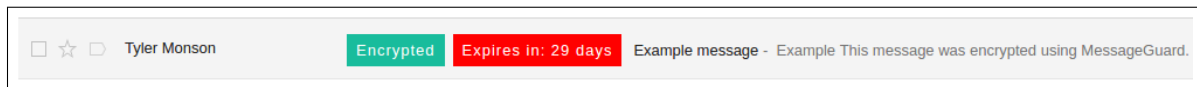
Figure 4.14: Labels for threads protected by an unexpired short-lived key show how long until the thread expires.
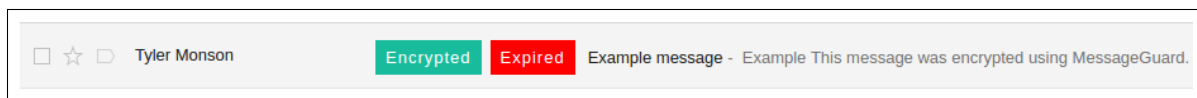
Figure 4.15: Labels for threads protected by an expired short-lived key indicate the thread has expired.
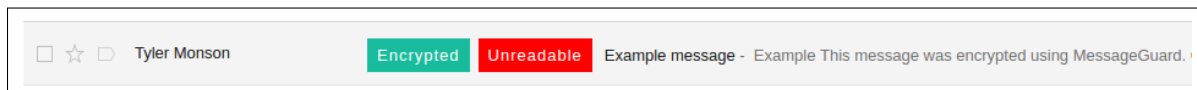
Figure 4.16: Labels for threads protected by a destroyed short-lived key show the thread is unreadable.

Inline tutorials are also provided for this prototype. While the tutorial provided for composing encrypted email remains the same as the one used in the long-term keys prototype, the read tutorial is extended for this prototype. Read tutorial text boxes

41

are provided to give users information about the read expiration labels and the "Make Unreadable" button.

4.7   Design Iterations

As these prototypes were developed, simple design iterations were conducted. These design iterations involved adding user interface elements and functionality to the prototypes, then reviewing the additions between lab members and professors. Those reviewing the additions to the prototypes did their best to put themselves in the shoes of potential users. Suggestions on changes were taken from reviewers and applied to the prototypes wherever possible. Feedback from two user study pilot studies also led to some minor changes in the software.

4.8   Model Comparison

In this section, we compare the threat model of our short-lived keys model to the short-lived keys model introduced by Schneier and Hall [37]. Their model assumes a trusted long-term key pair is established for each user that is used to sign short-lived keys. The short-lived keys can then be uploaded to an untrusted server. One drawback of this model is that it requires participants to securely establish their long-term public keys before they can start using short-lived keys to communicate securely. An advantage of their model is that it reduces the attack surface to the long-term key establishment process.

In contrast, our model depends on a trusted webmail provider to distribute the short-lived keys, which are protected during transmission using DKIM. It has a simpler startup process for new users. However, the trusted server presents a larger attack surface for tampering with the short-lived keys by hackers or through government coercion.

42

Chapter 5

Research Methodology

To evaluate the usability of our secure email prototypes, we conducted an IRB-approved within-subjects user study. We recruited pairs of participants to test each prototype together, a user study methodology that was first introduced by Ruoti et al. [31]. In prior studies using this methodology, participants report feeling more comfortable working with a friend or family member in studies with this methodology, but this approach also better simulates grassroots adoption scenarios of these prototypes because participants are not corresponding with study coordinators they do not know. In studies using this methodology, study coordinators have limited interactions with study participants; they answer questions related only to the study procedures and purposely ignore those related to using the prototypes.

To participate in this study, participants were required to have an active Gmail account because the prototypes were designed for Gmail only. Further, each participant was required bring a friend with a Gmail account. Each participant was compensated $15 USD. The user studies were approximately 50–60 minutes in duration.

To begin the study, participants were warned that the software being tested was research software and should not be used outside of the user study, and should not be used with legitimate sensitive information. Further, users were informed that they would be given information about the role they would play throughout the course of the study and were warned to use the provided fake sensitive information and not their own. Knowing users may encounter times where they would need to wait for an email and to help them

feel more comfortable and natural, they were informed they could use their phones or the Internet during these down times. Finally, users were informed they could communicate with each other the way they normally would while working through scenarios like the ones they would encounter in the study. This statement was used to preemptively remove any doubt users may have about whether they could contact their friend if they had questions or needed help.

Our user study ran from June 28, 2017 to July 12, 2017. In total, 30 participant pairs (60 total participants) engaged in our user study. Due to various technical difficulties and other problems, we rejected the results from 6 participants pairs (12 participants) from our data analysis. Details found in section 5.6 give more information on the reasons for rejecting this data. In the rest of this thesis, any participants and data from participants referred to are exclusively from the 24 accepted participant pairs (48 participants).

5.1 Participant Demographics

Participants for this user study were Gmail users recruited from the Brigham Young University campus. Most participants were recruited through posters distributed across the campus in order to attract a diverse set of participants. A small portion of the participants were recruited through email. These participants had asked to be informed about upcoming user studies after attempting to sign up for a fully-booked user study our lab was running just prior to this study.

Table 5.1 contains participant demographic information for the participants of this study. Most of the participants in this study were young (92% between the ages of 18 and 34 years old). There was almost an even split between male and female participants (54% female, 46% male). Almost all the participants had at least some college education (63% some college, 27% college or university degree, and 6% Post-Secondary Education). A majority of the participants considered themselves to have an intermediate level of

		Total	%
Gender	Male	22	46%
	Female	26	54%
	Prefer not to answer	0	0%
Age	18–24 years old	33	69%
	25–34 years old	11	23%
	35–44 years old	1	2%
	45–54 years old	2	4%
	55 years or older	1	2%
Education	Some school	0	0%
	High school graduate	2	4%
	Some college	30	63%
	College or university degree	13	27%
	Post-secondary education	3	6%
	Prefer not to answer	0	0%
Computer Expertise	Beginner	8	16%
	Intermediate	32	66%
	Advanced	8	16%

Table 5.1: Participant Demographics

computer expertise (66%), while fewer participants considered their computer expertise at beginner (16%) and advanced (16%) levels.

Participants in this study came with a wide range of occupational/educational backgrounds, including technical and non-technical fields. Thirty-six different occupations/majors were represented in this study, with almost all the represented occupations/majors having one or two participants, and only one occupation/major with four participants. Appendix B.2 has an extended participant demographics table.

5.2 User Study Tasks

After signing consent forms and after hearing the warnings and reminders in the introduction to the study, the study participants were assigned roles as participant A or participant B based on the flip of a coin. Once the study coordinators and participants moved to their respective rooms, the study coordinators introduced the participants to

45

our dual monitor setup. Coordinators instructed participants to use the left monitor for the instructions and survey questions and to use the right monitor for study tasks such as installing the extension and sending email.

For most of the study, a Qualtrics survey was used to guide participants through their assigned roles and tasks in a hypothetical email scenario. These instructions and scenarios were accompanied by data collection portions including Likert-scale questions, free response questions, and demographic questions.

This study involved having participants complete three task-based scenarios for two different secure emails prototypes: a long-term keys prototype (LTK) and a short-lived keys prototype (SLK). Participants were asked to role-play the same scenarios and work through the same tasks for each prototype. Each participant pair of the study tested the prototypes in a randomized order to account for the effects of test-enhanced learning.

In many tasks, participants were given worksheets providing information helping them complete their tasks. Some of these worksheets contained blank spaces allowing users to fill in the sensitive information acquired during the task. While some of these worksheets were still necessary, due to our dual monitor setup, we quickly realized it was not necessary to provide worksheets with blank spaces for the sensitive information. Instead of copying the sensitive information to the worksheet, participants usually just copied the info from their open email on the right screen to the survey questions on the left screen.

In all the following study tasks, participants role-played the passing of time in the scenarios they were working through by flipping the pages of a desk calendar to the appropriate date given in the scenario. Requiring the participants to physically change the date on the desk calendars was done to create a break in the normal study flow and emphasize the passing of time in the role-played scenarios. The following tasks and scenarios are presented in the same order participants were asked to complete them in.

46

5.2.1 Initial Scenario

In the initial scenario of the study, participant A and participant B are asked to play different roles. Participant B plays the role of participant A's friend who is an accountant working on participant A's taxes. In the scenario, participant B has requested participant A's social security number (SSN) and personally identifying number (PIN) from last year's taxes. Participant A is required to send the SSN and PIN to participant B securely using encryption. Participant A is provided a URL they can follow to obtain the secure email prototype for this purpose. Participant B is not informed that participant A is required to encrypt the SSN and PIN.

The tasks for this scenario require:

1. Participant A to download and install the prototype using the site at the provided URL.

2. Participant A to send the provided SSN and PIN in an encrypted email to participant B.

3. Participant B to open the encrypted email and follow the instructions to install the prototype.

4. Participant B to retrieve the encrypted SSN and PIN.

5. Participant B to send the provided tax confirmation code and tax confirmation PIN in an encrypted email to participant B.

6. Participant A to open and retrieve the confirmation code and confirmation PIN.

7. Participant A to send a final message to participant B saying they received the confirmation information. Study instructions did not require this final message to be encrypted.

8. Participant B to open and read this final message.

9. Each participant to enter the sensitive information they received into their survey.

47

After these tasks were completed, study coordinators took control of the study computers for a short time to prepare for the next task. If participants were in the midst of testing LTK the study coordinator would simply close the participant's Gmail tab, explaining that they were about to simulate the passing of 31 days and needed to close the tab to help the participant simulate coming back to Gmail after that time period. On the other hand, if the participant was in the midst of testing SLK, the study coordinator would not only close the open Gmail tab, but would also use the options of the Chrome extension to expire the short-lived keys of the prototype. This was done with the simple click of one button on the extension's options page. Participants witnessed these steps taken by study coordinators.

5.2.2 Retrieval Scenario

Study instructions for this scenario continued the roles that participants were given in the initial scenario. It begin with role-playing the passing of 31 days after the initial task ended. This length of time was chosen because participants' encrypted messages from the initial tasks for SLK expired after 30 days. After role-playing this passage of time, participants were given the scenario of needing to go back to their email to retrieve the SSN and PIN (participant B) or the confirmation code and confirmation PIN (participant A). Both participants were informed that they had disposed of hard copies they had of this information because they thought they didn't need it anymore. Further, in both scenarios, the motivation for retrieving the information again was based on the United States Internal Revenue Service (IRS) needing more information from each participant.

Participants were required to retrieve this information from their email accounts and not to request it directly from the other participant again. The scenario ended with participants entering the retrieved information into their surveys. If participants could not retrieve the information, the study coordinator would provide the necessary information so that the participant could continue the study.

5.2.3 Removal Scenario

For this scenario, participants were informed that they were 100% certain they would not need the information again. Considering this, they were tasked with doing what they thought necessary to make the information inaccessible. No passing of time was role-played for this scenario, as participants were instructed they wanted to remove emails containing the sensitive information immediately after retrieving it.

5.2.4 Snooper Scenario

The final scenario began with participants role-playing the passage of one day since they completed the removal scenario. At this point, participants role-played a scenario where they left their computers and email open for several minutes and also lost sight of their computers for several minutes. Each scenario involves a suspected snooper having access to the participant's computer. For example, participant A role-played leaving their computer for several minutes to get some fresh air outside and saw their roommate sneaking out of the participant's room as the participant came back inside. Participant B role-played leaving their cubicle to get a snack at a vending machine and coming back to see their coworker casually walking out of their cubicle.

After reading these scenarios, participants were tasked with determining the encrypted information the snooper could have seen if the snooper was on the participant's computer. For this task, participants were encouraged to look through their email inbox to help determine what the snooper could have seen. This task ended with the participants using the survey to answer the question, "Was your roommate[co-worker] able to read any of your encrypted email?".

5.2.5 Prototype Scoring and Free Response

Once the scenarios and related tasks were complete, participants were asked questions related to the prototype they tested. First, they completed 10 Likert-scale questions

49

regarding the usability of the prototype. Next, they were asked to give a free response on what they liked about using the prototype. Finally, they answered a free response question asking what they would change about the prototype.

5.3   Final Survey Questions

After participants completed all scenarios, tasks, and survey questions for both prototypes, they were given a final set of survey questions. The first question from this set asked participants to choose their favorite of the two prototypes, also allowing for participants to say they didn't like either of the prototypes. After this, they were asked to explain why they chose the prototype they did. Finally, they were asked to give two Likert-scale responses to the following prompts:

1. I want to be able to encrypt my email.

2. I would encrypt my email frequently.

5.4   Browser Cleanup

Between the final survey and the exit interview, the study coordinators wiped the Chrome browser history for safety and consistency—to protect participant accounts and reset the browser for the next test.

5.5   Exit Interview

At the end of the study, each coordinator conducted a 10–15 minute semi-structured exit interview with the participant they monitored. The interview questions, enumerated in Appendix B.1.9, first explored the participant's experience with the study and the prototypes. Some questions encouraged the participants to make comparisons between the two prototypes, while other questions focused on ideas related to short-lived keys and message/information permanence. Participants were also asked if they would like to use

either of the two prototypes in the future. If enough time was left, study coordinators asked participants questions related to SLK's user interface elements and other design choices. Study coordinators consistently asked the participants for more details if their answers were unclear or did not contain enough explanation.

5.6 Quality Control

A total of 6 participant pairs were removed from the results of this study. Even though problems generally only occurred for one participant during these studies, data from both participants was removed. Technical issues with the prototypes ruled out 4 participant pairs. For two of those pairs, one of the prototypes they were testing became completely inoperable for unclear reasons during the first exchange of sensitive information participants attempted. These technical failures could not be resolved by trying the task again from scratch or through study coordinator debugging efforts. With two other participant pairs, technical failures prevented SLK user interface elements from appearing on screen. This restricted some of the participants' ability to complete tasks and make reasonable comparisons between the two prototypes tested.

The data from another participant pair was thrown out because it was discovered that one of the participants from this pair was 17 years old, whereas we only intended to recruit those 18 years old and older. Another pair's data was removed from the results of this study because mistakes made by the study coordinators affected the participant's ability to complete some of the tasks. Any participants sent away early were still compensated $15 USD for their time.

5.7 Study Machines and Key Generation

Efficient key generation time is especially important in SLK, because the keys are generated close to the time they are needed for the key exchange. Key generation times are likely to differ based on hardware configuration. The hardware configuration shown below was used

51

for both of our user study machines that participants used. This hardware configuration generates keys without noticeable delays (generation time < 1 second) in both prototypes.

- **CPU:** AMD Ryzen 7 1700 (8 cores, 16 threads, 3.0 GHz)

- **RAM:** 16 GB DDR4

- **Graphics:** NVIDIA GEFORCE GTX 1050 Ti

- **Disk:** Samsung 960 EVO M.2 NVMe SSD

5.8 User Study Pilot

We conducted two pilot studies using pairs of participants from our lab. Feedback from these pilot studies were used to further refine the prototypes, as well as our user study design. These pilot studies also helped to train study coordinators for the upcoming study.

5.9 Limitations

As stated above, this study involved role-playing the passage of time in several instances. Although we took steps necessary to help participants better internalize the simulation of time passing, the scenarios and tasks in this study would be more realistically contextualized in a longitudinal study. A longitudinal study would not only allow short-lived keys to expire naturally, but it would also allow participants to more naturally experience how the prototypes integrate with their email accounts over time.

Requiring users to send, receive, and manage encrypted emails with only one other person is another limitation of this methodology. Using these prototypes to securely communicate with more than one other person is likely to be more complicated and may introduce more usability complexities, as well as more opportunities for users to make mistakes. In the end, we choose to create tasks and scenarios around securely

communicating with only one person, because it allowed us to focus on the basic usability of the prototypes and the general participant interest in these prototypes.

Our study also has limitations common to all existing secure email studies. First, our population is not representative of all groups, and future research could broaden the population (e.g., non-students, non-Gmail users). Second, our study is a lab study and has limitations common to all studies run in a trusted environment [23, 31, 39].

Chapter 6

Quantitative Results

This chapter reports the quantitative results from our user study. We first present the SUS scores participants gave for each prototype. SUS results are followed by an analysis of the data showing differences based on the order the prototypes were tested. Following that, quantitative data for participants' favorite prototypes and the mistakes participants made are presented. This chapter ends with a discussion on the limitations of our quantitative results. We will refer to the long-term keys prototype as LTK and the short-lived keys prototype as SLK.

The data for this study can be downloaded at isrl.byu.edu.

6.1 System Usability Scale

The System Usability Scale (SUS) is a standard usability metric. It is based on ten usability Likert-scale questions and a method for computing a single score between 0 and 100. The questions and methodology for calculating SUS scores can be found in Appendix B.3. SUS has an established track record in the usability community and there is evidence that it is reliable across different sets of participants [29]. It has been used in hundreds of usability studies [5] and the original SUS paper [10] has been cited over 4,900 times as of August 2017. After comparing SUS to four other usability metrics, Tullis and Stetson determined SUS gives the most reliable results [42].

