A Usability Study of Secure Email Deletion

Tyler Monson
Brigham Young University
monson@isrl.byu.edu

Joshua Reynolds
University of Illinois at Urbana-Champaign
joshuar3@illinois.edu

Trevor Smith
Brigham Young University
tsmith@isrl.byu.edu

Scott Ruoti
MIT Lincoln Laboratory
scott@ruoti.org

Daniel Zappala
Brigham Young University
zappala@cs.byu.edu

Kent Seamons
Brigham Young University
seamons@cs.byu.edu

***Abstract*—Messaging applications like SnapChat illustrate that users are concerned about the permanence of information. We find that this concern extends to email. In this paper we present a usability study of an end-to-end secure email tool with the option to securely delete messages. This tool uses ephemeral keys, one per message thread, and default expiration times, with a user prompt to renew or delete keys. Deleting keys causes the messages in the thread to be unreadable for that user. We compare the usability of this tool to a nearly identical tool that uses long term keys and lacks a feature to expire keys. We also interview participants about their email use patterns and attitudes towards information permanence. We find that participants are especially interested in the ability to control the lifetime of an email message. Participants also report trusting the tool that allowed them to make their email messages ephemeral more than the tool that just encrypted their email.**

I. Introduction

Users have regularly expressed fears about information permanence, arising from the uneasiness of personal information disclosed online [11], [16], [19], [21]. This permanence can affect friendships, business reputations, job opportunities, and more. The popularity of messaging applications that support self-destructing messages, such as Snapchat, suggests that reliable deletion of messages containing personal information is important to users. We hypothesize that this desire for reliable deletion extends to email as well.

With current, plaintext email systems, all parties can delete their copy of the message. However, this approach is problematic because neither the sender nor receiver can ensure that their respective mail providers delete all copies of their messages. With secure email systems that are based on end-to-end encryption, each party can again delete their copy of the message, and any extra copies kept by a mail provider or eavesdropper cannot be read by the third party. However, if the system uses long-term encryption keys, a compromise of the key means all old messages can be decrypted.

We study a third approach, in which users encrypt their communications with per-message-thread, ephemeral (i.e., short-lived) keys and delete those keys when they want the thread to expire. This approach is consistent with the Signal protocol [13] implemented by the Signal, Facebook, and WhatsApp messaging applications. In this paper we explore the usability of this approach when applied to email. We evaluate two secure email prototypes, one that uses long-term keys, and one that uses deletable, short-lived keys. We built these prototypes using the MessageGuard framework [15] because it allows us to easily implement new key management schemes while maintaining a consistent user interface.

Using these prototypes, we conducted a within-subjects, paired-participant [14] user study with a total of 24 participant pairs (48 participants total). In this study, participants worked with their partners to set up each tool and begin sending secure email. Participants also completed tasks related to deleting their secure messages. After completing these tasks for both systems, participants 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 regarding messages on their devices and the Internet in general.

Quantitative results from this study show that users preferred the short-lived keys prototype. Qualitative feedback makes it clear that this preference was driven by the ability to control message permanence. Additionally, participant responses show that most participants send sensitive information through email and that most are also interested in technology that provides message ephemerality.

The contributions of our work are as follows:

1) The first usability study of a secure email prototype containing an explicit deletion feature that renders a message unreadable.
2) Strong qualitative feedback demonstrating that users want to be able to control the lifespan of email messages containing sensitive information.
3) Qualitative user feedback on a range of topics, including message expiration preferences, interest in secure email generally and email ephemerality specifically, and user trust in secure email software.
4) Quantitative evidence demonstrating that paired participant studies are able to elicit different experiences for each user role.

Artifacts: A companion website at https://isrl.byu.edu/data/eurousec2018/ provides study materials, including survey questions, interview guide, and participant responses.

European Workshop on Usable Security (EuroUSEC) 2018
23 April 2018, London, England
ISBN 1-891562-54-1
https://dx.doi.org/10.14722/eurousec.2018.23018

II. Related Work

Formal user studies of secure email began in 1999 when Whitten and Tygar [20] conducted a user study of PGP 5.0, revealing that users struggled with key management. Later, Garfinkel and Miller [8] conducted a modified version of that first study with a secure email prototype based on S/MIME. Results from the study showed improvements can be made by applying automated key generation, key management, and message signing.

Bai et al. [1] conducted a user study evaluating the usability and security trade-offs of a key directory model and a key-exchange model. They found that users considered the key directory model "good enough" even though the key exchange model has stronger security guarantees.

Ruoti et al. [14] introduced a paired-participant methodology that include two users completing a secure email task together to better simulate grassroots adoption scenarios. Later, Ruoti et al. [15] created MessageGuard, a framework supporting a pluggable key management scheme, that they used to compare three secure email prototypes that differed only in their key management schemes. Using MessageGuard as a base for the three prototypes removed confounding factors so that the usability of the key management schemes themselves could be directly compared.

