Leading Johnny to Water: Designing for Usability and Trust

Erinn Atwater, Cecylia Bocovich, Urs Hengartner, Ed Lank, Ian Goldberg
Cheriton School of Computer Science
University of Waterloo
{erinn.atwater, cbocovic, urs.hengartner, lank, iang}@uwaterloo.ca

ABSTRACT

Although the means and the motivation for securing private messages and emails with strong end-to-end encryption exist, we have yet to see the widespread adoption of existing implementations. Previous studies have suggested that this is due to the lack of usability and understanding of existing systems such as PGP. A recent study by Ruoti et al. suggested that transparent, standalone encryption software that shows ciphertext and allows users to manually participate in the encryption process is more trustworthy than integrated, opaque software and just as usable.

In this work, we critically examine this suggestion by revisiting their study, deliberately investigating the effect of integration and transparency on users' trust. We also implement systems that adhere to the OpenPGP standard and use end-to-end encryption without reliance on third-party key escrow servers.

We find that while approximately a third of users do in fact trust standalone encryption applications more than browser extensions that integrate into their webmail client, it is not due to being able to see and interact with ciphertext. Rather, we find that users hold a belief that desktop applications are less likely to transmit their personal messages back to the developer of the software. We also find that despite this trust difference, users still overwhelmingly prefer integrated encryption software, due to the enhanced user experience it provides. Finally, we provide a set of design principles to guide the development of future consumer-friendly end-to-end encryption tools.

1. INTRODUCTION

Despite recent revelations of mass government surveillance of the Internet [11] and more than 20 years of availability of end-to-end (E2E) encryption for email, we have yet to see the widespread adoption of encrypted email tools by the general population. Usability is undoubtedly one issue blocking adoption; since the 1999 paper "Why Johnny Can't Encrypt" by Whitten and Tygar [21], E2E encryption tools have been repeatedly criticized for their significant usability issues [4, 10, 18]. Alongside usability, Renaud et al. [16] identified a number of additional issues including awareness, concern, understanding, and knowledge of privacy that must be overcome before privacy-enhancing software is widely adopted.

While Renaud et al. argue that these problems must be solved with user education, we believe they instead show that E2E encryption must be integrated into email clients by *default* for typical users. Indeed, Gaw et al. [7] found that even activists with significant vested interest in securing their communications frequently did not realize the need for E2E encryption. Thus we believe service providers have a duty to help protect their users with the implementation of sane and secure default settings. Email service providers will not be willing to make E2E the default, however, until the usability issues have been solved in a satisfactory way so as to not drive users away from their offering. This paper aims to tackle exactly this issue.

Recent work by Ruoti et al. [17] examined the integration of E2E email encryption tools. Specifically, they proposed an E2E solution for the Gmail webmail client that used a key escrow system, requiring dependence on a trusted third party, to address the usability issues that stem from key management. They also compared an integrated solution called Pwm (pronounced "poem") against a standalone encryption program called Message Protector (MP). Their work postulated that users trust the system more when they must manually interact with ciphertext. This raises questions that tie into the work by Renaud et al. [16]: does interaction with ciphertext foster awareness, understanding, or knowledge, and thus increase trust?

We argue that the difference between Ruoti et al.'s two email implementations, Pwm and MP, exists around two orthogonal factors: integration of E2E encryption into the email client, and visibility of ciphertext. We describe the design and results of a study that evaluates user trust around these two dimensions of integration and transparency. To do this, we create a new integrated solution that makes use of the OpenPGP standard for interoperability. We use focus group testing and pilot studies to guide the design and enhance the usability of three E2E email encryption tools: a browser extension that encrypts email transparently (i.e., displays the ciphertext to the user), a browser extension that encrypts email while hiding the details of encryption, and a standalone encryption tool that requires direct interaction with ciphertext. We then evaluate these approaches with users by replicating and extending the original study by Ruoti et al. using our three different implementations of

Symposium on Usable Privacy and Security (SOUPS) 2015, July 22–24, 2015, Ottawa, Canada.

E2E email encryption.