Work done by several researchers helps give greater context to SUS scores. Bangor et al. [5] analyzed 2,324 SUS surveys, and derived a set of acceptability ranges that

describe whether a system with a given score is acceptable to users in terms of usability. Bangor et al. [5] also associated specific SUS scores with adjective descriptions of the system's usability. Sauro et al. [36] also analyzed SUS scores from Bangor et al. [4], Tullis et al. [42], and their own data. They calculated the percentile values for SUS scores and assigned letter grades based on percentile ranges. The contextual clues are shown in Figure 6.1.
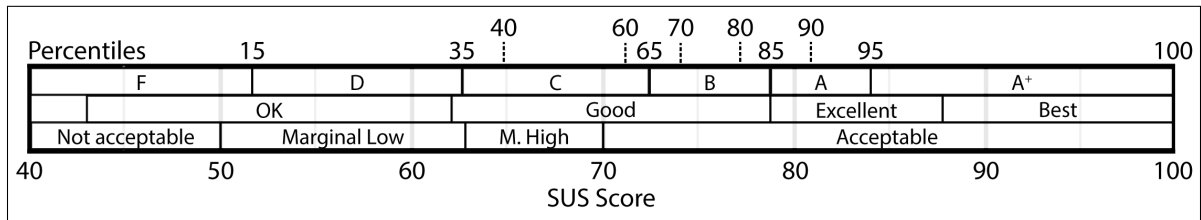
Figure 6.1: Adjective-based ratings and percentiles to help interpret SUS scores. SUS scores are given across the bottom of the figure. Bangor et al. [5] developed the acceptability ranges seen on the bottom bar, as well as the adjectives (OK, Good, etc.) on the middle bar. Sauro et al. [36] developed the letter grades seen on the top bar and the percentile ranges seen across the top of the figure.

The System Usability Scale was used to evaluate the usability of the two prototypes. A breakdown of the scores given to each prototype can be seen in Table 6.1. Overall, SLK received a mean SUS score of 73.3, while LTK received a mean SUS score of 66.8. According to the contextual scales, both prototypes are rated as having "Good" usability. While the difference in SLK's SUS score and LTK's SUS score is statistically significant (two tailed student t-test, equal variance, $p < 0.005$), we argue that this is not a strong indication of SLK being more usable than LTK. Limitations introduced in Section 6.2 and further discussed in Section 6.5 provide supporting results and possible explanations for this argument.

Ruoti et al. [31] introduced paired participant studies, but never explored whether the participants playing different roles in the study were having different experiences with the tools, showing the experience of different participant roles contributes research value. To confirm we gained real research value from our paired-participant study, we calculated the correlation between the SUS scores participants gave for each prototype

	Participant	Count	Mean	Standard Deviation	Min	Q1	Median	Q3	Max
LTK	A	24	66.5	19.9	22.5	56.9	68.8	80	100
LTK	B	24	67.2	16.0	27.5	65	70.0	77.5	92.5
LTK	Both	48	66.8	17.9	22.5	61.3	70.0	77.5	100
SLK	A	24	72.3	15.3	32.5	65	72.5	82.5	95.0
SLK	B	24	74.4	7.8	45.0	72.5	75.0	77.5	90.0.
SLK	Both	48	73.3	12.1	32.5	70.0	75.0	78.1	95.0

Table 6.1: SUS Scores

against the SUS scores participant B gave for each prototype. Figure 6.2 displays the two linear regressions for the scores of both participants based on prototype. There was little correlation with these scores (Pearson product-moment correlation coefficients [1]: LTK — 0.188, SLK — -0.119), suggesting that participants playing role A are having different experiences testing the usability of these prototypes than the participants playing role B are. Participants having different experiences with the prototypes during the study is evidence that we gained real value from using pairs of participants in our study.

---

[1]The Pearson product-moment correlation coefficient (bivariate correlation) measures linear correlation between two variables. The coefficient values range between -1 and +1. A coefficient of 0 represents no correlation, while coefficient values of -1 and +1 represent total negative and total positive correlation respectively.
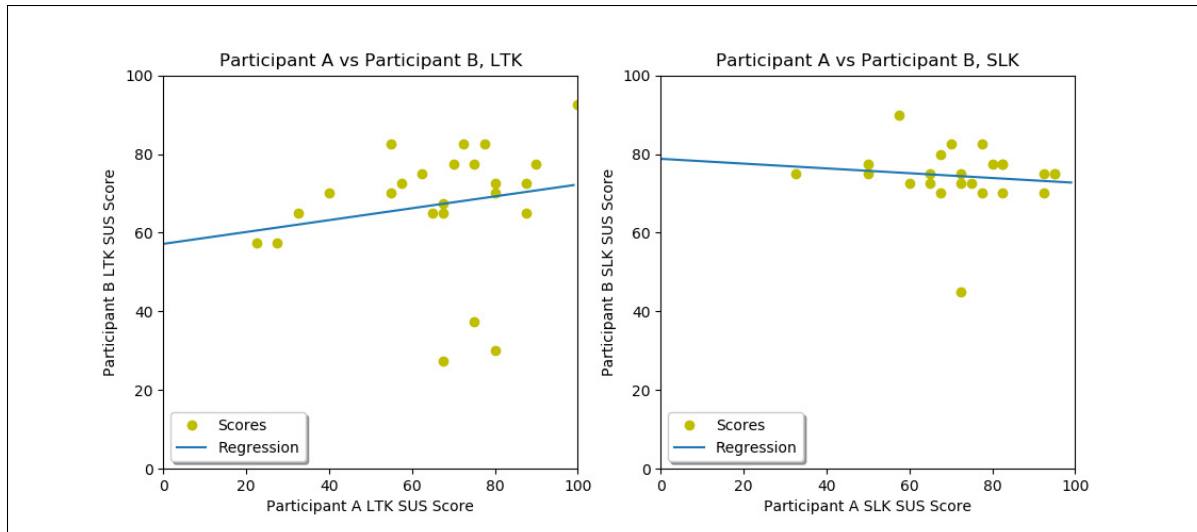
Figure 6.2: Linear regressions for the correlation of scores between participants for prototypes.

While there was little correlation between participants A and B in terms of SUS score, there was a moderate correlation between how participants each rated LTK and SLK (Perason product-moment correlation coefficient — 0.624). Figure 6.3 displays two linear regressions for these scores (the second is the inverse of the first). The moderate correlation between these scores suggests that participants were fairly consistent in their ratings of the two prototypes. So, if participants gave a high score to the first prototype they tested, they fairly consistently gave the second prototype they tested a high score as well.

6.2 Differences Based on Test Ordering

We analyzed our SUS data to determine if the test order of our prototypes affected the scores participants gave them. The mean SUS scores for each prototype based on their test order are provided in Table 6.2. These mean values show interesting differences between the mean scores of the prototypes based on their test order. For example, LTK has a statistically significantly higher SUS score when it is tested first as compared to its SUS score when it is tested second (two tailed student t-test, equal variance, p <
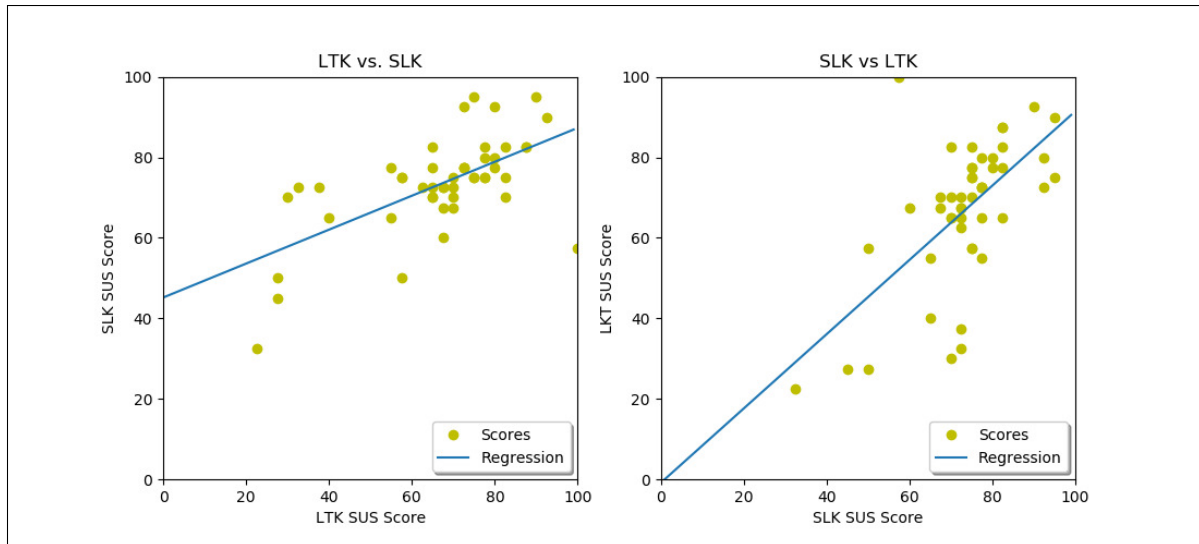
Figure 6.3: Linear regressions for the correlation of scores between prototypes given by participants.

	LTK First	LTK Second	SLK First	SLK Second
SUS Mean	74.06	59.58	70.31	76.35

Table 6.2: Mean SUS scores for prototypes based on their test order.

0.05). On the other hand, even though SLK has a higher mean SUS score when it is tested second, the difference is not statistically significant (two tailed student t-test, equal variance, $p = 0.065$).

There are several possible explanations for the statistically significant difference in SUS scores for LTK. First, given that users were asked to make their encrypted messages inaccessible, participant's usability scores for LTK may be affected by the participant's exposure, or lack of exposure, to other methods of making their messages inaccessible. For example, when LTK was tested second, participants had already been exposed to SLK's "Make Unreadable" functionality. Having an obvious, easy-to-use option for making messages inaccessible in the first prototype and having no "Make Unreadable" functionality in the second prototype is likely to have been a sharp contrast in usability from the perspective of the participants that tested the prototypes in this order. Shedding more light on this, we observed several participants give up on making their messages

	Participant A	Participant B	Total
LTK	6 (25%)	6 (25%)	12 (25%)
SLK	17 (71%)	18 (75%)	35 (73%)
Disliked Both	1 (4%)	0 (0%)	1 (2%)

Table 6.3: Participants' favorite prototypes.

inaccessible (meaning they took no action at all) when they tested LTK second. In contrast to this, all but one of the participants that tested LTK first took some kind of action to delete their encrypted messages.

6.3   Favorite Prototype

After completing all tasks for both prototypes, participants were asked to choose which of the two prototypes was their favorite. The results are summarized in Table 6.3. SLK received more support from participants with 35 (73%) participants choosing it as their favorite. LTK received 12 (25%) favorite prototype votes and one participant indicated they didn't like either of the prototypes.

Interestingly, we see very little difference in the number of A and B participants that chose LTK or SLK as their favorite prototype. At first, this seems to suggest that although participants A and B are having different usability experiences with the prototypes (see Section 6.1), pairs of participants are coming to similar conclusions about their favorite prototypes. However, only 15 out of the 24 participant pairs (63%) agreed on their favorite prototypes. This may be even more evidence that participants A and B are having different experiences and that paired-participant studies are providing real research value.

The proportion of participants that chose SLK as their favorite prototype was statistically significant (Test for one proportion, null hypothesis 50%, observed proportion 74.46%, population 47, p < 0.001). We are 95% confident the proportion of users from

the represented population that will choose SLK as their favorite prototype lies between 59% and 84% ("exact" Clopper-Pearson confidence interval).

While these data points show most of our participants chose SLK as their favorite prototype, this data does not explain why these participants chose SLK as their favorite. Participants may have chosen SLK as their favorite prototype for a variety of reasons. This problem is further explored in the Limitations section (Section 6.5).

6.4   Mistakes

After reviewing the results of the study, we defined two mistakes participants could make as they worked through their study tasks:

1. **Failure to Make Messages Inaccessible:** Participants could make this mistake by not taking the necessary actions to make all their sensitive encrypted information inaccessible. For example, this mistake can be made by deleting sensitive information, but not deleting it from Gmail's trash. Further, this mistake can be made by not fully deleting every message on an encrypted thread. Each participant has the chance to make 2 mistakes for this category, one for each prototype.

2. **Incorrect Snooper Question Answer:** After participants were asked to make their sensitive encrypted messages inaccessible, they were given a scenario where they suspected someone snooping on their computer. They were tasked with determining if the snooper could have seen any of their encrypted email. It was counted a mistake if participants answered this question incorrectly. For example, answering that the snooper could not see any encrypted messages, even though at least one of the encrypted messages wasn't deleted, is counted as a mistake. We counted answers of "I'm not sure" as a mistake, because participants should have no doubt whether or not their sensitive information is inaccessible. Each user has the chance to make 2 mistakes for this category, one for each prototype.

| | | Mistakes | |
		Making Messages Inaccessible	Answering Snooper Question
Participant A	LTK	11	5
	SLK	2	1
Participant B	LTK	9	6
	SLK	3	3
	Totals	25	15

Table 6.4: A summary of mistakes made by participants in our user study.

The mistakes made by participants in our user study are summarized in Table 6.4. Participants made more mistakes using LTK than SLK. Further, more mistakes were made while users were making their messages inaccessible compared to the number of mistakes made while answering the snooper question. This may suggest that while many participants couldn't successfully make their messages inaccessible, given a snooper scenario, a majority of them were still able to successfully identify when their messages were still accessible. Encountering the snooper scenario may have made participants work harder to determine if their messages were actually inaccessible. The limitations of these quantitative results are also further discussed in Section 6.5.

6.5 Limitations

Even though SLK was given a higher SUS score, was chosen as a favorite prototype more than LTK, and experienced less mistakes by participants, our study was conducted in such a way that the quantitative data cannot inform us on what factors led to these results. Asking users to make their messages inaccessible may have given an unfair advantage to SLK, because SLK had functionality directly related to this task, while LTK required users to figure out how to do this without any help from the prototype. Further, the quantitative data collected in this study does not help us understand whether users liked SLK's short-lived keys more than LTK's long-term keys. Participants may have chosen SLK as their favorite and may have given SLK higher SUS scores based on either, or a

61

combination of these factors. While quantitative data cannot shed light on these issues, an analysis of our qualitative data provides more clarity.

Chapter 7

Qualitative Results

In this chapter, we present a discussion of our two secure email prototypes based on an analysis of the qualitative data from the study. Qualitative data was gathered through free response questions in the study survey, as well as a 10–15 minute exit interview conducted for each participant. The questions participants were asked in the survey and the interview are available in Appendix B. In the interviews, participants referred to the LTK prototype as prototype A, or just A. Similarly, participants referred to the SLK prototype as prototype B, or just B.

Given that we used a semi-structured interview approach for the exit interviews and because some interviews were shorter than others, not every interview question was asked to every participant. Due to this, we simply report the number of participants that expressed specific points without giving the proportion of participants that expressed specific points. Further, some participants expressed conflicting ideas. For example, when asked which prototype was more secure, one participant (P8A) stated they felt LTK was more secure, but also said they felt both prototypes were equally secure.

7.1 Interview Coding Methodology

We analyzed the qualitative data from the exit interviews through a simple coding process. Out of our 48 participants, audio for 47 exit interviews was available for transcription and analysis. Audio for one of the interviews was not captured because the audio recorder's batteries died before the interview started. In this case, the study coordinator took

notes throughout the interview. We used these notes during the coding process for this participant's data.

For convenience and searchability, the interviews were first transcribed. After transcription, two researchers worked together to code the interviews at the same time. The two coders listened to the exit interview audio, following along with the dialogue on the related transcript. As they listened, the coders jointly extracted and recorded codes and interesting participant quotes. If ever the coders disagreed on a code, they would discuss the disagreement, referring to the transcription and audio until they agreed.

7.2 Security and Trust

During the exit interview, participants were asked which of the two prototypes they felt was more secure and which of the two prototypes they trusted more. Overall, 37 participants indicated they felt SLK was more secure, 3 felt LTK was more secure, and 8 felt that both prototypes were secure in general. Of those that felt SLK was more secure, 26 participants expressed that the functionality of the "Make Unreadable" button made SLK feel more secure. Also, 15 participants commented that the expiration of messages in SLK also made SLK feel more secure. Commenting on the reasons they felt SLK was more secure, P19B and P23B respectively said:

> "By hitting that button to make it unreadable, to me gave me more confidence that even though the message was still in the trash, the information within the message was... gone."

> "I liked the extra buttons on it. That I could see where to make it secure so nobody else could read. So in the scenario, somebody going into my cubicle, I could say with confidence I had secured it. I knew it was. On the second one (tool A), I couldn't remember how to secure it."

64

Code	#
SLK more secure	37
"Make Unreadable" button made SLK more secure	26
Expiration made SLK more secure	15
Both prototypes secure in general	8
LTK more secure	3
It was more developed or had more options	3
I felt more comfortable with the prototype	2
It was more clear	2
Access requests made it feel more secure (key exchange)	1

Table 7.1: Codes related to the security of the prototypes.