A. Short-Lived Keys and Forward Secrecy

Brown et al. [5] as well as Schneier and Hall [17] have suggested that forward secrecy should be obtained by using different keys for every encrypted message. Brown et al. discuss one potential complication for short-lived keys—worse usability. They present this complication as a cost tradeoff between security and key distribution. In this paper, we test the usability of a short-lived key prototype similar to what was suggested by Brown et al. and measure its usability. Contrary to Brown et al.'s supposition, our findings show that participants viewed such a tool as highly usable.

Boneh and Franklin [3] describe a variation of Identity-Based Encryption (IBE) that supports short-lived keys. The user's identity string (pubic key) could consist of an email address combined with a date string so that a new key pair be required each day to decrypt new messages. In IBE, keys can never be permanently deleted as long as the IBE server is in operation.

Off-the-Record Communication (OTR), outlined by Borisov et al. [4], uses short-lived keys to obtain confidentiality, perfect forward secrecy, and repudiation for instant messaging. A prototype of OTR was implemented, but 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.

Green and Miers [9] introduced 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. It also does not require redistribution of public keys after the decryption key is updated.

B. Permanence of Personal Information

Odom et al. [12] showed that users desire both permanence and ephemerality for different digital possessions. As an example, Cecchinato et al. [6] show that email users frequently use email archives to retrieve important information. Others have proposed expiration dates for email [18] and ephemerality for email [10].

Several studies reveal that Internet users are concerned about their information being accessible on the Internet. Ruoti et al. [16] 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. [11], individuals expressed concerns about modern technology making public records readily available. Work by Wang et al. [19] shows social media users are worried about unintended audiences seeing their posts, which may lead to job loss or relationship complications. Finally, Woodruff [21] conducted a "qualitative study of how users manage their reputations online," showing 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."

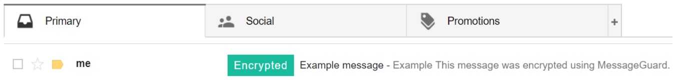
III. Prototypes

In order to compare email systems with and without the ability to delete short-lived keys, we developed two prototypes, one that supported long-term keys and another supporting short-lived keys, referred to as the LTK and SLK prototypes throughout the remainder of the paper. The LTK prototype serves as a baseline for traditional secure email supporting long-term keys.

The prototypes are browser extensions implemented using MessageGuard [15], a platform that was developed to enable quick prototyping of secure email tools that tightly integrate with Gmail. The platform separates key management functionality from the user interface, making it possible to easily add and compare new key management schemes. This architecture also allows for a mostly common user interface across different prototypes, which reduces confounding factors. Developing our prototypes with MessageGuard also provided the benefits of unencrypted greetings in encrypted emails, tools to create inline tutorials, a workflow to bootstrap communication with new users, and a framework for working with data packages included in emails sent with MessageGuard.

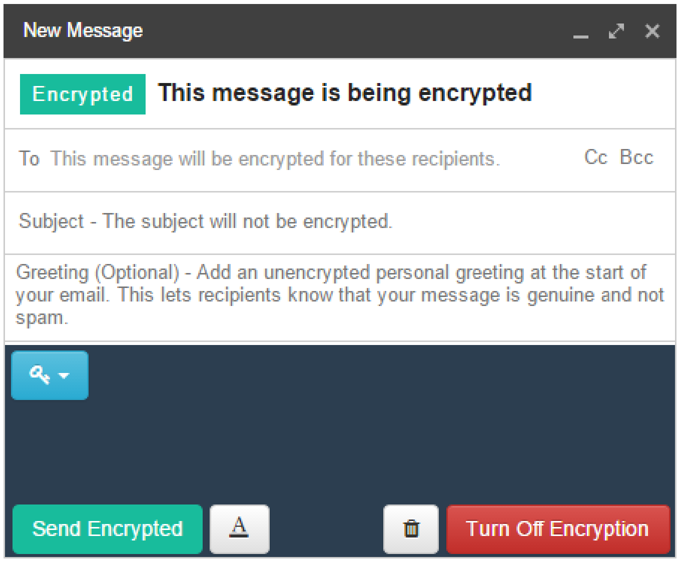
Both prototypes were designed to use Pretty Good Privacy (PGP) [7], [22] as their underlying cryptographic system. PGP is an end-to-end encryption system based on public key cryptography that allows users to both encrypt and sign their messages.

Rather than rely on a model that uses a public key server [1], the prototypes establish short-lived and long-term keys exchanged via email. Figure 1 shows the user interface



me Encrypted Example message - Example This message was encrypted using MessageGuard.

Fig. 1: Labeling encrypted threads.



Encrypted This message is being encrypted

To This message will be encrypted for these recipients. Cc Bcc

Subject - The subject will not be encrypted.

Greeting (Optional) - Add an unencrypted personal greeting at the start of your email. This lets recipients know that your message is genuine and not spam.

Fig. 2: Composing an encrypted message.

when receiving an encrypted message in the Gmail inbox, and Figure 2 shows the user interface for composing an encrypted email. The pros and cons for the LTK and SLK prototypes are discussed in the remainder of this section, and are summarized in Table I.

A. Short-Lived Keys

Suppose Bob wants to send a sensitive message to Alice. For the short-lived keys (SLK) prototype, they exchange keys as follows:

- Assuming Bob has already installed the extension, he turns on encryption, composes his message, then clicks "Send Encrypted". While Bob composes his message, the system automatically generates a new public/private key pair for Bob that is unique to this thread with Alice. When the message is sent, the system attaches Bob's public key to the email containing the encrypted message. The message is encrypted with a random symmetric key that is stored in Bob's browser and associated with the thread.
- Once Alice installs the extension and has the email from Bob open, that extension automatically generates a new public/private key pair for Alice for this thread. The extension also stores Bob's public key for later use.
- Alice must click a "Send Access Request" button to request access to the encrypted message, as shown in Figure 3a. Note, we require manual action here to allow Alice to decide not to decrypt the email and exchange keys, for example, in case the email is spam.

TABLE I: LTK vs. SLK Comparison

	LTK	SLK
Key Setup	Long-term keys are created when the software is installed.	Short-lived keys are created for each message thread.
Deletion	Users must remember to delete individual messages and remove them from the trash.	Users are prompted to delete or renew decryption keys at the end of their lifetime.

  Once Alice clicks the "Send Access Request" button, the extension automatically sends an email to Bob requesting access to the message. Among other data, this email contains Alice's public key.

- Once Bob opens Alice's access request email shown in Figure 3b, his extension stores Alice's public key and uses it to encrypt the symmetric key that was used to encrypt his original message. The extension automatically sends the encrypted symmetric key to Alice in a reply email.
- Alice must open the thread containing the access response shown in Figure 3c. The response includes the encrypted symmetric key. The extension then extracts and stores the encrypted symmetric key and decrypts the original message from Bob.

We did not implement a key exchange process for the situation where Alice wants to request sensitive information from Bob and send a public key at the same time, though one could be similarly designed.

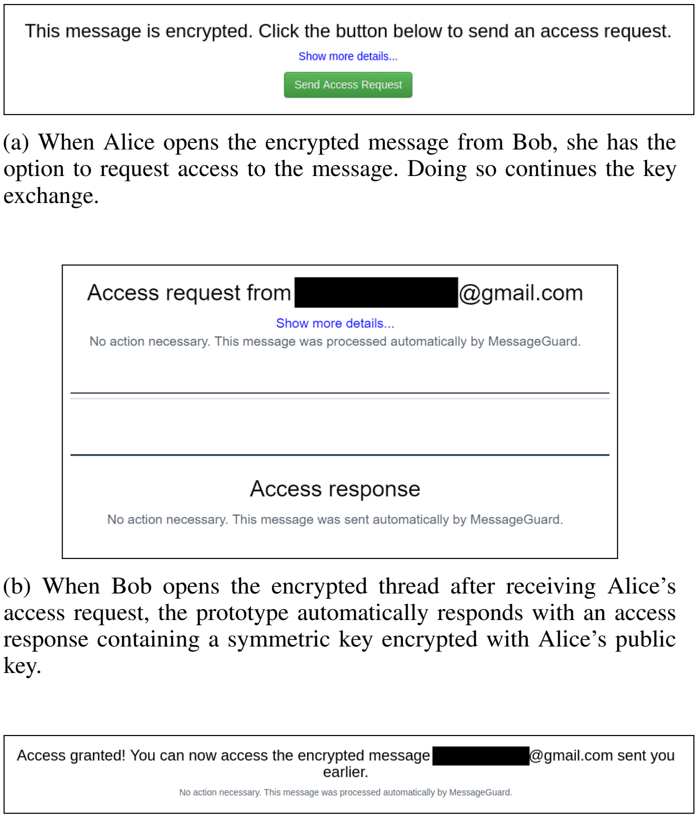
Short-lived keys have a default lifespan of 30 days. When a key's expiration timer expires, users are warned that it is time to make the message thread unreadable. Users can then choose either to delete the message key or postpone their decision by adding two more days to the expiration timer. Users can also choose to destroy their short-lived keys at any time before the 30 days have passed.

Users are not presented with the concept of keys at any time. Instead, they are told that message threads are expiring. Figure 4a shows how threads are labeled with their expiration date. The user can choose to make the thread unreadable immediately (which will delete the associated keys) using the "Make Unreadable" button. Once a short-lived key pair is destroyed, encrypted messages are overlayed to show they can no longer be accessed, as shown in Figure 4b. Messages are likewise labeled with expiration dates, or shown as unreadable, in the main inbox view.