The responses participants gave for saying one prototype was more secure than the other are summarized in the codes seen in Table 7.1.

When asked about which prototype they trusted more, participants generally responded in similar ways as they did when asked about security. Overall, 34 participants trusted SLK more, 5 trusted prototype A more, and 7 trusted both about the same. Participants expressed that they trusted SLK more because of its make unreadable functionality (16) and because of its expiration functionality (10). Interestingly, two participants expressed they trusted prototype A over SLK, because they did not understand SLK's "Make Unreadable" button. On this subject, P2A and P22A respectively said:

"I guess the wording on the button, it also makes me concerned about other buttons and I could be misinterpreting those maybe."

"I think I trust A more, just because with that make unreadable, I didn't understand what I had done. So, if I were like in this situation where I couldn't even access it anymore and he also like revoked his access too... We're done... It's done... There's nothing we can do. So, I think I trust A more, just because I myself am clumsy, or like... I didn't process or didn't understand this would make it unreadable to me too."

Participants P17B and P22B both expressed trusting SLK more than LTK. When asked about why they felt this way, they respectively responded, saying:

Code	#
Trust SLK more	34
Trust SLK because of "Make Unreadable" button	16
Trust SLK because of expiration	10
Trust both prototypes in general	7
Trust LTK more	5
It gave me more control	5
It felt more secure	3
More choices provide more safety	2
Did not understand "Make Unreadable" button	2

Table 7.2: Codes related the degree of trust in the prototypes.

"I feel like being confident in how to use the software makes me more confident that it works. I know that's a logical fallacy, but, I do admit that being able to know how to use it makes me again more confident that it's effective in what it's supposed to do."

"I guess B just because I liked how B really like... thought about what the user wants. I mean you know, I think it's cool that B allows you to, like it does it for you in the sense that after 30 days it's expired. However, if you want to keep it, the messages, then you can, you know, keep the thread and then after that make it unreadable if you don't need it again. So, I think B... gave you more options and so it seemed more..."

The responses participants gave for saying they trusted one prototype over another are summarized in the codes seen in Table 7.2.

7.3 SLK Features

Participants were asked several questions about SLK's features throughout the course of the exit interview. These questions focused on user interface elements, functionality, and key management approaches for SLK. The qualitative feedback participants gave after being asked these questions are summarized in this section.

7.3.1 Expiration Timing

We asked participants if they thought SLK's default expiration lifespan of 30 days was a good default value for expiration timing. A little over half of the participants (26) indicated that a month is good for expiration timing in general. However, some participants, even those that thought one month was good, expressed a need for making the expiration time shorter or longer. More participants suggested making the expiration time shorter (20) compared to the number of participants that indicated they wanted it to be longer (5). Four participants expressed the expiration timing for secure emails should only be long enough to let you save or write the sensitive information somewhere else. Thirty-one participants indicated the expiration timing of some secure emails depends on context. For example, participants explained expiration time should be dependent not only on what sensitive information is being sent, but should also be based on who you are sending the information to.

Most participants mentioned a desire to override the default expiration time before being asked. Overall, 45 participants expressed interest in having control over expiration timing for their secure emails. On this subject, P17B and P22A respectively stated:

> *"Personally, I would like to be able to set it every time. Yeah, it's an extra step, but it's something that I think is important enough that I would like to be able to say, okay I want to keep this information for three days. After three days I'm not going to really need it anymore… Or, I'd like to be able to keep it for 30 days. And I feel like there should even maybe be a cap maybe like no longer than 3 months… After 3 months then you should just request the information again."*

> *"I think it would be really cool though if you could customize the amount of days you would want it to be set. That would be really cool, cuz that way*

like the sender can make sure that, okay you can only have access to this
information in this span of time, because this work needs to be done."

Even though many participants expressed a desire to control the expiration time, fifteen participants expressed a desire for a default setting so they would not have to set an expiration time for every secure email they send. For example, P10B said:

"I think it's nice to have a default. But then if I have something that I want
to last like shorter or longer, it would be nice to be able to like have some sort
of capability to go in the settings."

This feedback indicates that a short-lived keys tool should support a default expiration time, but should also give users the option to revise the expiration timing on a per-message basis. Since the results on an appropriate default time are mixed, users could select the default time during the setup phase of the tool.

7.3.2   Expiration Labels

Participants were also asked what they thought about the red expiration labels shown next to inbox and thread secure email headers in SLK. In response to this question, 22 participants expressed that these labels stood out, 11 said they were helpful, and 4 stated the labels were easy to understand. Expressing these ideas, P3A and P19B respectively stated:

"It was very easy to see, very easy to notice. I didn't have to rummage around
to find that it had that function, that feature. Like I understood what it meant
as soon as I saw it."

"I think they're helpful. I mean that was the first thing that kind of drew
[me in] when I started the second one [SLK] was the fact that it looked like
it expired and I thought that was kinda cool as opposed to the first one that
didn't have that."

Thirteen participants explained that these labels are good because they provide good reminders to users and help prevent users from being surprised or frustrated when secure messages become unreadable. For example, P10B said:

"It's a nice reminder, like, 'Oh yeah, this expires in this many hours or days.' So, it's like a nice reminder to, 'Kay, look, have I done everything I need with the information?' Can I actually like delete it delete it?"

Participants were also asked if they thought these kinds indicators should exist in any tool that expires messages. Out of the participants that were asked this question, 21 stated these kinds of indicators should exist in tools that expire messages.

While many participants expressed support for these labels, some did not. Some participants were indifferent to the labels and others expressed negative sentiments about them. Expressing concerns about labels showing attackers exactly where to look for sensitive information, P6A stated:

"I also think if someone new was to jump onto my computer and think, 'What's sensitive here?' That's exactly where their eyes would go to."

Other opinions on the expiration labels include: one participant stating the red on these labels was scary, two participants explaining the labels were annoying/nagging, and another participant saying the red on the labels was distracting.

7.3.3 Automation and Making Messages Unreadable

As a reminder, SLK does not automatically make secure emails unreadable. Users of SLK have the option to manually make messages unreadable whenever they want using the "Make Unreadable" button. Or, they can wait until the email expires and manage the expired email through a popup that gives them the choice of retaining the email or making it unreadable. When asked about the way the prototype handled expired encrypted emails

in general, 27 participants indicated they liked the approach. Even though this approach received positive feedback, participants had other ideas on how this could be handled. For example, 10 participants indicated they would like to have messages automatically become unreadable after they have expired. Supporting this approach, P10A and P11A stated:

"You could also have it like... Have it preset to expire at that time... So you don't have to say keep or delete it. It just deletes it if you already say at that time that you can delete."

"I think that gives you fair warning and so I think it's pointless almost if it doesn't do it itself after that period of time."

On the other hand, 11 participants voiced their opinions on how messages should not automatically become unreadable after they have expired. For example, P17B and P22B said:

"I think, I like the idea of being able to see the message before it gets deleted forever... That way I can know what's disappearing. You know? Just as it's important for me to see what is coming in to my email, I like to see what is leaving my email... That way I don't have any errors come up, like oh man, I lost that email. I still needed it for another day and a half or whatever... Having a warning is nice."

"No, I like that it, you know, allows you to keep it a little longer. I mean you know in this busy life sometimes we might forget, or you know, that little mini heart attack where you think oh shoot it's deleted after 30 days but like now it gives you that option to revive it I suppose..."

As two participants were thinking about the trade–offs between our implementation and making messages unreadable automatically, they came up with another solution: to

70

warn users before messages expire, but automatically make them unreadable once they do expire. This is best expressed in the words of P24B:

> *"I think that would be useful on the one hand because... I tend to let my email inbox pile up I don't keep it cleaned out and permanently delete things... But on the other hand, I would like the option... I would like to have a message popup that says, this message is due to become unreadable in the next 24 hours if you want to save it do this. If you don't care click here, kind of a thing."*

Interestingly, 9 participants indicated they want SLK to allow them to re-request access to information once it is made unreadable. Essentially, they believed their friend could send them some kind of access code to let them into the information again after having revoked their own access. While this doesn't make much sense in terms of short-lived keys, it is important to know that some users have this expectation of a tool that makes messages inaccessible after a period of time. On the other hand, this may only be a critique applicable to our prototype, because we combined short-lived keys with our email-based key exchange that was disguised as an access request.

7.3.4 Message Management Popup

When asked if they liked the popup asking them to manage expired messages, most participants answered in the affirmative. Overall, 32 participants said they liked it, while 2 participants said they didn't. In general, users liked the popup because it was a good reminder to them to manage their messages and because it gave them options for managing their expired messages. Seven participants gave feedback saying the popup was helpful. Participants supporting the use of the popup included P21A and P22B. They respectively stated:

> *"I think that popup was necessary. It was good. I liked the choice."*

"No, I liked that. I like it to be right in my face. Not just like, you know, a little sign on the title of the message but like right in my face, that way I know like when I missed a day."

Two of the participants who did not like the popup said it was, or would become, too annoying. For example, P24A said:

"I found it a little aggravating... When it's important to me, I'm generally pretty good about managing what I need to and taking care of my stuff... Having stuff coming up to me and saying, 'Hey, take care of me! Hey, look at me!' kind of stresses me out."

Even two participants who said they liked the popup were also worried about it becoming annoying. In line with this, P3B, who initially expressed support for the popup, said:

"If it was like every time that would be dumb."

This comment helps show the limitations of the results of this user study in terms of the message management popup. Even though many participants liked the popup after limited exposure to it, it may be that a majority of them would begin to dislike it after encountering it more frequently or over a longer period of time.

7.3.5 Protection and Expiration Bundling

One of the final interview questions asked for feedback about the way SLK encrypted and expired messages on the thread level. Participants were asked if they would prefer expiration and protection to work on the thread level, or if they would prefer it to work another way. Generally, if participants could think of no other way for this to work on their own, the study coordinators would give them ideas, such as single email messages expiring by themselves.

72

In response to this question, 24 participants indicated they liked the idea of bundling expirable encrypted messages at the thread level. Seven of these explained that this is a good idea because threads are generally about one thing and if any of it should go, it should all go. Further, some participants supported the concept of bundling expiration and protection at the thread level because it would be convenient to do so. In support of thread bundling, P17B stated:

"My gut reaction is yes, I love having it for the whole thread… That way, in the event of somebody looking into the information, they have no idea what was exchanged. They have no idea. Maybe they can guess from the subject lines what the nature of the correspondence was, but other than that, I like the idea of… not having access to anything… Because at the end of the day, like, if you're getting rid of the most important piece of information, why not get rid of the entire thread… Just cause you don't need the less important stuff if you don't need the more important stuff."

After considering other options for message bundling, 17 participants expressed a desire for an option allowing messages to be protected and expire by themselves. Interestingly, 9 of those participants also expressed interest in thread bundling. P18B made the following statements expressing the desire for this option:

"I personally like it all handled inside the thread but I can see some cases where you'd want to… to delete one email instead of the entire thread."

Several participants thought hard about the trade–offs between approaches to message bundling, thinking out loud about some of the benefits and problems they would encounter with them. This is best expressed through P23B's comments as they worked through the problem out loud:

"I have to say it depends on the email. If I have to do one or the other, I'd probably say all of them at once… Part of me sits down and says in this thread

73

I was doing all the emails at once so it's all at one time and it's all over one day, but sometimes, and this goes back to my study for my doctorate, that emails come over two or three day. That we're talking back and forth and I have to wait 6, 8 hours to get a response back from you, back and forth. In which case, then no, maybe it would be better to just go by thread by thread... It would depend on the length of time from the start to the end of the thread. So like if it's a few hours, then yeah all of it together. But if it's a thread that's being spread out over several days, or several weeks, then I may not necessarily want it to. But if I had to choose one, then I would choose all or nothing, not each one. Because I would find each one, having to got through each one being annoying. Or the best again is a toggle that says the control of I want to, do you want to do all of them or do you want to do them one at a time and give me the choice of which one I want to go for."

7.4   Information Permanence

Study coordinators asked participants four questions related to the permanence of their information on their devices and in the Internet. Summaries of the feedback given in response to these questions are provided in the following four sections.

7.4.1   Participant Email Persistence

When study coordinators asked participants how long their emails exist in their email inboxes, most participants indicated that they keep their emails indefinitely or for a long time. Fourteen participants indicated that their emails exist in their inbox forever, 14 stated their emails exist forever unless they delete them, 7 participants mentioned that they never or only rarely delete their emails, and 18 participants gave a more general answer of "a long time". In terms of emails existing in their inboxes for long periods of time, P21A and P10B said:

74

"It's an embarrassingly long amount of time. Indefinitely."

"They last a long time actually. Like, if I don't go put them in the trash and then delete things forever from the trash... I was looking for something last night actually because I couldn't remember the emails I was looking for. It pulled up search results from like 2015. I was like, 'Oh! I still have those.' So... Yeah, unless I go in and delete them then they just stay for a while."

Only 6 participants said their emails don't last in their inboxes for a very long time. One of those participants, P4B, said:

"Usually like 2 weeks. I delete my emails pretty quickly. I try to keep my inbox pretty empty. Like less than 50 on the front page."

One participant (P17B) expressed concern about their emails actually being completely deleted after deleting them from Gmail's trash folder when they said:

"I admit I don't know exactly what happens to that data after I delete it from the trash can. I imagine it is still accessible on some level... But these days, if I don't need an email, I get rid of it completely... Again, mostly to save space."

Finally, 14 participants indicated that how long their emails last in their inboxes depends on the email. For example, P20B stated:

"It depends. If it's more academic work, business related, I will keep that for longer. But if it's school related I probably delete every semester. If it's an advertisement I delete it right away."

7.4.2 Worry About Message Permanence

When participants were asked about how much they worry about the permanence of their messages on any of their devices, most participants expressed having little to no worry. Specifically, 33 participants said they were "not too worried", while 11 were somewhat worried, and only 3 were "pretty worried". Out of the participants that were "not too worried", or only somewhat worried, 10 indicated they felt this way because they don't have sensitive emails. The outlook of many of the participants that weren't too worried about the permanence of their messages on their devices are reflected in the words of P15A and P4B, shown in that order below:

> "I don't really have very many important things in my email boxes, just, I'm not really very old. So, I don't really have to do many important things. My main issue is they just take up lots of space and it's just I have to go in and manually delete them to clear up space on my computer—so that's annoying."

> "I don't know, I don't feel that worried about it. I don't really have a lot of incriminating things on my phone or on my laptop. I'm a pretty open person, so it doesn't really matter to me a whole bunch. But, I can see how it would be very important in a professional situation."

In response to this question about their message permanence, 6 participants answered saying that permanence can be good thing. However, participants differentiated between permanence being good for regular messages and permanence being bad for messages containing sensitive information. This feedback further confirms results from work by Odom et al. [26], Jacek Gwizda [19], and Waugh et al. [45]. Qualitative data gathered from participants in work by Odom et al. [26] indicates that Internet users regard information permanence as important in many ways, including the data safety that comes with backing up pictures and important documents. Odom et al.'s work also indicates Internet users want to be able to delete their messages and information, something that

76

is shown in the results of this study and work by Jacek Gwizda [19]. Some of these sentiments were expressed as P17B stated:

> "So, permanence is nice for general day to day stuff... I don't have too much sensitive information going over the Internet, so it's nice that I have a record... I used a messaging app for two years and it was nice to be able to scroll way back to the beginning and see messages from two years ago. But, I wasn't dealing with sensitive information. There is, again, like I described this uneasiness about the permanence of more sensitive information, just the longer something stays on the Internet, the more likely it is that somebody is going to stumble across it."

While expressing that they were not worried about their message permanence, several participants expressed that they probably should be more worried. On this vein of thought, P13B said:

> "Generally I'm not sending too many personal things, sensitive information, pretty confident people can't see it, but I think I probably should be..."

One participant (P22A), explained that they worried about the permanence of their information to different degrees for different devices. They said:

> "Maybe this is ignorance on my part, but I'm not that worried about my phone, just because I don't feel that phone hackings are really common. And I could be wrong on that, maybe because it hasn't happened to me... As far as laptops... Laptops and computers I do get a little worried, just because I've had my laptop stolen before with sensitive information on it. So... That's a real big like, 'Oh shoot like now I have to like clean everything before they can figure out the password to that laptop'... Then what was hard though on that was that I did have actual tax information on that laptop saved... I don't think whoever stole it got into it though because nothing ever happened."

Representing those who are "pretty worried" about the permanence of their messages on their devices, P23A simply stated:

"Quite worried... Cuz email's easily hacked no matter [how] secure they say it is."

Interestingly, while answering this question, several participants expressed concerns that once you delete your messages, they may still exist in some form on the Internet or your devices. For example, P23A and P20A respectively stated:

"Well I know you can delete them and they're still there anyways, they still can be retrieved."

"I'm actually kind of nervous about old things that I have on my phone. Whether it's emails or like information I've downloaded. There's just so much information that I forget where I've subscribed and what kind of information I've given out so...even if I delete it or unsubscribe, it could still be on there. So, that makes me a little nervous, but I don't know how to fix it."