In addition to this interface, users are reminded to manage expired threads through a popup. Figure 5 shows an example of the popup that users encounter once one or more of their keys expire. Users must choose one of the two options for each thread listed before the popup disappears.

B. Long-Term Keys

The long-term key (LTK) prototype generates only a single public/private key pair for each user during the setup phase of the extension. For example, Alice, Jane, and Johnny will all use the same public key for Bob to encrypt the symmetric keys that


This message is encrypted. Click the button below to send an access request.

(a) When Alice opens the encrypted message from Bob, she has the option to request access to the message. Doing so continues the key exchange.

Access request from [redacted]@gmail.com

No action necessary. This message was processed automatically by MessageGuard.

Access response

No action necessary. This message was sent automatically by MessageGuard.

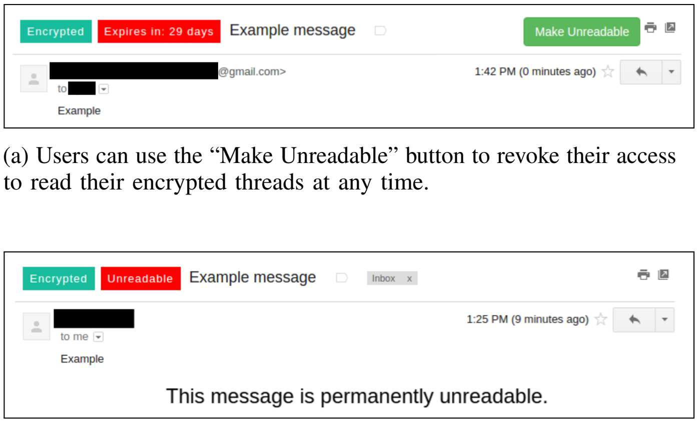
(b) When Bob opens the encrypted thread after receiving Alice's access request, the prototype automatically responds with an access response containing a symmetric key encrypted with Alice's public key.

Access granted! You can now access the encrypted message [redacted]@gmail.com sent you earlier.

No action necessary. This message was processed automatically by MessageGuard.

(c) Once Alice opens the thread after receiving Bob's access response, the prototype extracts the encrypted symmetric key, stores it, and decrypts the original message.

Fig. 3: Exchanging keys.



[redacted]@gmail.com>

1:42 PM (0 minutes ago)

to [redacted]

Example

(a) Users can use the "Make Unreadable" button to revoke their access to read their encrypted threads at any time.

[redacted]

1:25 PM (9 minutes ago)

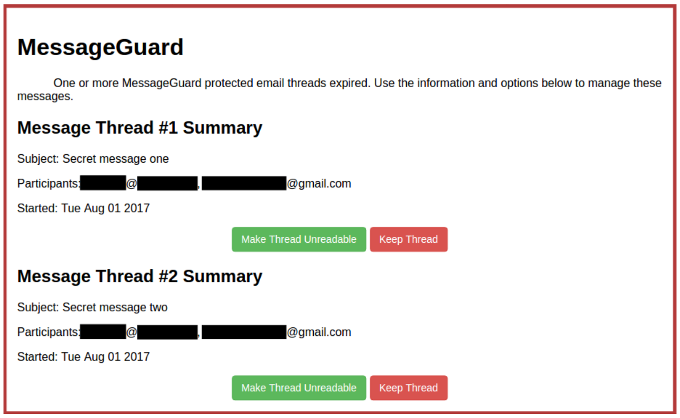
to me

Example

This message is permanently unreadable.

(b) After using the "Make Unreadable" button, messages on encrypted threads are permanently unreadable.

Fig. 4: Making messages unreadable.


MessageGuard

One or more MessageGuard protected email threads expired. Use the information and options below to manage these messages.

Message Thread #1 Summary

Subject: Secret message one

Participants:[redacted]@[redacted],[redacted]@gmail.com

Started: Tue Aug 01 2017

Message Thread #2 Summary

Subject: Secret message two

Participants:[redacted]@[redacted],[redacted]@gmail.com

Started: Tue Aug 01 2017

Fig. 5: A popup helping users to manage expired email threads.

protect the information they send to him. Keys are exchanged, as described above, but only the first time a pair of users start a secure conversation. Subsequent encrypted threads re-use the same long-term keypair for each user. A user's long-term key pair is stored in encrypted local storage until the prototype is uninstalled.

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 address. The prototype is limited to only working with the email account related to the address provided by the user during setup. Future work on this prototype could expand the capabilities of the prototype to work with multiple email accounts.

IV. 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 methodology from Ruoti et al. [14]. 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. Results from 6 participants pairs (12 participants) had to be excluded from our data analysis, due to technical issues with the software (4 pairs), mistakes by a study coordinator (1 pair), and ineligibility due to age (1 pair). All rejected participants still received compensation. We present results from 24 pairs (48 total participants).