Participants provided numerous other explanations for their level of concern about the permanence of their messages. The codes for this subject are summarized in Table 7.3.

Overall, participants seem to express little concern about the permanence of their messages on their various devices. There are several reasons that were given. First, many participants care little about the permanence of their day-to-day messages, but do what they deem necessary to manage their sensitive messages. They either delete the messages or don't send them at all. On the other hand, participants aren't worried about the permanence of their messages, because they haven't been attacked and assume that entities like the government already have their information anyways. Other participants express little worry because they can't control what other people and devices do with their messages.

78

Code	#
Not too worried	33
Somewhat worried	11
I can delete things already	6
Permanence is a good thing	6
Pass codes on phone and email accounts keep them secure	4
Pretty worried	3
I can't control what other people do with my information	3
I don't do secretive or illegal things	3
I delete things when I run out of space	3
Worried that someone else can access my emails	2
Avoid creating accounts where unnecessary	2
Nothing has happened to me yet	2
I know someone who is concerned	2
I should probably be more worried	2
The Government already knows my information	2
I don't really think about it	2
"How many people can see this?"	1
I delete when I run out of space	1
It's important to keep other people's information safe	1

Table 7.3: Codes for message permanence concerns.

Not Too Worried	A Little Bit Worried	Medium Worry	Fairly Worried	Really Worried
10	12	1	3	8

Table 7.4: A summary of responses on how worried participants are on the permanence of their information on the Internet in general.

7.4.3 Worry About General Information Permanence on the Internet

When users responded to questions asking about how worried they are about the permanence of their information on the Internet, they defined more levels of worry than they did while talking about the permanence of their messages. A summary of the responses of the degree of worry participants felt are given in Table 7.4. This table shows that a majority of participants had little to no worry about the permanence of their information on the Internet in general. However, there were 11 participants that were either "Fairly Worried" or "Really Worried" about this subject.

Some participants said they focus on being careful with their sensitive information online. Other participants explained they are careful with what they put on the Internet in general or don't put that much information on the Internet in the first place. Many of the participants that employed these self-filtering coping mechanisms expressed little to no worry about the permanence of their information on the Internet. These responses confirm the results of other work showing users self-filtering as they deal with their data on the Internet [22, 48]. Along the lines of self-filtering, P10A stated:

"I am like fairly cautious about what I do decide to like post, cuz I know that it is pretty permanent what I put out there. And so, I guess I just like think about what I'm posting. So, I'm not like, super frivolous with the information that I send out I guess."

Some participants gave feedback about their concerns of their information reaching unintended audiences or negatively affecting their future. This feedback further confirms the results of work by Wang et al. [44] and Woodruff [47] as described in Section 2. Even though several participants explained they employ strategies to limit their data footprint online, the same participants expressed worries about their information negatively affecting their future or reaching unintended audiences. For example, P18A and P23B respectively stated:

"I just don't want stuff to come back and haunt me. You know, people do stupid stuff and post it, so, I don't want to be that guy not to get the job because some stupid decision way back when… I don't know, I feel like I don't need to put information online, and so I limit it."

"I'm considerably worried about it. It was one of the main reasons that I took so long to go into social media. Because of my background teaching junior high and high school, I know that anything you put on the Internet is there

forever and it is virtually impossible to clear, like if I were to post a picture on there and say no I don't want that picture on there and somebody wanted to... If people wanted to use it maliciously it would be virtually impossible to track down in my opinion."

When asked about information permanence on the Internet, 12 participants discussed social media. In general, the points participants brought up about permanence and social media reflect responses and data gathered in other work involved with social media, data ephermerality, and data permanence [7, 20, 49]. For example, some participants indicated they would be more willing to frequently use social media if it didn't come with problems related to data permanence. While discussing the permanence of information on social media, participants not only expressed the need to be careful about what individuals post about themselves, but they also expressed worry about the permanent effects of someone posting others' sensitive information. For example, P22A and P7A stated:

"I get pretty worried. Not because I post anything dumb. But just because like if someone posts something about what I'm featured in or tagged in for an example, like I have no control of that."

"... Like Facebook, like all your information is just kind of out there all the time, and again I try to be aware of what I put out there, but I feel like there's just all these creepy people out there, you know, that just will kind of just read more into it... Or just like pull... I don't know, maybe just kind of glean stuff little by little, so that worries me sometimes, but... I try not to think about that all the time."

81

Code	#
Careful with what I put on the Internet in general	13
Has social media worries	12
Careful with sensitive information online	10
Worried about identity theft	6
Don't put that much on the Internet in the first place	5
There's probably information I don't want out there	4
I'm not sure what I can do past what I'm doing already	4
Worried about identity theft	4
I probably should be more worried	3
I trust websites and their security measures	3
I forget where I have put my information on the Internet	3
I try not to think about it	2
I don't know how the Internet works	2
Google knows everything	1
It's not life or death	1
I'm not worried about my flirtatious texts being read	1
Worried about permanence of passwords	1
I research companies I do business with	1
Companies have the right to look at my information	1

Table 7.5: Codes for Internet permanence concerns.

Participants expressed many ideas and reasons for their level of concern for this subject. For brevity, the codes related to these concerns, as well as the number of participants that expressed them, are summarized in Table 7.5.

7.4.4   Interest in Short-Lived Keys Tools

When asked about whether they would want to use a tool that makes messages unreadable after a certain period of time, 35 participants answered in the affirmative, 5 responded negatively, and 3 participants expressed indifference to the idea. P20A expressed their desire for this kind of tool saying:

"Yeah, I'd be really interested in that. I feel like if I don't use something in a long time, then I don't need it. And if it gets deleted then it won't matter. But, because you forget that you even have that information it's still out there

82

www.manaraa.com

somewhere and you forget to go back and delete it, so...it would be nice to have somewhere and you forget to go back and delete it, so...it would be nice to have something that gets it automatically deleted."

Seven participants explained they would want to use the tool only if they had the need, and 6 said they liked having the option available. On this vein of thought, P24B said:

"I tend to let my email inbox pile up I don't keep it cleaned out and permanently delete things probably the way I should. But on the other hand, I would like the option."

Further, 3 participants stated they wouldn't want to use this kind of tool in their everyday transactions. For example, P17A stated:

"I wouldn't necessarily want that to happen with everything. Cause there is some stuff I put on the Internet to make sure it stays there forever because eventually I need to delete pictures off my phone, you know."

7.5   Sending Sensitive Emails

During the exit interview, participants were also asked about whether they send sensitive information through email. Overall, 12 participants responded saying they do send sensitive information through email, 20 said they don't, and 16 indicated they sometimes do this. Participants who indicated they have or do send sensitive information were also asked what kind of sensitive information they send. The responses to this question are summarized in Table 7.6.

Some participants expressed that they prefer to use modes of communication besides email to communicate sensitive information. For example, 10 participants said they preferred to use phone calls for this purpose, 5 participants prefer to talk to people in person, 6 participants prefer texting, and 1 participant prefers sending sensitive

Type of Sensitive Information	#
Account (usernames, passwords)	7
Tax	5
Insurance	5
School	4
Banking	4
Credit card	3
Immigration	1
Addresses	1

Table 7.6: Types of sensitive information participants indicated they send through email.

information through pictures. Interestingly, participants that expressed preferring some of these modes of communication to email for communicating sensitive information reasoned about the security of their preferred modes compared to the security of email. For example, P10A and P23A stated:

"I guess a text isn't much better than email. But yeah, I haven't done too much through email. Just because I know that like people can view that information... Kinda freaks me out."

"I don't like... to send stuff through email that needs to be secure. I would rather, like, phone and talk to the person that needs the information and give it to them. Or... Take a picture of it and send it via picture rather than email. I don't know that it's any more secure, but I kinda feel like it is."

7.5.1 Likes

Participants were asked about what they liked about LTK and SLK. In response to this question, 29 participants said they liked that the prototypes were user friendly and 23 expressed that the prototypes were straightforward/simple. For example, P8B, and P6A respectively stated:

"I feel that they were extremely easy to operate... After I understood the basics, it was breeze to get through... the task... It was very easy to use. Very user

84

friendly, like it wouldn't take an advanced specialist to operate the tasks. It could be anybody."

"They definitely didn't seem complicated at all. Which was nice. Like, it made it feel like, you know, there were less ways to make it go wrong or something."

Some participants also liked the make unreadable functionality (9) and expiration functionality (8) of SLK. Expressing ideas related to this functionality, P3A said:

"When I was using the second one, I realized that it was very nice to be able to kind of digitally destroy whatever information I had already sent and received."

Ease of setup was another thing participants brought up that they liked about the prototypes. Overall, 10 participants expressed appreciation for this feature. Relatedly, 9 participants commented on the usefulness of the inline tutorials that were available as participants first used the prototypes. Talking about the tutorials, P4B stated:

"It would like highlight a little area and it would say, 'This button is what you do for this and this is how you encrypt and this how you delete' and stuff like that. So, I thought that was very helpful and made it much more user friendly instead of just me having to play around and figure out what the prototypes do by myself. I thought that was very good."

Importantly, participants said that they liked that they felt secure while using these prototypes. Overall, 14 participants expressed this view. For 5 of these participants, the access requests they saw as part of the email-based key exchange made the prototype feel more secure. P4B expressed this feeling of security by simply saying:

"I like that it gave me a sense of security in my emails."

Tutorials in both prototypes highlighted that Google would not be able to read emails encrypted with the prototypes. This concept stood out to 5 of our study participants.

85

Code	#
The prototypes were user friendly	29
The prototypes were straightforward/simple	23
They felt secure	14
Easy to set up	10
Liked make unreadable functionality	9
Liked inline tutorials	9
Liked expiration functionality	8
Liked labels	6
Access request made it feel more secure	5
Google cannot read encrypted emails	5
Integrated well with Gmail	2
Liked color scheme	2
It was quick	1
Liked button placement	1
Liked encryption animation	1
Good level of privacy vs. ease of use	1

Table 7.7: Codes for things participants liked about LTK, SLK, or both.

Interestingly, for several of the participants, this concept seems to have stood out to them, because they had never thought about Google having the ability to read their emails before. For example, P12A stated:

> "I thought it was cool that Gmail couldn't read it. I didn't think of Gmail reading my emails before."

The things participants liked about the two prototypes are expressed in the codes summarized in Table 7.7.

7.5.2 Dislikes

Participants were also asked about what they disliked about LTK and SLK. While at least 8 participants expressed that it was hard to think of something they disliked, or they couldn't think of anything they disliked, many participants gave useful feedback and criticisms on how the prototypes could be better. Generally, participants' dislikes of the prototypes centered on what was confusing and what they did not understand. Much

86

of these misunderstandings were related to the functionality of the "Make Unreadable" button, the meaning of expiration, and the access requests participants saw through the key exchange process. Some of these misunderstandings and points of confusion are seen in the following statements made respectively by P22A and P12B:

> "I guess I just didn't understand that when I clicked make unreadable, it wasn't make unreadable to the person who I sent it to. It's make unreadable to me."

> "I didn't understand why the other person has to... ask for access. And then the system will automatically send that or give the access... I just don't know why there is this step... I think that if the system can automatically verify the other person's identity why should I, why should it bother. And then if it can't why does it automatically send out the access?"

While many participants liked the make unreadable functionality of SLK, others did not. For example, P20B said:

> "The finality of the unreadable tool. Which makes sense though because if you like shred a paper it would be the same thing, just for your email. But, it made me feel a little nervous for some reason. Because you expect to be able to find anything you need on the Internet."

Three participants felt the two prototypes would be improved by adding a master password to keep unwanted eyes off encrypted email. Another participant, P20B, expressed a need for a way to give people access to secure emails from multiple devices. They explained:

> "Let's say that like people left... Their computer and not check email, but like let's say ask for the access. That person can receive the notification from their phone and just push it and give me access to get the... information so I don't have to wait for the person to get back to me."

Codes gathered from responses to this question are summarized in Table 7.8.

Code	#
Hard to think of something I disliked	8
LTK was better	4
SLK was better	4
Did not like the color choices in the prototypes	4
Needs master password	3
Did not like the key exchange	3
Both prototypes were good	2
Wants warning added to "Make Unreadable" button	2
Worried about whether or not friend revokes access	2
Did not like the prototype download page	2
Wants encryption turned on automatically	1
Secure email tools should work with multiple email accounts	1
Should work with multiple devices	1
Did not like MessageGuard overlays	1
Misunderstood MessageGuard overlays	1
Wants unencrypted replies available	1
Wants more involved key exchange ceremony	1
Did not feel secure	1
Make tutorials always available	1

Table 7.8: Codes for things participants disliked about LTK, SLK, or both.

7.5.3 Other Feedback

We received a large array of miscellaneous feedback throughout the course of the exit interviews we conducted. For example, 3 participants expressed that many of the problems they encountered were their own fault due to their exploration of the prototype. Five participants were unsure about installing the prototypes due to the permissions they requested. Further, two participants highlighted the significance in prototype testing order by saying the second prototype they tested was easier to use due to their experience with the first prototype.

Several participants expressed concerns about knowing whether a secure email tool is actually secure. Another participant indicated that a secure email tool that is too easy to use may seem insecure. These ideas are summarized, respectively, by P10B and P11A in the following statements:

"I'm just kinda a paranoid person when it comes to Internet security in general. It's probably cuz my dad works in IT security stuff, so he's kinda just ingrained in that into me and my siblings... I'm just wary of sending any type of sensitive, personal information on the Internet regardless. So, I think if I were to use something like this, I would want to heavily do more research to be 100% sure that this is safe to use."

"That made it all seem like phony because it was so easy."

One participant (P22B) argued that nothing can be too secure these days. They also summarized many other participants' concerns about hackers when they said:

"I got sold I suppose on it just because you can never be too secure, especially these days, how you know technology's improving and like how hackers are getting more powerful as well. And so, I thought, you know, it gave me satisfaction to be able to use a tool in particular the version B of it. And so, it was good I liked it. It was simple."
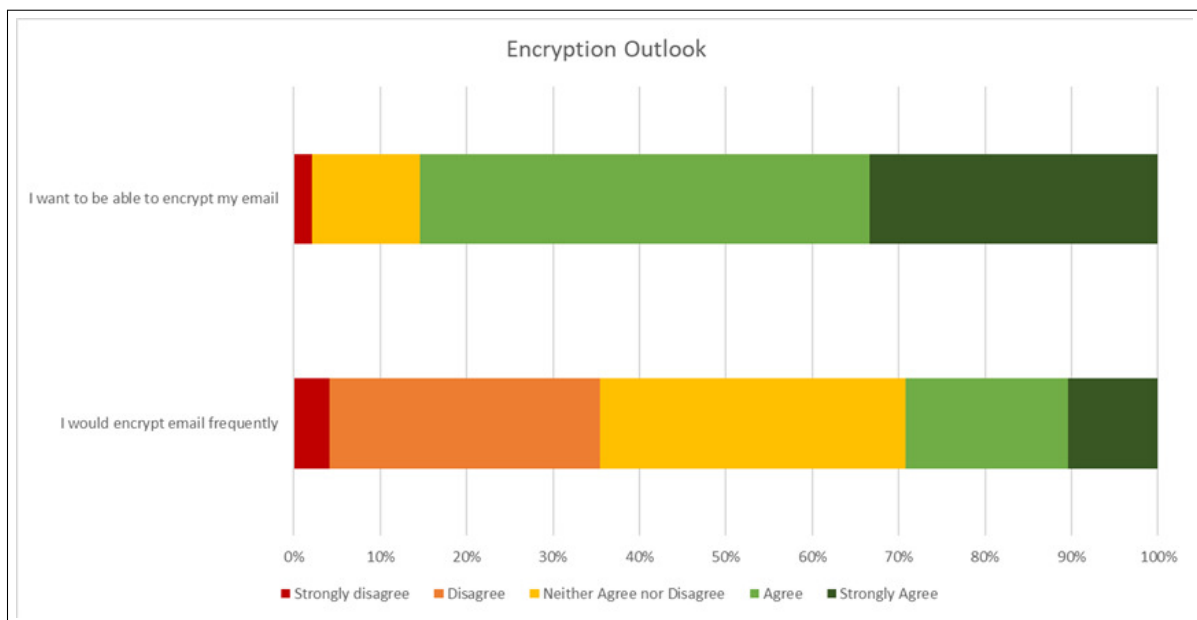

Figure 7.1: A summary of participant responses to the encryption outlook questions.

Not Very Likely	Unsure	Likely	Very Likely
6	2	26	10

Table 7.9: A summary of responses on how likely participants are to use either LTK or SLK in the future.

7.5.4 Encryption Outlook

At the end of the study survey, we asked participants to answer two Likert-scale questions based on their future potential use of encrypted email:

1. I want to be able to encrypt my email.

2. I would encrypt email frequently.

The results of the responses to these questions are shown in Figure 7.1. In general, we see a trend similar to data gathered by Ruoti et al. [31, 34]. Essentially, a majority of participants indicate they want to be able to encrypt their email, but the majority of participants do not expect to encrypt their email frequently. This calls for secure email tools to be implemented in such a way that their usability is tailored to infrequent use scenarios.

As part of the exit interview, participants were also asked about how likely they were to use either LTK or SLK in the future. A summary of the responses participants gave related to their likelihood of future use of these prototypes are shown in Table 7.9. The table shows that most participants would be at least likely to use one of these prototypes in the future.

More commentary from participants on this subject sheds more light on these likelihood values. For example, 15 participants indicated that they don't need this kind of tool in their current life situations. Further, 13 participants guessed they would use a tool like this infrequently in the future, while only 3 expect to use it frequently. This feedback matches up well with the encryption outlook quantitative data shown in Figure 7.1. Fifteen participants also indicated their future use of one of these prototypes would

90

depend on the context they find themselves in. Some of the participants expressing these sentiments were P13B and P17B, who respectively said:

> *"I probably wouldn't use it unless I had like a business and I had to have like secure information. If I had a business where I need to delete information, I can see myself wanting to have something like that maybe. But, there are some emails I don't want people to see that I could see using it if it's a free thing. But, if I had a business I could see wanting to buy it. But, if it's on my own, I probably wouldn't buy it."*

> *"I'm planning on going into counseling psychology where we just, we have... this obsession almost with keeping things confidential and I love keeping whatever I can confidential... So, being able to have a system, especially if the system were well known and were well trusted by the general population, being able to exchange that sensitive information when absolutely necessary would be useful..."*

Some participants expressed that having these prototypes available would allow them to feel safe sending secure information through email, something they have avoided in the past because they didn't feel safe doing so. For example, P17A stated:

> *"I don't often send secure information over email anyway. But, I think if I had the option to, I would... I didn't ever have a tool like this, so I've always thought like 'oh it's not safe', so I don't send anything over email that's secure. With an option like this, I would be much more likely to do it. Definitely with B and even A."*

Participant P10B made an important point about the importance of trust in a secure email tool dictating possible future use. They said:

Code	#
I don't need this right now	15
Future use depends on context	15
I would only use this occasionally in the future	13
Would prefer to use SLK	10
I would use this frequently in the future	3
Wants more information on how it works	3
Would prefer to use LTK	2
Would use either of the prototypes	2
Doesn't expect to send encrypted messages	2
Onboarding complicates desire for future use	2
Prefer to do sensitive things in person	2
Wants other people/entities to use one of these prototypes	1
Feel safer with these prototypes vs phone	1

Table 7.10: Codes related to the likelihood of participants using LTK or SLK in the future.

"I mean a big part of it depends on if I actually 100% trust the system. So, assuming that I do. Being raised with the paranoia of Internet security, I probably would use it a lot. But again, that's assuming that I have full confidence in it."

The sentiments expressed by participants when they were asked this question are summarized by the codes seen in Table 7.10.

7.6 Misconceptions

As we conducted, transcribed, and coded interviews, we discovered several misconceptions users had about our prototypes, especially SLK. For instance, several participants expressed that they could simply send their friend another access request on the encrypted thread once they revoked access to the thread by making it unreadable. Interestingly, several other participants made statements indicating they thought access to the encrypted thread was revoked for their friend when they revoked it on their side. For example, talking about why they liked SLK, P3B said:

"Because you could... delete the encrypted message so my friend couldn't see it anymore."

This misconception not only shows that steps should be taken to explain this is not the case in our prototype, but it also gives some insight into the fact that participants are thinking about this kind of feature and may want it in future tools.

Several participants made incorrect statements on how the encryption works and is used in these prototypes. For example, one participant (P18B) thought that before using the "Make Unreadable" button, their secure emails were in an unencrypted state. This participant reasoned that using the "Make Unreadable" button encrypted the messages so they couldn't be read again. Future work can clearly do more to provide tutorials and steps in the prototype setup that help clear up these misconceptions.

Chapter 8

Discussion

In this chapter, we further discuss quantitative and qualitative results presented in Chapters 6 and 7. Reviewing and discussing these two sets of results at the same time allows us to better understand the information in each section. For example, while there were some limitations to the quantitative results in Chapter 6, looking at the qualitative results from Chapter 7 gives more clues as to why SLK had a significantly higher SUS score and was chosen as the favorite prototype more often.

8.1   SUS and Favorite Prototypes Revisited

While many participants felt both prototypes were "straightforward" and very user friendly, a majority of participants choose SLK as their favorite prototype and the difference between the two prototype's SUS scores was statistically significant. However, due to the limitations presented by the lack of make unreadable functionality in the LTK prototype, it is unclear whether participants preferred short-lived keys over long-term keys. Still, the qualitative data presented in Section 7 sheds some light on why a majority of participants responded in favor of B. A majority of participants feel SLK is more secure than LTK and they trust SLK more than LTK. Participants feel this way about SLK for a variety of reasons, but in general, they like the "Make Unreadable" button, the expiration functionality, and the management of expired messages through the popup. Overall, participants showed strong preferences to functionality allowing them to make their messages unreadable.

94

Participants who chose LTK as their favorite also did so for a variety of reasons. However, the most common reason seems to be that they felt more confident using LTK. In general, participants expressing more confidence in using LTK over SLK seem to do so because they appeared to lose confidence in SLK due to a misunderstanding of the "Make Unreadable" button. For example, several participants misunderstood how the "Make Unreadable" button works and either clicked it too early while exploring its functionality or felt nervous using it.

8.2   Information Permanence

In general, participants from our study seem to be more worried about the permanence of their general Internet information than the permanence of their messages on of their devices. However, in both categories of permanence, a majority of participants express little to no worry. Interestingly, many of the participants that expressed little to no worry about these categories also explained that they feel this way because they employ a variety of strategies to manage their information, especially sensitive information. The actions they take to manage their information may indicate that they are more worried about the permanence of their information than they expressed in their interviews.

In our interviews, a majority of participants responded that they would be interested in using a tool that makes their messages unreadable after a certain period of time. There could be several reasons explaining why so many users are interested having such a tool available. As discussed above, participants have a wide variety of information permanence worries and they employ a range of strategies to manage their information on the Internet. In some cases, participants expressed that even though they employ these strategies, they still may have information they don't want "out there".

It may be that participants from our study recognize that such a tool would help them with these worries and would simplify the way they manage their messages online. Further, several participants expressed that they do not send sensitive emails or share

95

much information online, but would like the ability to do so safely. It may be that these participants also recognized that a tool that makes messages or information unreadable after a certain period of time would give them the option to start safely sending and sharing sensitive information.

8.3   Prototype Design

Analysis of the qualitative results from this study makes it clear the design of both of our prototypes can be improved. The most important design lesson we learn from participant feedback is that there is no one size fits all solution for a secure email tool that uses short-lived keys. While some participants like the idea of completely automatic key destruction, others support the partially automatic approach we took. Other participants even suggested a hybrid approach: one that would warn users with the expired message management popup sometime before the keys expire, but would also automatically destroy the keys once they expire. Further, even though most participants support the notion of thread-based bundling for encryption and expiration, many of the same participants expressed interest in single message expiration. Finally, many participants supported the idea of an option to explicitly set expiration times for messages, but also supported the use of a default time. Even though many participants supported the use of one month as a default time, some participants suggested a variety of longer and shorter default times.

Essentially, participants' opinions of the ideal short-lived keys tool differ drastically. This suggests the need for a tool that has reasonable defaults, but also allows its short-lived keys options to be customized. Customizable options could include settings for key coverage, expiration, message bundling, and automation of key destruction. Adding such options would require further design exploration, as well as security and usability analysis. A better short-lived keys tool would also add more tutorials to help reduce user misconceptions. These tutorials would teach users about the key exchange as well as specific features related to short-lived keys.

96

Chapter 9

Conclusion and Future Work

In this thesis, we advanced the knowledge of the usability and security of short-lived keys. First, we explored the usability and security space of short lived keys, generating important questions and ideas, as well as gathering initial feedback from people through a short pilot study. Second, we designed and implemented a short-lived keys secure email prototype supporting email-based key exchange. Third, we conducted a user study using pairs of participants. In the study, participants tested two secure email prototypes: one based on short-lived keys and one that used a more traditional long-term keys approach. Quantitative and qualitative data was gathered through survey questions and semi-structured interviews.

The user study results show that study participants trusted the short-lived keys prototype more and felt it was more secure than the long-term keys prototype. In general, participants liked the ability to make their messages permanently unreadable and would like to have short-lived keys tools available in the future. They also gave valuable feedback about short-lived keys user interface features and key management options. This feedback can be used to help implement short-lived keys tools that are highly usable and meet user needs.

Participants from this study also gave interesting feedback about their perceptions and worries about information permanence. In general, participants said they had little worry about the permanence of their online information and messages, but revealed a greater degree of worry than they initially admitted as they discussed the coping strategies

97

they use to manage the permanence of their information. Even though participants say they aren't worried about the permanence of their information, they generally try not to include sensitive information in their messages and are careful about what they put on the Internet.

While this thesis research answers many important questions, the results reveal further research to better understand short-lived keys, as well as user perceptions and worries of information permanence. Ideas for future work include the following:

1. **Design Short-Lived Keys Longitudinal Study:** Our lab study of the short-lived keys prototype was an appropriate first step in exploring the viability of short-lived keys. Use the promising results and feedback from this study to design a longitudinal study that will more realistically test of the usability of the prototype. One significant challenge in this effort is that users only expect to use secure email occasionally.

2. **Customizable Short-Lived Keys Prototype:** Implement a short-lived keys secure email prototype that provides some amount of customization for key coverage, key expiration, automation of key destruction, and message bundling. More exploration, prototypes, and usability studies are needed to find a balance between usability and these customization options.

3. **Deeper Study of Trade-offs:** Users may be able to provide better short-lived keys feedback if they better understand a prototype's usability and security models. Conduct a user study where coordinators explain these models to participants and gather this feedback. The study can use a methodology similar to that used by Bai et al. [3] where users receive some instruction about the prototypes they test, allowing them comment on the trade-offs and give further design insights.

4. **Multiple Device Study:** Conduct a user study exploring the usability and security of short-lived keys in a scenario involving multiple devices.

5. **Mechanical Turk Survey on Information Permanence:** Design a Mechanical Turk survey that involves a much larger sample to explore user information permanence concerns, perceptions, and coping strategies.

References

[1] E Allman, J Callas, M Delany, M Libbey, J Fenton, and M Thomas. Domainkeys identified mail (dkim) signatures. RFC 4871, RFC Editor, May 2007.

[2] Erinn Atwater, Cecylia Bocovich, Urs Hengartner, Ed Lank, and Ian Goldberg. Leading Johnny to water: Designing for usability and trust. In *Proceedings of the Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 69–88, 2015.

[3] Wei Bai, Moses Namara, Yichen Qian, Patrick Gage Kelley, Michelle L. Mazurek, and Doowon Kim. An inconvenient trust: User attitudes toward security and usability tradeoffs for key-directory encryption systems. In *Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 113–130. USENIX Association, 2016.

[4] Aaron Bangor, Philip Kortum, and James Miller. An empirical evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction*, 24 (6):574–594, 2008.

[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 *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 85–95. ACM, 2017.

[7] Andrew C. Billings, Fei Qiao, Lindsey Conlin, and Tie Nie. Permanently desiring the temporary? snapchat, social media, and the shifting motivations of sports fans. *Communication & Sport*, 5(1):10–26, 2017.

[8] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[9] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use PGP. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 77–84. ACM, 2004.

[10] John Brooke. Sus-a quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194):4–7, 1996.

[11] I. Brown, A. Back, and B. Laurie. Forward secrecry extensions for OpenPGP. RFC 2440, RFC Editor, October 2001. URL `https://tools.ietf.org/id/draft-brown-pgp-pfs-03.txt`. Last accessed 18 October 2017.

[12] Ian Brown and Ben Laurie. Security against compelled disclosure. In *Proceedings of the 16th Annual Computer Security Applications Conference*, pages 2–10. IEEE, 2000.

[13] Marta E Cecchinato, Abigail Sellen, Milad Shokouhi, and Gavin Smyth. Finding email in a multi-account, multi-device world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1200–1210. ACM, 2016.

[14] Ramaswamy Chandramouli, Simson Garfinkel, Stephen Nightingale, and Scott Rose. Trustworthy email. In *Second Draft NIST Special Publication 800-177*. NIST, March 2016.

[15] 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 *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 27–39. ACM, 2015.

[16] Simson Garfinkel. *PGP: pretty good privacy.* O'Reilly Media, Inc., 1995.

[17] 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 First Symposium on Usable Privacy and Security (SOUPS 2005)*, pages 13–24. ACM, 2005.

[18] Matthew D Green and Ian Miers. Forward secure asynchronous messaging from puncturable encryption. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 305–320. IEEE, 2015.

[19] Jacek Gwizdka. Timely reminders: a case study of temporal guidance in pim and email tools usage. In *Proceedings of the 2000 CHI Extended Abstracts on Human Factors in Computing Systems*, pages 163–164. ACM, 2000.

[20] Vanessa Hernandez. Exploring ephemerality in social media with the facebook timebomb. In *Proceedings of the 2012 Society for Advancement of Hispanics/Chicanos and Native Americans in Science National Conference*, October 2012.

[21] Andrew Hough. Google engineer fired for privacy breach after 'stalking and harrassing teenagers'. September 2010. URL `http://www.telegraph.co.uk/technology/google/8003925/Google-engineer-fired-for-privacy-breach-after-stalking-and-harrassing-teenagers.html`. Last accessed 18 October 2017.

[22] Ruogu Kang, Laura Dabbish, Nathaniel Fruchter, and Sara Kiesler. "my data just goes everywhere:" user mental models of the internet and implications for privacy and security. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS 2015)*, pages 39–52. USENIX Association, 2015.

[23] Stanley Milgram and Ernest Van den Haag. *Obedience to Authority*. Ziff–Davis Publishing Company, New York, NY, 1978.

[24] Deirdre K Mulligan. Reasonable expectations in electronic communications: A critical perspective on the electronic communications privacy act. *George Washington Law Review*, 72:1557, 2003.

[25] Sean A. Munson, Daniel Avrahami, Sunny Consolvo, James Fogarty, Batya Friedman, and Ian Smith. Attitudes toward online availability of US public records. In *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, dg.o '11, pages 2–9, New York, NY, USA, 2011. ACM.

[26] William Odom, Abi Sellen, Richard Harper, and Eno Thereska. Lost in translation: understanding the possession of digital things in the cloud. In *Proceedings of the 2012 SIGCHI Conference on Human Factors in Computing Systems*, pages 781–790. ACM, 2012.

[27] Hilarie Orman. *Encrypted Email: The History and Technology of Message Privacy*. Springer, 2015.

[28] Ronald Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Proceedsings of the 2001 ASIACRYPT Advances in Cryptology Conference*, pages 552–565. Springer, 2001.

102

[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 *Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS 2013)*, page 5. ACM, 2013.

[30] Scott Ruoti, Jeff Andersen, Daniel Zappala, and Kent Seamons. Why johnny still, still can't encrypt: Evaluating the usability of a modern pgp client. *arXiv preprint arXiv:1510.08555*, 2015.

[31] 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*, pages 4298–4308. ACM, 2016.

[32] 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.

[33] Scott Ruoti, Jeff Andersen, Tyler Monson, Daniel Zappala, and Kent Seamons. MessageGuard: A browser-based platform for usable, content-based encryption research, 2016. arXiv preprint arXiv:1510.08943.

[34] Scott Ruoti, Jeff Anderson, Tyler Monson, Daniel Zappala, and Kent Seamons. The quest to secure email: A usability analysis of key management alternatives, July 2016. URL `http://scholarsarchive.byu.edu/etd/6461/`. Last accessed 19 October 2017.

[35] Scott Ruoti, Tyler Monson, Justin Wu, Daniel Zappala, and Kent Seamons. Weighing context and trade-offs: How suburban adults selected their online security posture. In *Proceedings of the 13th Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 211–228. USENIX Association, July 2017.

[36] Jeff Sauro. *A practical guide to the system usability scale: Background, benchmarks & best practices.* Measuring Usability LLC, 2011.

[37] Bruce Schneier and Chris Hall. An improved e-mail security protocol. In *Proceedings of the 13th Annual Computer Security Applications Conference*, pages 227–230. IEEE, 1997.

[38] 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 *Proceedings of the 2nd Symposium On Usable Privacy and Security (SOUPS 2006)*, pages 3–4, 2006.

[39] Andreas Sotirakopoulos, Kirstie Hawkey, and Konstantin Beznosov. "I did it because I trusted you": Challenges with the study environment biasing participant behaviours. In *Proceedings of the Usable Security Experiment Reports Workshop at the Symposium On Usable Privacy and Security (SOUPS 2010)*, 2010.

[40] Wenley Tong, Sebastian Gold, Samuel Gichohi, Mihai Roman, and Jonathan Frankle. Why King George III can encrypt. http://randomwalker.info/teaching/spring-2014-privacy-technologies/king-george-iii-encrypt.pdf, 2014. Last accessed 19 October 2017.