Participants for this user study were Gmail users recruited from the Brigham Young University campus. 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 after another study with a similar recruitment procedure filled up. To participate in this study, participants were required to have an active Gmail account because the prototypes integrate with the Gmail interface. Each participant was compensated $15 USD. The user studies were approximately 50–60 minutes in duration.

A. *Demographics*

Table II contains 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

TABLE II: Participant Demographics

		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%

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

B. User Study Procedure

Participants first received a warning that the prototypes were research software and should not be trusted with truly sensitive information. A coin toss was used to assign the roles of participants A and B. Coordinators showed participants how to use the left monitor for the instructions and survey questions while using the right monitor for study tasks. The first task was to complete a demographic survey.

The scenario described in the next subsection was repeated once with each tool in a randomized order. Study coordinators did not assist participants in completing the tasks. After each completion of the scenario using each tool, participants completed a SUS questionnaire and gave free response comments on their likes and dislikes of the system.

After participants completed the scenario and survey 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: (a) I want to be able to encrypt my email, (b) I would encrypt my email frequently.

At the end of the study, each coordinator conducted a 10–15 minute semi-structured exit interview (see Appendix B). These interviews explored participants' experiences, exploring differences between SLK and LTK. Participants were also prompted to share their opinions regarding short-lived keys and message permanence. Finally, participants were asked questions related to SLK's user interface design.

C. Scenario

Participant B played the role of a tax accountant. Participant A played the role of their client who wants to start using the prototype tool for secure email.

Phase 1: Participant A was first given a link to install the prototype tool and instructed to use it to securely email a fake social security number and PIN to her "tax accountant." Participant B received the email, installed the tool, and initiated an access request (under the hood this is the asymmetric key exchange). Participants who were hesitant to install the extension were told that it was safe to do so on the study computer. Participant A granted access and Participant B decrypted the fake social security number and PIN. Participant B is given a fake confirmation code of the completed tax documents to securely send back to Participant A.

Phase 2: Participants are told that 31 days have passed. Participants flip through a desktop calendar to a date 31 days in the future to simulate the passage of time. If the short-lived key tool is in use, study coordinators click a button in the tool to trigger the expiration of keys and close the Gmail tab. Participants observed this action and were told it was part of simulating the passage of time.

Both participants are asked to go back to their previous conversation and extract the confidential information they were sent. After either obtaining this information or deciding that it was impossible, participants were told that they would never need the information again. They should take the necessary steps to make sure no one can read the email ever again.

Phase 3: Participants now imagine they have a nosy roommate or office mate. Returning after leaving their computer unlocked, they suspect this person of snooping through their email. Participants are asked to look through their inbox to find what information the snooper may have been able to glean from the secret tax conversation.

D. Qualitative Data Analysis

The audio from participants' interviews was recording and then transcribed for review. Two coders worked together to extract codes from the audio recordings and transcriptions. Whenever there was a disagreement regarding a code, the two coders would review the material and reach consensus on the correct coding, thereby reaching perfect agreement.

E. Limitations

We only asked participants to communicate with the tool with one other person. We also asked them to do this out of context of their real life. Our efforts to simulate the passage of time with a calendar could only go so far in the context of an in-laboratory experiment. A longitudinal study with a larger network of participants might reveal additional insights.

The partially automated key exchange for the SLK prototype was aided by both participants being online during the laboratory study to help reduce the delay between messages. The exchange would still work if participants were not online together, it would just take more time. The study did not explore whether users would be willing to tolerate these delays in practice when the sender and recipient are not online together.

TABLE III: SUS Scores

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

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

TABLE V: Participants' favorite prototypes.

	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%)

Our study also has limitations inherent in our population. Our population is not representative of all groups, and future research could broaden the population (e.g., non-students, non-Gmail users). Since the study was conducted in a laboratory, participants may not behave the same as they would in the real world.

V. Quantitative Results

A. *SUS Scores*

We used the System Usability Scale to evaluate the usability of the two prototypes. Table III shows the breakdown of the scores given to each prototype. Overall, short-lived key prototype (SLK) received a mean SUS score of 73.3, while the long-term key prototype (LTK) received a mean SUS score of 66.8. According to the contextual scales developed by Bangor et al. [2], both prototypes are rated as having Good usability. The difference in the short-lived key prototype's SUS score and the long-term key prototype's SUS score is statistically significant (two-tailed student t-test, equal variance, $p < 0.005$).

Despite finding a significant difference in the SUS scores of the two prototypes, we believe that this difference is not indicative of user preference for short-lived keys, but rather a preference for being able to easily delete secure emails. Having an obvious, easy-to-use option for making messages inaccessible in the SLK prototype and having no "Make Unreadable" functionality in the LTK prototype is likely to have been a sharp contrast in usability from the perspective of the participants that tested the prototypes, particularly if they use the LTK prototype second. For example, we observed several participants give up on making their messages 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.

We therefore analyzed the SUS data to determine how tool ordering affected scores. The mean SUS scores for each prototype based on their test order are provided in Table IV. 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, $p < 0.05$). 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, $p = 0.065$).