[41] Emin Topalovic, Brennan Saeta, Lin-Shung Huang, Collin Jackson, and Dan Boneh. Towards short-lived certificates. *Web 2.0 Security and Privacy*, 2012.

[42] Thomas S Tullis and Jacqueline N Stetson. A comparison of questionnaires for assessing website usability. In *Proceedings of the Usability Professional Association Conference*, pages 1–12, 2004.

[43] Manas Tungare and M Pérez-Quiñones. best if used by: Expiration dates for email. In *Proceedings of the 2009 CHI Workshop on Interacting with Temporal Data*, 2009.

[44] Yang Wang, Gregory Norcie, Saranga Komanduri, Alessandro Acquisti, Pedro Giovanni Leon, and Lorrie Faith Cranor. "I regretted the minute I pressed share": A qualitative study of regrets on Facebook. In *Proceedings of the 7th Symposium on Usable Privacy and Security (SOUPS 2011*, SOUPS '11, pages 10:1–10:16. ACM, 2011. ISBN 978-1-4503-0911-0.

[45] Andrew Waugh, Ross Wilkinson, Brendan Hills, and Jon Dell'Oro. Preserving digital information forever. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 175–184. ACM, 2000.

[46] Alma Whitten and J.D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium (USENIX Security 1999)*, pages 14–28, Washington, D.C., 1999. USENIX Association.

[47] Allison Woodruff. Necessary, unpleasant, and disempowering: Reputation management in the Internet age. In *Proceedings of the 2014 SIGCHI Conference on Human*

Factors in Computing Systems, pages 149–158, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1.

[48] Justin Chun Wu. Peering through the cloudinvestigating the perceptions and behaviors of cloud storage users. March 2017. URL `http://scholarsarchive.byu.edu/etd/6175/`. Last accessed 19 October 2017.

[49] Bin Xu, Pamara Chang, Christopher L Welker, Natalya N Bazarova, and Dan Cosley. Automatic archiving versus default deletion: What snapchat tells us about ephemerality in design. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1662–1675. ACM, 2016.

[50] Philip Zimmermann. Building in big brother. chapter Pretty Good Privacy: Public Key Encryption for the Masses, pages 93–107. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94441-9. URL `http://dl.acm.org/citation.cfm?id=212412.212422`. Last accessed 24 October 2017.

Appendices

<center>**Appendix A**</center>

<center>**Pilot Study Documents**</center>

A.1   Semi-Structured Interview Outline

1. My research involves creating secure email tools that help everyday people send encrypted messages so their information can be more secure. Weve already had some success with a tool for Gmail called MessageGuard, but are looking for ways to make it even better. Right now, were working on a new version of the tool that makes encrypted emails unreadable after a period of time.

2. In a tool like this, how long would you want these messages to exist before they become unreadable?

   - Get initial response.
   - Explain and ask about "Currently, we are thinking about letting a message have an initial expiration of 2 weeks, then have the expiration go down to 2 days after the message is opened for the first time" and show the related image.
   - Would this approach work in this tool?
   - Should the time be longer? Shorter?

3. What do you think about adding these countdown timers to the webmail interface to let users know when the message will become unreadable? Show the appropriate image.

4. When an encrypted message expires and should be rendered unreadable, should the software:

   (a) A: Do it automatically without telling you?
   (b) B: Warn you, letting you choose to keep the message around longer if you want to? (Show popup image)
   (c) C: Solely rely on your to initiate making the message unreadable? (Show "Make Unreadable" button image).
   (d) D: Have a combination of the above options?

<center>107</center>

(e) Would you prefer to have options like this integrated into the UI, or show up as an interactive pop up?

5. Ask about the wording of the text and labels. (Show text options)

6. How should expirable messages be bundled?

   (a) Get initial response from participant.

   (b) Explain, "Currently, we are thinking about having all messages on an email thread be bundled together for expiration." Would you prefer it to work like this?

   (c) Should each message expire by itself?

   (d) Should each email thread expire together?

   (e) Should all the messages from all contacts in a period of time expire together?

7. Is a tool with these features something you could see yourself using?

   (a) What could it be used for?

   (b) How often would it be used?

A.2   User Interface Mock-ups
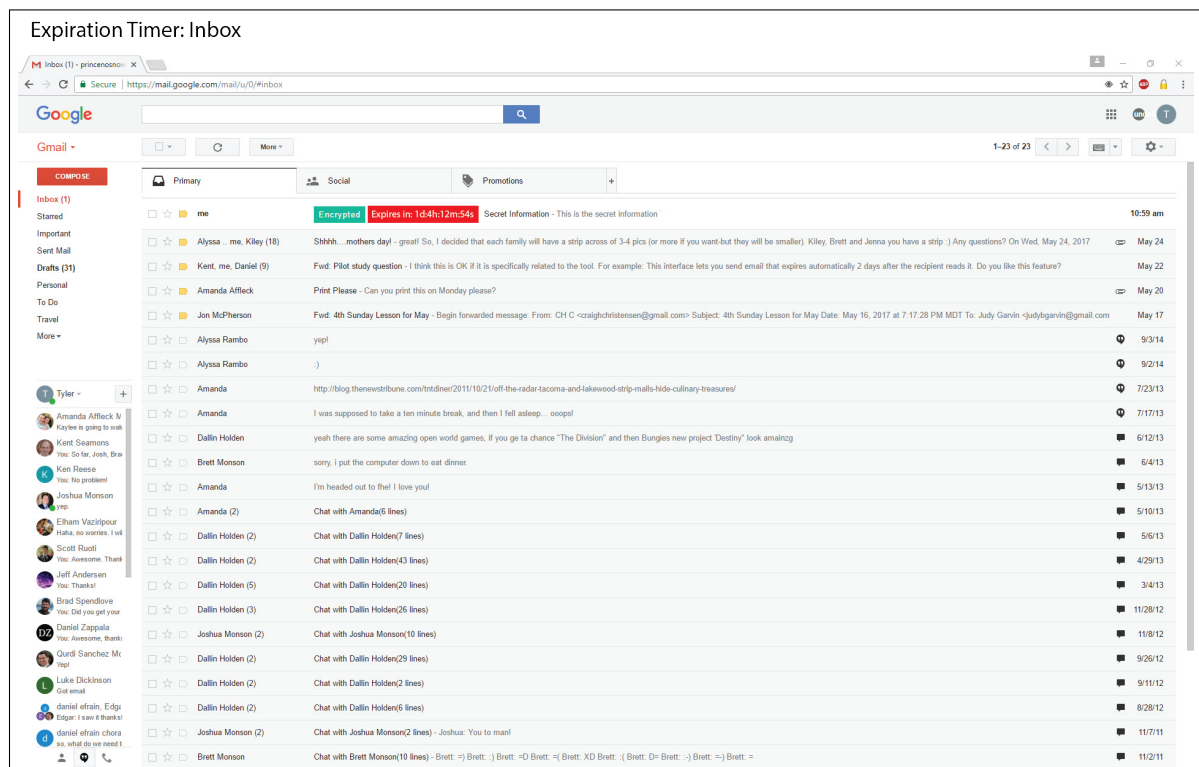
Expiration Timer: Opened Message

Expired Messages: Dialog

Expired Messages: Integrated

Expired Messages: Manual Inbox

Expired Messages: Manual In Opened Message

What are some words/phrases for the tool that would be good for buttons making messages so they can no longer be read?

"Make Unreadable"   "Delete"   "Forgot"   "Scramble"   Other?

What are some words/phrases for the tool that would be good to show a message is expired and should be made unreadable?
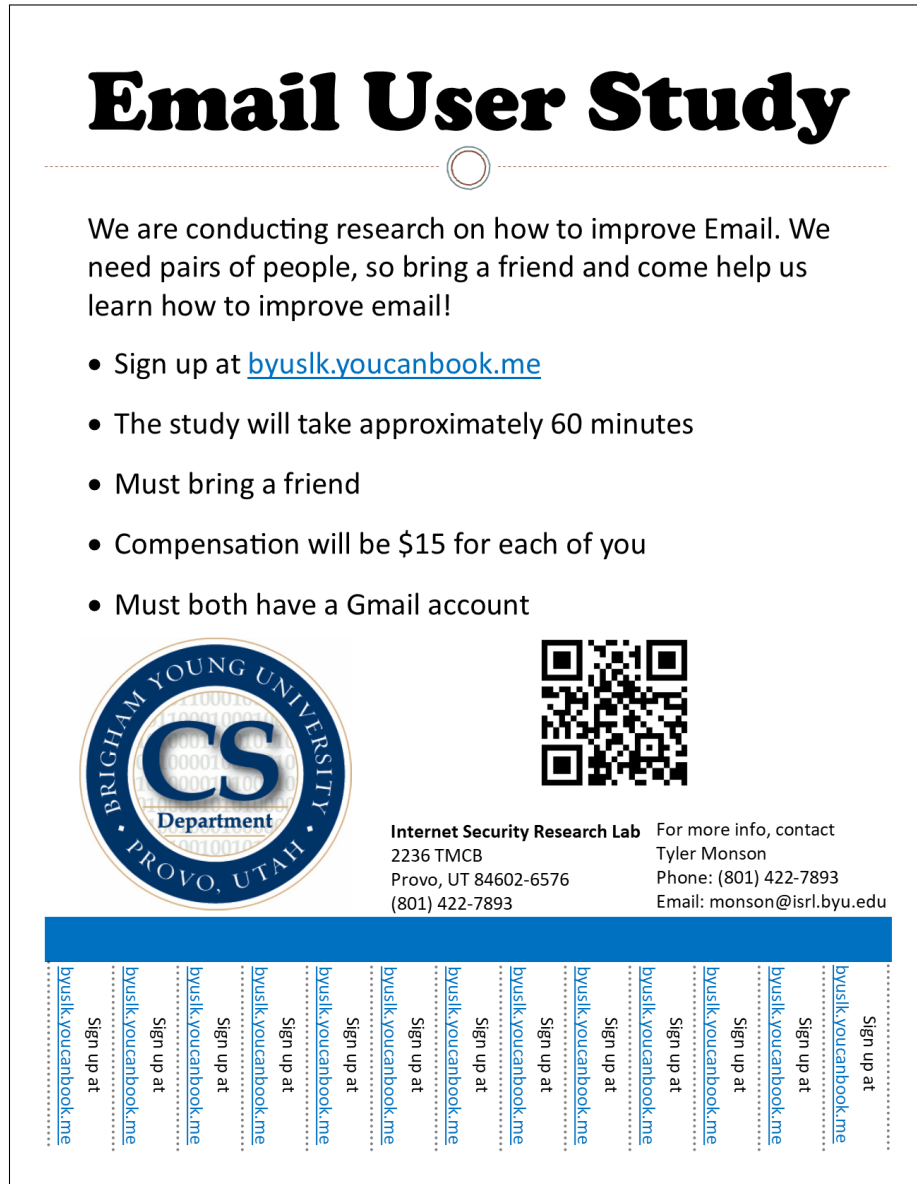
"Expired"   "Overdue"   "Marked for Deletion"   "Unresolved"   Other?

<center>

Appendix B

User Study Methodology

</center>

B.1   User Study Documents

B.1.1   Recruitment Poster

B.1.2   Participant Consent Form

Consent to be a Research Subject

Introduction

This research study is being conducted at Brigham Young University under the supervision of Dr. Kent Seamons, a faculty member in the Computer Science Department at BYU.

Procedures

If you agree to participate in this research study, the following will occur:

- In this study, the two of you will be in different rooms and will use email to communicate with each other.
- You will each be asked to play the role of another person. We will provide you with information about this person. During the study, please use the provided information and not your own personal information.

Risks/Discomforts & Benefits

There is little to no risk to the participants in this study. In order to minimize the risk of any harm, we will not collect any personal information. The purpose of the study is to learn how to design systems that are easy to use. Do not be concerned if you happen to make a mistake or fail to complete a task. You will be helping us reach our objective. If you experience any discomfort, you may stop the study at any time. There are no direct benefits to you for participating in this study. The study will take approximately 50 minutes.

Confidentiality

During the course of this study we will be recording what is happening on the computer screen you are using as well as any verbal communication with the study coordinators. These recordings will not be seen by anyone beside the researchers and will be destroyed once our research is complete. We will not collect any personally identifying information. Your answers to the study survey will be stored in a password-protected account. Only the researchers will have access to this data. A unique, random ID will be generated for each study participant, and this ID will be used in place of any personally identifying information. Data will largely be presented in aggregate, but when direct quotes are required, they will be provided alongside the associated ID and will not contain personally identifying information. We may share research data on the Internet, but will not include any personally identifying information with this data.

Compensation & Participation

You will be compensated $15 for your participation. Participation in this study is entirely voluntary. You have the right to withdraw at any point during the study or to refuse participation entirely. If you withdraw before the end of the study, you will still receive the full $15 compensation.

Research Software Warning

The software you use in this study is research software. For your own safety, you should not use this software outside of this study. Usage of this software outside of the study may result in stolen information or computer compromise.

Questions about the Research

If you have any questions about this study, you may contact Dr. Kent Seamons for further information:
Kent Seamons; seamons@cs.byu.edu ; (801) 422-3722

Questions about Your Rights as a Research Participant

If you have questions regarding your rights as a research participant, please contact:
    IRB Administrator at (801) 422-1461; A-285 ASB, Brigham Young University, Provo, UT 84602; irb@byu.edu.

Statement of Consent

I have read, understood, and received a copy of the above consent and desire of my own free will to participate.

Name (Printed):   _____

Signature: _____

Date: _____

B.1.3 Study Coordinator A Instructions

Study Coordinator A

1. When the two users arrive and have signed the consent form, read them the following:

   Welcome to our email study. We are the study coordinators and are here to assist you as needed.

   In this study, the two of you will be in different rooms and will use email to communicate with each other. You will each be asked to play the role of another person. We will provide you with information about this person. During the study, please use the provided information and not your own personal information. Further, feel free to communicate with your friend the way you normally would while working through the scenarios. There may be times where you have to wait for an email. In these times, please feel free to use the Internet or your phone.

   If you have any questions or concerns, feel free to ask us. A study coordinator will be with you at all times to observe the study and also to answer any questions you may have.

2. Flip a coin. If the coin is heads up, you will be working with the participant on the left. If the coin is tails up, you will be working with the participant on the right.
3. Make sure the other coordinator knows the study number.
4. Take the participant to study room A. Complete the following setup steps:
   a. Ask the participant to sit down.
   b. Start the audio recorder
   c. Open "Open Broadcasting Software". Start recording.
   d. On the desktop, click the "Start Survey" icon.
   e. Enter the study number, click next.
   f. Instruct the participant to start working on the survey.
5. Before using each system, the survey will instruct the participant to tell you they are ready to begin the next task. When they do so, complete the following steps:
   a. Look at which system the participant will be using, and provide them with the corresponding information sheet (A.1 or B.1).
   b. Start the VM software.
   c. Restore the appropriate snapshot (VM->snapshot->Two Person Study)
   d. Change the view to full screen-exclusive mode. (View->Full Screen; View->Exclusive)
   e. Instruct the participant to begin the task.
6. During the course of the test, pay attention to the following items:
   a. Make notes of anything interesting you see. At the end of the study, you can ask the participant about these events.
   b. If the participant sends sensitive information in the clear, make a note of this, then instruct them that they need to use the secure email system to send that information.
   c. During the study, participants may have questions for you. Answer any questions regarding the study task, but do not instruct participants on how to use the systems being tested. Instead, encourage them to continue trying. Remind them that there may be help on the website they were provided.
   d. End the tasks if they take too long. Record this failure.

      i. End the test after 20 minutes unless the participant seems to be 1 – 2 minutes away from finishing all the tasks for the current system

     ii. If you end the task, inform the other study coordinator that you have done so.

    iii. The following codes can be input into the participant's survey to allow them to continue when you have had to end a task early:

System	Confirmation Code	PIN
Long-Term Keys (A)	LDDLDHVO	3866
Short-Lived Keys (B)	OLOWXTUU	8111

7. After the initial task, the participant will inform you they are ready to begin the next task. This task requires them to go back to their email to retrieve the information they received from their friend in the last task. Do the following for the next task:
   a. Take their info sheet with the confirmation code and the PIN. Do not give the sheet back to the participant.
   b. **If testing the Short-Lived Keys system (B), use the options page of the MessageGuard chrome extension to expire the participant's keys.**
   c. Help them role play the passing of time by telling them 3 days have passed and it is now April 6th. Instruct them to flip the calendar to that date.
   d. Put the VM back in exclusive mode and give them the appropriate worksheet (A.2 or B.2).
   e. **For Short-Lived Keys (B):** If the user doesn't attempt to make sure the message can never be read again, remind them to do so. If they indicate they have already tried, let the task end.
   f. If the user indicates they can't retrieve the information again, take note of the occurrence, direct them to the survey, and give them the confirmation code and PIN to get to the next part of the survey.

8. Through the survey, the participant will be given the task of determining what a snooper could have seen. This does not require anything on your part. However, you should watch them closely and be ready to help them if they need it. Remember, don't give the answer, but encourage the participant to try to figure it out.