Another issue related to system scores is whether participants A and B were having similar experiences. We calculated several correlations for the SUS scores: (1) between scores given by paired participants for the same prototype, and (2) between the scores the same participant gave for each prototype. Linear regressions are shown in Appendix A. There was little correlation in the first case (Pearson product-moment correlation coefficients: 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. There was a moderate correlation in the second case (Pearson product-moment correlation coefficient: 0.624), suggesting that participants were fairly consistent in their ratings of the two prototypes. This provides evidence that participants A and B were having different experiences.

B. *Favorite System*

After completing all tasks for both prototypes, participants were asked to choose which of the two prototypes was their favorite. Table V shows these results. SLK received more support from participants with thirty-five participants (35; 72%) choosing it as their favorite. LTK received twelve votes (12; 25%) as the favorite, and one participant (1; 2%) indicated they didn't like either of the prototypes. The proportion of participants that chose SLK as their favorite prototype was statistically significant (Test for one proportion, null hypothesis 50%, observed proportion 75%, population 47, $p < 0.001$).

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 fifteen (15; 63%) of the twenty-four participant pairs agreed on their favorite prototypes. This is more evidence that participants A and B are having different experiences.

C. *Mistakes*

We defined two mistakes participants could make as they worked through their study tasks:

- *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. Each participant has the

TABLE VI: A summary of mistakes made by participants in our user study.

		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

chance to make two mistakes for this category, one for each prototype.

- *Incorrect Snooper Question Answer:* During Phase 3, when participants were given a scenario where they suspected someone snooping on their computer, we asked if they could determine whether the snooper could have seen any of their encrypted email. We counted incorrect answers to this question as a mistake. For example, it would be a mistake to say that the snooper could not see any encrypted messages, even though at least one of the encrypted messages wasn't deleted. 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.

As Table VI shows, 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.

VI. Qualitative Results

In addition to quantitative measurements, we also gathered qualitative feedback about both systems. This feedback helps explain the participants' preference for SLK.

A. Security and Trust

While every participant felt that the tools they used were secure, most (37; 77%) felt that SLK was more secure and trustworthy. Of these participants, most (26; 70%) indicated that the ability to make message unreadable was why SLK was better than LTK. P19B and P23B reported, respectively,

> *"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 [SLK]. 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 [LTK], I couldn't remember how to secure it."*

B. Message Expiration Feedback

Participants provided various feedback about message expiration in SLK.

1) Expiration Timing: In the SLK prototype, messages were set to expire after 30 days by default. Half of the participants felt that this was a good default (26; 54%), while most other users felt it should be longer (20; 41%). Still, nearly all participants (45; 93%) indicated that they wanted the ability to override this default and set their own expiration timespan. In particular, thirty-one participants (31; 64%) noted that the expiration time should take into account both the sensitivity of the information and who the recipient is. On these topics, participants P17B and P22A shared, respectively,

> *"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."*

2) Expiration Labels: Participants felt that SLK's expiration labels were effective, with twenty-two participants (22; 45%) saying expiring messages clearly stood out in their inbox, eleven (11; 22%) stating that they were helpful, and four (4; 8%) indicating they were easy to understand. P10B expressed the following sentiment regarding these labels:

> *"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?"*

3) Automating Message Destruction: SLK does not automatically expire messages, but rather prompts users to choose between deleting or extending the message lifetime after it expires. Half of participants (27; 56%) were OK with this two-step process. Still, ten (10; 20%) participants indicated that they would prefer for messages to auto-expire. For example, 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."*

In contrast, eleven participants (11; 22%) explicitly stated that they didn't think messages should automatically be destroyed after expiration. 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 … "*

Participants were generally cognizant of this trade-off between availability and privacy. As expressed by 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."*

4) Misconceptions: Participants had misconceptions regarding SLK's functionality. Some believed that after message destruction it was possible to restore access to that message by requesting it from their friend. Additionally, several participants believed that when they destroyed a message it would also be inaccessible to the other user. While misconceptions, these comments suggest features that users might be interested in.

C. Interests in Secure Email

Participants shared their viewpoints on the need for secure email.

1) Sending Sensitive Emails: During the exit interview, participants were also asked about whether they send sensitive information through email. Overall, twelve (12; 25%) participants said they do send sensitive information through email, sixteen (16; 33%) said they do so occasionally, and twenty (20; 41%) said they do not. The types of sensitive information sent over email varied: account credentials (7; 14%), tax data (5; 10%), insurance documents (5; 10%), school data (4; 8%), banking information (4; 8%), and credit card numbers (3; 6%).

Several participants indicated that they would use alternative communication technologies to transfer sensitive information: phone call (10; 20%), text message (6; 12%), in person (5; 10%). P23A indicated that they send sensitive information as a picture instead of as text:

> *"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."*

2) Interested in Encrypted Email: Similar to results from Ruoti et al. [14], we found that most participants wanted the ability to encrypt their email, but indicated that they were unlikely to use this functionality frequently. In general, possible uses for secure email was speculative at best. For example, P13B and P17B expressed, respectively,

> *"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 … "*

3) Interest in Message Ephemerality: We asked participants a number of questions about email permanence, the permanence of information on their personal devices, and permanence of information on the Internet in general. Most participants indicated that they store emails in their inbox permanently, or "a long time", or only rarely delete emails. They were generally unconcerned about the permanence of information stored on their devices, but more concerned about the permanence of information on the Internet.

When asked about whether they would want to use a tool that makes messages unreadable after a certain period of time (e.g., SLK), thirty-five participants (35; 72%) answered in the affirmative, five (5; 10%) responded negatively, and (3; 6%) expressed indifference to the idea. In favor of such tools, P20A expressed,

> *"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 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 (7; 14%) explained they would want to use the tool only if they had the need, and (6; 12%) said they liked having the option available. In this regard, P24B shared,

> *"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."*

D. Other Feedback

Participants were asked to report what features they liked about each prototype. Twenty-nine participants (29; 60%) indicated that they both prototypes were user friendly and twenty-three (23; 47%) said that they were straightforward and simple. Ten participants (10; 20%) said that the prototypes were easy to setup and begin using, with nine (9; 18%) stating that the contextual, inline tutorials were especially helpful. For example, P4B said,

> *"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."*

Three participants (3; 6%) felt that both prototypes would be improved by adding a master password to keep unwanted eyes off encrypted email. Another participant, P20B, expressed a desire to have the access requests sent as a push notification on their phone:

> *"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."*

Many participants felt that being truly safe on the Internet was difficult or impossible. Participant P22B summed up these feelings, saying,

> *"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."*

Several participants expressed concerns about knowing whether a secure email tool is actually secure. P10B and P11A shared,

> *"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."*

VII. Conclusion

Both the quantitative and qualitative results show the participants are strongly interested in the ability to limit the lifespan of their sensitive messages. This aligns with previous work [16], indicating that more attention should be paid to this aspect of secure email. Our results also suggests that there are areas that need additional research: (1) allowing and guiding participants in setting custom message lifespans, (2) allowing messages to be destroyed for both senders and receivers if any of those parties want to, (3) allowing users to regain access to destroyed messages with the help of other users with access to the message, and (4) reducing the numbers of mistakes people make when attempting to destroy their messages.

There are also a variety of issues related to message deletion that future studies can explore. First, nothing prevents a user from making a plaintext copy of a sensitive message and storing it somewhere outside the secure email system. For instance, a user can take a picture of a message on the screen or cut and paste the text from an email message. It isn't clear from our study whether users were aware of this possibility, nor whether learning about this would change their attitude about the risks of storing sensitive information in an encrypted email. Second, another approach to handling sensitive email messages is to automatically extract the sensitive information from an outbound email message, store the data on a server, and then replace the data in the message with a link to the server that the recipient can use to access the data. When the sender's organization operates the server, the sensitive data is never stored at the recipient's email provider. The sender can control when to delete the data so that the recipient can not access it on the server. This architecture has not been considered in any usability studies. Third, little is known regarding what kinds of emails people want to be ephemeral, nor how people make that decision, and how this affects the design of secure email systems.

Finally, the quantitative results in our study show a measurable difference in user experiences, and in particular, the system usability scale scores, depending on the role a user plays in a study. We are the first study using paired participants to calculate a linear regression, which showed that each user in the pair has a different experience, providing quantitative evidence of the value of the paired-participant methodology. Based on these results, and similar benefits found in the original two-person secure email study [14], we recommend that most—if not all—laboratory studies of secure email employ a two-person methodology. At worst, such an approach would double the number of study participants, but as our study shows it is also likely to give a more holistic evaluation of a secure email tool's usability.

Acknowledgment

We thank the anonymous reviewers for their suggestions that helped improve the paper. This work was supported in part by the National Science Foundation under Grant No. CNS-1528022.

References

[1] W. Bai, M. Namara, Y. Qian, P. G. Kelley, M. L. Mazurek, and D. 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)*. USENIX Association, 2016, pp. 113–130.

[2] A. Bangor, P. Kortum, and J. Miller, “Determining what individual SUS scores mean: Adding an adjective rating scale,” *Journal of Usability Studies*, vol. 4, no. 3, pp. 114–123, 2009.

[3] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[4] N. Borisov, I. Goldberg, and E. Brewer, “Off-the-record communication, or, why not to use PGP,” in *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*. ACM, 2004, pp. 77–84.

[5] I. Brown and B. Laurie, “Security against compelled disclosure,” in *Proceedings of the 16th Annual Computer Security Applications Conference*. IEEE, 2000, pp. 2–10.