9. When all the tasks are complete for a system, the participant will be instructed to tell you they have finished. When they do so, complete the following steps:
   a. Ensure that the participants have correctly completed all the tasks.
   b. Reset the VM snapshot.
   c. Instruct the participant to continue working on the survey.

10. When the survey is finished, conduct the exit interview. Remember, this is a very important part of the study and should not be skipped or conducted too quickly unless participants are pressed for time.
    a. **Usability Experience Questions**
       i. Tell me about your experience using these tools (Asking more questions as necessary).
       ii. What are some things that stood out to you? Why?
       iii. What did you like about the two tools you tested? Why?
       iv. What did you not like about it? Why?
       v. In the {first, second} tool, an email thread has a lifespan of two weeks at first and is bumped down to two days after the first message in the thread is opened.

     1. Do you think this is a good default for expiration timing?

     2. How much control would you want over setting expiration times? Would you like to explicitly decide for each email thread, or would you rather the tool do it all for you?

   vi. What did you think about the red countdown timers? Would you want these kinds of indicators in a tool that expires messages?

   vii. In the {first, second} tool, you could choose to expire an encrypted thread by pushing the "Permanently Expire this Thread" button, or you could wait until the timer ran up before choosing to expire or retain the encrypted thread. Is there another way you would prefer to manage expired encrypted emails? For example, would you rather have the tool take care of it for you and not ask you?

   viii. Is modal or integrated SLK management better?

   ix. In the {first, second} tool, the encrypted emails on one thread were all protected together and expired together. Is there another way you would prefer to have these expirable encrypted messages protected? For example, would you prefer each message be protected by itself, or would you rather have all encrypted messages from all contacts over a period of time be protected?

   x. Of the two tools you tested, which one did you feel was more secure and why?

   xi. Of the two tools you tested, which one do you trust more and why?

   xii. Without a tool like this, how long do your emails exist in your inbox before they are deleted?

   xiii. How likely are you to use tool A or tool B in the future?

 b. **Message Permanence Questions**

   i. To what degree are you worried about the permanence of your emails and messages on your mobile devices, laptops, desktops, or other devices? Why?

   ii. To what degree are you worried about the permanence of your information on the Internet in general?

   iii. Would you like to use a tool that makes your message unreadable after a certain period of time? Why?

 c. **General Questions**

   i. Do you ever send sensitive information through email? (usernames, passwords, SSN, financial information, other...)

    1. If so:

     a. How often do you send sensitive information and what kind of information do you send?

     b. Do you ever hesitate, to do this for security purposes?

     c. Do you delete the sensitive email after sending it?

    2. If not:

     a. In what situations can you see yourself sending sensitive information?

     b. How often would you expect this to happen?

     c. Would you delete the sensitive email after sending it?

11. Thank the participants for their time. Help them fill out the compensation forms. Send them to the CS office to be compensated.

<ol type="a">
Inform the participants they must each take their own form to the CS office
Inform the participants of the general CS office operating hours
  <ol type="i">
  Generally 9am – 5pm on weekdays, closed from 12pm – 1pm for lunch

12. Stop the screen recorder. Stop the audio recording.

B.1.4   Study Coordinator B Instructions

Study Coordinator B

1. When the two users arrive, the other coordinator will read them the introduction and select one of the participants. You will work with the other participant.
2. Get the study number from the other coordinator.
3. Take the appropriate participant to the other study room. Complete the following setup steps:
   a. Ask the participant to sit down.
   b. Start the audio recorder
   c. Open "Open Broadcasting Software". Start recording.
   d. On the desktop, click the "Start Survey" icon. Enter the study number.
   e. Instruct the participant to start working on the survey.
4. Before using each system, the survey will instruct the participant to tell you they are ready to begin the next task. When they do so, complete the following steps:
   a. Provide the participant with the general worksheet.
   b. Start the VM software and restore the appropriate snapshot.
   c. Change the view to full screen-exclusive mode. (View->Full Screen; View->Exclusive)
   d. On the first task, inform them it may be a few minutes before their friend sends them an email. Inform them they should check their mail once or twice a minute, but can use the Internet in the meantime.
5. During the course of the task pay attention to the following items:
   a. Make notes of anything interesting you see. At the end of the study, you can ask the participant about these events.
   b. If the participant sends sensitive information in the clear, make a note of this, then instruct them that they need to use the secure email system to send that information.
   c. During the study, participants may have questions for you. Answer any questions regarding the study task, but do not instruct participants on how to use the systems being tested. Instead, encourage them to continue trying. You may remind them that they can communicate with their friend to get help.
   d. End the tasks if they take longer than 20 minutes (give or take 1-2 minutes). Record this failure.
      i. If you end the task, inform the other study coordinator that you have done so.
      ii. The following codes can be input into the participant's survey to allow them to continue when you have had to end a task early:

System	SSN	PIN
Long-Term Keys (A)	979-65-3363	1988
Short-Lived Keys (B)	264-94-8748	6576

6. After the initial task, the participant will inform you they are ready to begin the next task. The new task requires them to go back to their email to retrieve the information they received from their friend in the last task. Do the following for the next task:
   a. Take their info sheet with the SSN and the PIN. Do not give the sheet back to the participant.

b. **If testing the Short-Lived Keys system (B), use the options page of the MessageGuard chrome extension to expire the participant's keys.**
c. Help them role play the passing of time by telling them 3 days have passed and it is now Thursday, April 6th. Instruct them to flip the calendar to that date.
d. Put the VM back in exclusive mode and give them the appropriate worksheet (A.2 or B.2).
e. **For Short-Lived Keys (B):** If the user doesn't attempt to permanently expire the keys, remind them to do so. If they indicate they have already tried, let the task end.
f. If the user indicates they can't retrieve the information again:
    i. Take note of the occurrence
    ii. Direct them to the survey
    iii. Give them the confirmation code and PIN to get to the next part of the survey.
g. When the user is done with this task, help them get to the survey again.

7. Through the survey, the participant will be given the task of determining what a snooper could have seen. This does not require anything on your part. However, you should watch them closely and be ready to help them if they need it. Remember, don't give the answer, but encourage them to try to figure it out.

8. When all the tasks are complete for a system, the participants will be instructed to tell you they have finished the task. When they do so, complete the following steps:
    a. Ensure the participant have correctly completed all of the tasks.
    b. Exit exclusive mode by pressing right ctrl + right alt.
    c. Reset the snapshot on the VM.
    d. Switch to the window that had the participants survey open.
    e. Instruct the participant to continue with the survey.

9. When the survey is finished, conduct the exit interview. Remember, this is a very important part of the study and should not be skipped or conducted too quickly unless participants need to leave quickly.
    a. **Usability Experience Questions**
        i. Tell me about your experience using these tools (Asking more questions as necessary).
        ii. What are some things that stood out to you? Why?
        iii. What did you like about the two tools you tested? Why?
        iv. What did you not like about it? Why?
        v. In the {first, second} tool, an email thread has a lifespan of two weeks at first and is bumped down to two days after the first message in the thread is opened.
            1. Do you think this is a good default for expiration timing?
            2. How much control would you want over setting expiration times? Would you like to explicitly decide for each email thread, or would you rather the tool do it all for you?
        vi. What did you think about the red countdown timers? Would you want these kinds of indicators in a tool that expires messages?
        vii. In the {first, second} tool, you could choose to expire an encrypted thread by pushing the "Permanently Expire this Thread" button, or you could wait until the timer ran up before choosing to expire or retain the encrypted thread. Is there

another way you would prefer to manage expired encrypted emails? For example, would you rather have the tool take care of it for you and not ask you?

    viii. Is modal or integrated SLK management better?

    ix. In the {first, second} tool, the encrypted emails on one thread were all protected together and expired together. Is there another way you would prefer to have these expirable encrypted messages protected? For example, would you prefer each message be protected by itself, or would you rather have all encrypted messages from all contacts over a period of time be protected?

    x. Of the two tools you tested, which one did you feel was more secure and why?

    xi. Of the two tools you tested, which one do you trust more and why?

    xii. Without a tool like this, how long do your emails exist in your inbox before they are deleted?

    xiii. How likely are you to use tool A or tool B in the future?

b. Message Permanence Questions

    i. To what degree are you worried about the permanence of your emails and messages on your mobile devices, laptops, desktops, or other devices? Why?

    ii. To what degree are you worried about the permanence of your information on the Internet in general?

    iii. Would you like to use a tool that makes your message unreadable after a certain period of time? Why?

c. General Questions

    i. Do you ever send sensitive information through email? (usernames, passwords, SSN, financial information, other...)

        1. If so:

            a. How often do you send sensitive information and what kind of information do you send?

            b. Do you ever hesitate, to do this for security purposes?

            c. Do you delete the sensitive email after sending it?

        2. If not:

            a. In what situations can you see yourself sending sensitive information?

            b. How often would you expect this to happen?

            c. Would you delete the sensitive email after sending it?

10. Stop the screen recorder. Stop the audio recording.

B.1.5 Participant A Qualtrics Survey

Study Num and Scenario

Which study is this? (Study coordinator answers this question)

○ A

○ B

Study Option 1 Tasks

In this user test, you will be role playing the following scenario:

Today is Monday, April 3rd, 2017. Please make sure the calendar on the desk is set to this date.

Your friend graduated in accounting and you have asked their help in preparing your taxes. They told you that they needed you to email them your last year's tax PIN and your social security number. Since this information is sensitive, you want to protect (encrypt) this information when you send it over email.

You will be asked to send this information using two different secure email systems. In each task, you'll be told which system to use and assigned a new PIN and SSN. After correctly sending the information, your friend will reply to you with a confirmation code that can be used to continue with the study.

Tell the study coordinator that you are ready to begin this task.

System: **${lm://Field/1}**

In this task, you'll be using **${lm://Field/1}**. The system can be found at the following website: ${lm://Field/2}

Please encrypt and send the following information to your friend using ${lm://Field/1}:

SSN: ${lm://Field/3}
PIN: ${lm://Field/4}

Enter the confirmation code provided by your friend.

```
[                                                                    ]
```

Enter the PIN provided by your friend.

```
[                                                                    ]
```

Please tell the study coordinator you are ready for the next task and give the study coordinator your worksheet with the confirmation code and confirmation PIN.

It is the now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar on the desk is set to that date.

It turns out the IRS needs to get some information directly from you so your taxes can be finalized. However, to communicate with the IRS, you need the tax confirmation code and confirmation PIN you received from your friend in the last task. Unfortunately, you shredded and disposed of the paper with this information, because you thought your taxes were all done. Go back to your email to retrieve this information.

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

It is the now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar on the desk is set to that date.

It turns out the IRS needs to get some information directly from you so your taxes can be finalized. However, to communicate with the IRS, you need the tax confirmation code and confirmation PIN you received from your friend in the last task. Unfortunately, you

shredded and disposed of the paper with this information, because you thought your taxes were all done. Go back to your email to retrieve this information. You are 100% certain you won't need this information after retrieving it again. **So, after retrieving the info, you should take the necessary action to make sure this information can never be accessed again.**

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

Enter the confirmation code you retrieved from your email.

```
[                                                                            ]
```

Enter the PIN retrieved from your email.

```
[                                                                            ]
```

It is the now Friday, April 7th, 2017 (another day has passed). Please make sure the calendar on the desk is set to that date.

After working on your computer for several hours, you decide to step outside for some fresh air. As you come back inside your apartment, you notice your roommate leaving your room. Walking into your room, you realize you left your computer unlocked and your email open. Your task is to determine what encrypted information could your roommate could have read if they were snooping through your email.

To complete this task, please feel free to explore your email inbox.

Was your roommate able to read any of your encrypted email?

○ Yes

○ No

○ I'm not sure

Please notify the study coordinator you are finished with this task.

You will now be asked several questions concerning your experience with **${lm://Field/1}**.

Please answer the following questions about **${lm://Field/1}**. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you don't have a response to a particular statement.

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I think that I would like to use this system frequently.	○	○	○	○	○
I found the system unnecessarily complex.	○	○	○	○	○
I thought the system was easy to use.	○	○	○	○	○
I think that I would need the support of a technical person to be able to use this system.	○	○	○	○	○
I found the various functions in this system were well integrated.	○	○	○	○	○
I thought there was too much inconsistency in this system.	○	○	○	○	○

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I would imagine that most people would learn to use this system very quickly.	○	○	○	○	○
I found the system very cumbersome to use.	○	○	○	○	○
I felt very confident using the system.	○	○	○	○	○
I needed to learn a lot of things before I could get going with this system.	○	○	○	○	○

What did you like most about using **${lm://Field/1}**?

What would you change about **${lm://Field/1}**? Why?

Study Option 2 Tasks

In this user test, you will be role playing the following scenario:

Today is Monday, April 3rd, 2017. Please make sure the calendar on the desk is set to this date.

Your friend graduated in accounting and you have asked their help in preparing your taxes. They told you that they needed you to email them your last year's tax PIN and your social security number. Since this information is sensitive, you want to protect (encrypt) this information when you send it over email.

You will be asked to send this information using two different secure email systems. In each task, you'll be told which system to use and assigned a new PIN and SSN. After correctly sending the information, your friend will reply to you with a confirmation code that can be used to continue with the study.

Tell the study coordinator that you are ready to begin this task.

System: **${lm://Field/1}**

In this task, you'll be using **${lm://Field/1}**. The system can be found at the following website: ${lm://Field/2}

Please encrypt and send the following information to your friend using ${lm://Field/1}:
SSN: ${lm://Field/3}
PIN: ${lm://Field/4}

Enter the confirmation code provided by your friend.

Enter the PIN provided by your friend.

Please tell the study coordinator you are ready for the next task and give the study coordinator your worksheet with the confirmation code and confirmation PIN.

It is the now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar on the desk is set to that date.

It turns out the IRS needs to get some information directly from you so your taxes can be finalized. However, to communicate with the IRS, you need the tax confirmation code and confirmation PIN you received from your friend in the last task. Unfortunately, you shredded and disposed of the paper with this information, because you thought your taxes were all done. Go back to your email to retrieve it. You are 100% certain you won't need this information again after retrieving it. **Thus, after retrieving the info, you should this information can never be accessed again.**

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

It is the now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar on the desk is set to that date.

It turns out the IRS needs to get some information directly from you so your taxes can be finalized. However, to communicate with the IRS, you need the tax confirmation code and confirmation PIN you received from your friend in the last task. Unfortunately, you shredded and disposed of the paper with this information, because you thought your taxes were all done. Go back to your email to retrieve it.

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

Enter the confirmation code you retrieved from your email.

```
[                                                                    ]
```

Enter the PIN retrieved from your email.

<div style="border:1px solid #000; height:40px;"></div>

It is the now Friday, April 7th, 2017 (another day has passed). Please make sure the calendar on the desk is set to that date.

After working on your computer for several hours, you decide to step outside for some fresh air. As you come back inside your apartment, you notice your roommate leaving your room. Walking into your room, you realize you left your computer unlocked and your email open. Your task is to determine what encrypted information could your roommate could have read if they were snooping through your email.

To complete this task, please feel free to explore your email inbox.

Was your roommate able to read any of your encrypted email?

○ No
○ Yes
○ I'm not sure

Please notify the study coordinator you are finished with this task.

You will now be asked several questions concerning your experience with **${lm://Field/1}**.

Please answer the following questions about **${lm://Field/1}**. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you don't have a response to a particular statement.

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I think that I would like to use this system frequently.	◯	◯	◯	◯	◯
I found the system unnecessarily complex.	◯	◯	◯	◯	◯
I thought the system was easy to use.	◯	◯	◯	◯	◯
I think that I would need the support of a technical person to be able to use this system.	◯	◯	◯	◯	◯
I found the various functions in this system were well integrated.	◯	◯	◯	◯	◯
I thought there was too much inconsistency in this system.	◯	◯	◯	◯	◯
I would imagine that most people would learn to use this system very quickly.	◯	◯	◯	◯	◯
I found the system very cumbersome to use.	◯	◯	◯	◯	◯
I felt very confident using the system.	◯	◯	◯	◯	◯
I needed to learn a lot of things before I could get going with this system.	◯	◯	◯	◯	◯

What did you like most about using **${lm://Field/1}**?

What would you change about **${lm://Field/1}**? Why?

Comparison

You have finished all the usability tasks for this study. Please answer the following questions about your experience.

Which system was your favorite? *(The systems appear in the order you tested them. Ask the coordinator if you are unclear which system is which.)*

- ◯ First system: ${e://Field/system1}
- ◯ Second system: ${e://Field/system2}
- ◯ I didn't like any of the systems I used

Please explain why.

---

www.manaraa.com

Please answer the following question. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you don't have a response to a particular statement.

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I want to be able to encrypt my email.	○	○	○	○	○
I would encrypt email frequently.	○	○	○	○	○

Demographics

What is your gender?

○ Male

○ Female

○ I prefer not to answer

What is your age?