[6] M. E. Cecchinato, A. Sellen, M. Shokouhi, and G. Smyth, “Finding email in a multi-account, multi-device world,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 1200–1210.

[7] S. Garfinkel, *PGP: pretty good privacy*. O’Reilly Media, Inc., 1995.

[8] S. L. Garfinkel and R. 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)*. ACM, 2005, pp. 13–24.

[9] M. D. Green and I. Miers, “Forward secure asynchronous messaging from puncturable encryption,” in *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 305–320.

[10] J. 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*. ACM, 2000, pp. 163–164.

[11] S. A. Munson, D. Avrahami, S. Consolvo, J. Fogarty, B. Friedman, and I. 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*. New York, NY, USA: ACM, 2011, pp. 2–9.

[12] W. Odom, A. Sellen, R. Harper, and E. Thereska, “Lost in translation: understanding the possession of digital things in the cloud,” in *Proceedings of the 2012 CHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 781–790.

[13] T. Perrin and M. Marlinspike, “The double ratchet algorithm,” https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf, 2016.

[14] S. Ruoti, J. Andersen, S. Heidbrink, M. O’Neill, E. Vaziripour, J. Wu, D. Zappala, and K. 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*. ACM, 2016, pp. 4298–4308.

[15] S. Ruoti, J. Andersen, T. Monson, D. Zappala, and K. Seamons, “MessageGuard: A browser-based platform for usable, content-based encryption research,” 2016, arXiv preprint arXiv:1510.08943.

[16] S. Ruoti, T. Monson, J. Wu, D. Zappala, and K. Seamons, “Weighing context and trade-offs: How suburban adults selected their online security posture,” in *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. USENIX, 2017, pp. 211–228.

[17] B. Schneier and C. Hall, “An improved e-mail security protocol,” in *Proceedings of the 13th Annual Computer Security Applications Conference*. IEEE, 1997, pp. 227–230.

[18] M. 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.

[19] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. 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)*. ACM, 2011, pp. 10:1–10:16.

[20] A. Whitten and J. Tygar, “Why Johnny can’t encrypt: A usability evaluation of PGP 5.0,” in *Proceedings of the 8th USENIX Security Symposium (USENIX Security 1999)*. Washington, D.C.: USENIX Association, 1999, pp. 14–28.

[21] A. Woodruff, “Necessary, unpleasant, and disempowering: Reputation management in the Internet age,” in *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 149–158.

[22] P. Zimmermann, “Building in big brother,” L. J. Hoffman, Ed. New York, NY, USA: Springer-Verlag New York, Inc., 1995, ch. Pretty Good Privacy: Public Key Encryption for the Masses, pp. 93–107, last accessed 24 October 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=212412.212422

Appendix A
Linear Regressions on SUS Scores

We calculated several correlations for the SUS scores: (1) between scores given by paired participants for the same prototype, and (2) between the scores the same participant gave for each prototype.

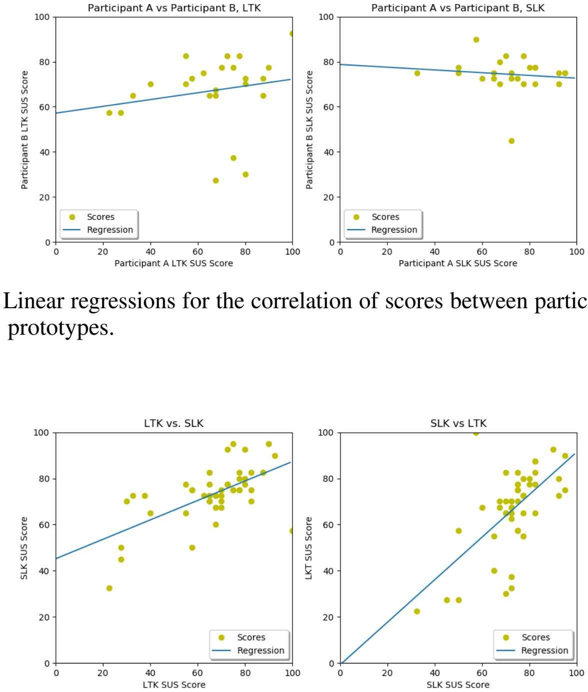
Figure 6a displays the two linear regressions for the scores of each participant pair for the same prototype. There was little correlation with these scores (Pearson product-moment correlation coefficients [1]: LTK — 0.188, SLK — -0.119), suggesting that participants playing role A are having different experiences of these prototypes than the participants playing role B are.

Figure 6b displays two linear regressions for these scores (the second is the inverse of the first). There was a moderate correlation between how participants each rated LTK and SLK (Pearson product-moment correlation coefficient — 0.624). 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.

Appendix B
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?

[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.



(a) Linear regressions for the correlation of scores between participants for prototypes.

(b) Linear regressions for the correlation of scores between prototypes given by participants.

Fig. 6: Linear regressions.

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?