○ 18 - 24 years old

○ 25 - 34 years old

○ 35 - 44 years old

○ 45 - 54 years old

○ 55 years or older

○ I prefer not to answer

What is the highest degree or level of school you have completed?

○ Some school, no high school diploma

○ High school graduate, diploma or the equivalent (for example: GED)

○ Some college or university credit, no degree

○ College or university degree

○ Graduate Education

○ I prefer not to answer

What is your occupation or major?

_____

How would you rate your level of computer expertise?

○ Beginner

○ Intermediate

○ Advanced

Thank you for your feedback! Please let the study coordinator know you are done with the survey. The coordinator will ask you a set of exit interview questions.

Powered by Qualtrics

B.1.6   Participant A Worksheets

MessageGuard - A.1 Worksheet

In this task, you'll be using **MessageGuard - A**. Go to https://a.messageguard.io and get the tool.

Please encrypt and send the following information to your friend using **MessageGuard – A**:

- SSN: 979-65-3363
- PIN: 1988

Enter the confirmation code provided by your friend:

Enter the PIN provided by your friend:

Once you have received the confirmation code and PIN from your friend, send an email to your friend letting them know you received this information. After you have sent this confirmation email, let the study coordinator know you have finished this task.

MessageGuard - B.1 Worksheet

In this task, you'll be using **MessageGuard - B**. Go to https://b.messageguard.io and get the tool.

Please encrypt and send the following information to your friend using **MessageGuard – B**

- SSN: 264-94-8748
- PIN: 6567

Enter the confirmation code provided by your friend:

Enter the PIN provided by your friend:

Once you have received the confirmation code and PIN from your friend, send an email to your friend letting them know you received this information. After you have sent this confirmation email, let the study coordinator know you have finished this task.

MessageGuard - A.2 Worksheet

In this task, you need to go back to your email to retrieve the confirmation code and confirmation PIN you received from your friend in the last task.

If you cannot retrieve the information, please notify the study coordinator.

Enter the confirmation code you retrieved from your email:

Enter the PIN you retrieved from your email:

Once you have retrieved the information, let the study coordinator know you have finished this task.

MessageGuard - B.2 Worksheet

In this task, you need to go back to your email to retrieve the confirmation code and confirmation PIN you received from your friend in the last task.

If you cannot retrieve the information, please notify the study coordinator.

Enter the confirmation code you retrieved from your email:

Enter the PIN you retrieved from your email:

Once you have retrieved the confirmation code and PIN from your email, take the appropriate action to make sure any emails containing this information are inaccessible. After ensuring emails containing this information are inaccessible, let the study coordinator know you have finished this task.

B.1.7 Participant B Qualtrics Survey

Study Num

Which study is this? (Study coordinator answers this question)

○ A

○ B

Demographics

What is your gender?

○ Male

○ Female

○ I prefer not to answer

What is your age?

○ 18 - 24 years old

○ 25 - 34 years old

○ 35 - 44 years old

○ 45 - 54 years old

○ 55 years or older

○ I prefer not to answer

What is the highest degree or level of school you have completed?

○ Some school, no high school diploma

○ High school graduate, diploma or the equivalent (for example: GED)

○ Some college or university credit, no degree

○ College or university degree

○ Post-Secondary Education

◯ I prefer not to answer

What is your occupation or major?

[   ]

How would you rate your level of computer expertise?

◯ Beginner
◯ Intermediate
◯ Advanced

Study Option 1 Tasks

In this user test, you will be role playing the following scenario:

Today is Monday, April 3rd, 2017. Please make sure the calendar on the desk is set to this date.

You graduated in accounting and have agreed to help a friend prepare their taxes. You have asked them to email you their last year's tax PIN and their social security number.

As part of the study, your friend will send you this information two different times. Each time, after receiving their PIN and SSN, you will be provided with a confirmation code and a PIN number to send to your friend so that both of you can continue with the study.

Tell the study coordinator that you are ready to begin this task.

Please wait for your friend's email with their last year's tax PIN and SSN.

Enter your friend's SSN. Include dashes.

Enter your friend's PIN.

Please tell the study coordinator you are ready for the next task and give the study
coordinator your worksheet containing your friend's SSN and PIN.

It is now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar
on the desk is set to that date.

You thought you had completed your friend's taxes, so you shredded the paper with their
tax information. However, the IRS needs more information about your friend's taxes and
you need to retrieve your friend's SSN and PIN from the email they send earlier. Go to
your email and retrieve this informaiton.

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

It is now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar
on the desk is set to that date.

You thought you had completed your friend's taxes, so you shredded the paper with their
tax information. However, the IRS needs more information about your friend's taxes and
you need to retrieve your friend's SSN and PIN from the email they sent earlier. Go to your
email and retrieve this information. You are 100% certain you won't need this information
after retrieving it again. **So, after retrieving the info, you should take the necessary
action to make sure this information can never be accessed again.**

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

Enter your friend's SSN you retrieved from your email. Include dashes.

[   ]

Enter your friend's PIN you retrieved from your email.

[   ]

It is now Friday, April 7th, 2017 (another day has passed). Please make sure the calendar on the desk is set to that date.

After a long couple hours of work, you decide to grab a snack from the vending machine. On the way back to your cubicle, you notice your co-worker slipping out of your cubicle. Entering your cubicle, you realize you forgot to lock your computer and you left your email open. Find out what encrypted emails your co-worker could have seen.

To complete this task, please feel free to explore your email inbox.

Was your co-worker able to read any of your encrypted email?

○ I'm not sure
○ No
○ Yes

Please notify the study coordinator you are finished with this task.

You will now be asked several questions concerning your experience with **${lm://Field/1}.**

Please answer the following questions about **${lm://Field/1}**. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you don't have a response to a particular statement.

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I think that I would like to use this system frequently.	○	○	○	○	○
I found the system unnecessarily complex.	○	○	○	○	○
I thought the system was easy to use.	○	○	○	○	○
I think that I would need the support of a technical person to be able to use this system.	○	○	○	○	○
I found the various functions in this system were well integrated.	○	○	○	○	○
I thought there was too much inconsistency in this system.	○	○	○	○	○
I would imagine that most people would learn to use this system very quickly.	○	○	○	○	○
I found the system very cumbersome to use.	○	○	○	○	○
I felt very confident using the system.	○	○	○	○	○
I needed to learn a lot of things before I could get going with this system.	○	○	○	○	○

What did you like about using **${lm://Field/1}**?

What would you change about **${lm://Field/1}**? Why?

Study Option 2 Tasks

In this user test, you will be role playing the following scenario:

Today is Monday, April 3rd, 2017. Please make sure the calendar on the desk is set to this date.

You graduated in accounting and have agreed to help a friend prepare their taxes. You have asked them to email you their last year's tax PIN and their social security number.

As part of the study, your friend will send you this information two different times. Each time, after receiving their PIN and SSN, you will be provided with a confirmation code and a PIN number to send to your friend so that both of you can continue with the study.

Tell the study coordinator that you are ready to begin this task.

Please wait for your friend's email with their last year's tax PIN and SSN.

Enter your friend's SSN. Include dashes.

[   ]

Enter your friend's PIN.

[   ]

Please tell the study coordinator you are ready for the next task and give the study coordinator your worksheet containing your friend's SSN and PIN.

It is now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar on the desk is set to that date.

You thought you had completed your friend's taxes, so you shredded the paper with their tax information. However, the IRS needs more information about your friend's taxes and you need to retrieve your friend's SSN and PIN from the email they send earlier. Go to your email and retrieve this informaiton.

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

It is now Thursday, April 6th, 2017 (3 days have passed). Please make sure the calendar on the desk is set to that date.

You thought you had completed your friend's taxes, so you shredded the paper with their tax information. However, the IRS needs more information about your friend's taxes and you need to retrieve your friend's SSN and PIN from the email they sent earlier. Go to your email and retrieve this information. You are 100% certain you won't need this information after retrieving it again. **So, after retrieving the info, you should take the necessary action to make sure this information can never be accessed again.**

If you can't retrieve the information, notify the study coordinator.

Let the study coordinator know when you are ready to begin this task.

Enter your friend's SSN you retrieved from your email. Include dashes.

[   ]

Enter your friend's PIN you retrieved from your email.

[   ]

It is now Friday, April 7th, 2017 (another day has passed). Please make sure the calendar on the desk is set to that date.

After a long couple hours of work, you decide to grab a snack from the vending machine. On the way back to your cubicle, you notice your co-worker slipping out of your cubicle. Entering your cubicle, you realize you forgot to lock your computer and you left your email open. Find out what encrypted emails your co-worker could have seen.

To complete this task, please feel free to explore your email inbox.

Was your co-worker able to read any of your encrypted email?

◯ No
◯ Yes
◯ I'm not sure

Please notify the study coordinator you are finished with this task.

You will now be asked several questions concerning your experience with **${lm://Field/1}.**

Please answer the following questions about **${lm://Field/1}**. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you don't have a response to a particular statement.

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I think that I would like to use this system frequently.	○	○	○	○	○
I found the system unnecessarily complex.	○	○	○	○	○
I thought the system was easy to use.	○	○	○	○	○
I think that I would need the support of a technical person to be able to use this system.	○	○	○	○	○
I found the various functions in this system were well integrated.	○	○	○	○	○
I thought there was too much inconsistency in this system.	○	○	○	○	○
I would imagine that most people would learn to use this system very quickly.	○	○	○	○	○
I found the system very cumbersome to use.	○	○	○	○	○
I felt very confident using the system.	○	○	○	○	○
I needed to learn a lot of things before I could get going with this system.	○	○	○	○	○

What did you like about using **${lm://Field/1}**?

![ ]

What would you change about **${lm://Field/1}**? Why?

![ ]

Comparison

You have finished all of the usability tasks for this study. Please answer the following questions about your experience.

Which system was your favorite? *(The systems appear in the order you tested them. Ask the coordinator if you are unclear which system is which.)*

⚪ First system: ${e://Field/system1}
⚪ Second system: ${e://Field/system2}
⚪ I didn't like any of the systems I used

Please explain why.

Please answer the following questions. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you don't have a response to a particular statement.

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
I want to be able to encrypt my email.	○	○	○	○	○
I would encrypt email frequently.	○	○	○	○	○

Thank you for your feedback! Please let the study coordinator know you are done with the survey. The coordinator will ask you a set of exit interview questions.

Powered by Qualtrics

B.1.8 Participant B Worksheets

MessageGuard - A.1 Worksheet

You have completed your friend's taxes and need to send them the confirmation code and this year's tax PIN from their tax submission.

Since your friend used MessageGuard - A to send sensitive information to you, please also use MessageGuard - A to send them the confirmation code and PIN.

- **Confirmation code: LDDLDHVO**
- **PIN: 3866**

Once you have sent the confirmation code and PIN to your friend, wait for them to reply to you and confirm they got the information. Once you have gotten this confirmation, let the study coordinator know you have finished this task.

MessageGuard - A.2 Worksheet

In this task, you need to go back to your email to retrieve the confirmation code and confirmation PIN you received from your friend in the last task.

If you cannot retrieve the information, please notify the study coordinator.

Enter the SSN you retrieved from your email:

Enter the PIN you retrieved from your email:

Once you have retrieved the information, let the study coordinator know you have finished this task.

MessageGuard - B.2 Worksheet

In this task, you need to go back to your email to retrieve the confirmation code and confirmation PIN you received from your friend in the last task.

If you cannot retrieve the information, please notify the study coordinator.

Enter the SSN you retrieved from your email:

Enter the PIN you retrieved from your email:

Once you have retrieved the confirmation code and PIN from your email, take the appropriate action to make sure any emails containing this information are inaccessible. After ensuring emails containing this information are inaccessible, let the study coordinator know you have finished this task.

MessageGuard - B.2 Worksheet

You have completed your friend's taxes and need to send them the confirmation code and this year's tax PIN from their tax submission.

Since your friend used MessageGuard - B to send sensitive information to you, please also use MessageGuard - B to send them the confirmation code and PIN.

- **Confirmation code: OLOWXTUU**
- **PIN: 8111**

Once you have sent the confirmation code and PIN to your friend, wait for them to reply to you and confirm they got the information. Once you have gotten this confirmation, let the study coordinator know you have finished this task.

General Worksheet

Please wait for your friend's email with their last year's tax PIN and SSN.

Enter your friend's SSN. Include dashes.

```

```

Enter your friend's PIN.

```

```

Once you have written down your friend's SSN and PIN, let the study coordinator know that you are ready to reply to your friend with their confirmation code and PIN.

B.1.9 Exit Interview Questions

1. Tell me about your experience using these tools.

2. What are some things that stood out to you? Why?

3. What did you like about the two tools? Why?

4. What did you not like about them? Why?

5. In the first,second tool, an email thread has a lifespan of 1 month. Do you think this is a good default for expiration timing?

6. How much control would you want over setting expiration times? Would you like to explicitly decide for each email thread, or would you rather the tool do it all for your?

7. What did you think about the red expiration labels? Would you want these kinds of indicators in a tool that expires messages?

8. Of the two tools you tested, which one did you feel was more secure and why?

9. Of the two tools you tested, which one did you trust more and why?

10. Without a tool like this, how long do your emails exist in your inbox before they are deleted?

11. How likely are you to use tool A or tool B in the future?

12. To what degree are you worried about the permanence of your emails and messages on your mobile devices, laptops, desktops, or other devices? Why?

13. To what degree are you worried about the permanence of your information on the Internet in general?

14. Would you like to use a tool that makes your messages unreadable after a certain period of time? Why?

15. Do you ever send sensitive information through email?

16. In the first,second tool, you could choose to expire an encrypted thread by pushing the "Make Unreadable" button, or you could wait until the timer ran up before choosing to expire or retain the encrypted thread.

    (a) Is there another way you would prefer to manage expired encrypted emails?

    (b) For example, would you rather have the tool take care of it for you and not ask you?

17. Did you like the pop up asking you to manage expired messages? If not, how would you like to have it work instead?

18. In the first,second tool, the encrypted emails on one thread were all protected together and expired together.

    (a) Is there another way you would prefer to have these expirable encrypted messages protected?

    (b) For example, would you prefer each message protected by itself, or would you rather have all encrypted messages from all contacts over a period of time be protected?

B.2   Participant Demographics - Extended

The extended demographic data can be seen in Table B.1.

B.3   System Usability Scale

This section of the appendix contains details on the System Usability Scale.

B.3.1   SUS Likert Questions

1. I think that I would like to use this system frequently.

2. I found the system unnecessarily complex.

3. I thought the system was easy to use.

4. I think that I would need the support of a technical person to be able to use this system.

5. I found the various functions in this system were well integrated.

6. I thought there was too much inconsistency in this system.

7. I would imagine that most people would learn to use this system very quickly.

8. I found the system very cumbersome to use.

9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

		Total	%
Gender	Male	22	46%
	Female	26	54%
	Prefer not to answer	0	0%
Age	18–24 years old	33	69%
	25–34 years old	11	23%
	35–44 years old	1	2%
	45–54 years old	2	4%
	55 years or older	1	2%
Education	Some school	0	0%
	High school graduate	2	4%
	Some college	30	63%
	College or university degree	13	27%
	Post-secondary education	3	6%
	Prefer not to answer	0	0%
Computer Expertise	Beginner	8	16%
	Intermediate	32	66%
	Advanced	8	16%
Occupation or Major	Mechanical Engineering	4	9%
	Web Designer	1	2%
	Vocal Performance	2	4%
	Dietetics	2	4%
	Electrical Engineering	2	4%
	Experience Design and Management	1	2%
	Marriage and Family Studies	1	2%
	Computer Engineering	2	4%
	Exercise and Wellness	1	2%
	Advertising	1	2%
	Editor	1	2%
	Chiropractic Practitioner	1	2%
	Music Education	1	2%
	Astronomy	1	2%
	Piano Performance	1	2%
	Student	1	2%
	Child Life Specialist	1	2%
	Studio Art	1	2%
	Pre–Illustration	1	2%
	Civil Engineering	2	4%
	Cosmotologist	1	2%
	Accounting	2	4%
	Computer Science	2	4%
	Biochemistry	2	4%
	History	1	2%
	Linguistics	1	2%
	Biodiversity and Conservation	1	2%
	Education	1	2%
	Bioinformatics	1	2%
	Piano Performance	1	2%
	Mathematics	1	2%
	Psychology	2	4%
	Finance	1	2%
	Secretary	1	2%
	Assessment Librarian	1	2%
	Unknown	1	2%

Table B.1: Participant Demographics Extended

B.3.2   SUS Score Calculation Method

SUS scores are calculated using the following method:

1. For each of the even numbered questions calculate: 5 - <the Likert score of the question>

2. For each of the odd numbered questions calculate: <the Likert score of the question> - 1

3. Add this calculated values together to get TOTAL

4. Multiply TOTAL by 2.5 to get the SUS score

Example

Question	Likert Score	Calculated Score
1	5	4
2	1	4
3	4	3
4	2	3
5	4	3
6	1	4
7	3	2
8	3	2
9	5	4
10	4	1
	Sum Total	30
		x2.5
	SUS Score	75