Our work makes the following contributions:

- We show that it is possible to make an encrypted email client that is well-liked by users, while also adhering to the OpenPGP standard, by incorporating recent proposals for key management systems [13].
- We show that users have a clear preference for integrated secure email clients, when compared against manual encryption tools with similar effort put into their implementation and user experience.
- We present evidence that the transparency of encryption tools (the degree to which the details and results of encryption are shown to the user) does not have an effect on user trust. Our results show that trust hinges primarily on the software developer's online reputation. We also find that users think browser extensions are more likely to access their personal information than desktop applications, despite the reverse typically being true (as browser extensions operate within a heavily sandboxed environment).

In what follows, Section 2 will discuss work related to usable encrypted email. Section 3 will talk about the focus groups and pilot studies we used to guide our designs, and Section 4 will outline our evaluation methodology. We present the results of the user study in Section 5. Section 6 discusses design implications learned through the process, and Section 7 concludes.

2. RELATED WORK

The lack of adoption of encrypted email systems is certainly not due to the lack of implementations available [1]. There are a variety of existing tools for secure email, including browser extensions, websites, plugins, and standalone programs, available for both users of webmail and desktop mail clients. Still, the results of usability studies and the obvious failure of any implementation at widespread adoption indicate major problems with the adoption and continued use of these systems.

A great deal of recent literature on the usability of email encryption has been devoted to pinning down the precise reason as to why the widespread adoption of long-standing systems such as PGP has failed. Some authors maintain that usability is the deciding factor. In fact, since the bleak diagnosis provided by Whitten and Tygar [21] 16 years ago, subsequent PGP studies have shown little improvement to its usability. In the original study, only four out of twelve participants were able to successfully encrypt a message using PGP. Only one participant successfully completed all secure email tasks. After changes to PGP, including more automation and automatic decryption, the original usability study was repeated in 2006 [18] to show similar results: none of the participants was able to successfully encrypt a message and key exchange was still difficult to perform and not well understood.

Alternatives to PGP such as Key Continuity Management (KCM) [12] have been proposed to increase the usability of secure email by automating key generation and management. In this system, users create their own S/MIME certificates, much in the same way they generate their own PGP public keys. Garfinkel and Miller [6] conducted a usability study of KCM in 2005 to assess the advantages of existing S/MIME tools over the PGP tools available at the time. The key improvements in S/MIME addressed adoption difficulties by automating key generation upon sending email from a new address, and attaching a certificate (public key) to every outgoing email. The KCM system utilizes certificate pinning and alerts users of inconsistencies between the sender and their saved certificate. The results of this study showed that, although these improvements allowed users to correctly identify attacks in which an adversary spoofed a sender's identity, they still experienced difficulty with encrypting sensitive messages for the correct recipients. While users were technically able to send encrypted messages, this success did not extend to the wider problem of providing users with the means to make simple trust decisions (such as whether or not to trust a key initially) and correctly identify who could read their secured emails.

In 2013, Moecke and Volkamer [15] compiled a set of desirable criteria for usable E2E encryption tools. They argued that a usable secure email system must allow users to easily join, give them the tools and information necessary to make clear and informed trust decisions, not require them to understand the underlying details of public-key cryptography, and allow for secure communication without out-of-bounds verification. After an analysis of existing systems, including systems that used PGP and S/MIME, they conclude that no existing systems for secure webmail satisfy all of these criteria. Many systems that ranked slightly higher (such as Hushmail[1]) require trust in a third-party service provider.

There is another camp of authors who claim usability is not the key reason why many users still choose not to encrypt their communications. Recently, Renaud et al. [16] presented a hierarchy of mental states a user needs to progress through before she is able to adopt and use E2E encryption. The authors place usability of E2E far at the end of the list; they argue users must first be aware of privacy violations, be concerned about them, and see a need and capability to act before successfully adopting these systems for daily use. After conducting an exploratory study on 21 participants, they found most participants lacked a fundamental understanding of how email worked, could be exploited, or how to solve these issues once they are established.

Whitten and Tygar addressed the problem of education in their seminal paper, calling for *metaphor tailoring* to help users unpack the meaning behind public and private keys, encryption vs. decryption, and signing and verification. A recent study titled *Why King George III can encrypt* [20] revisits this idea by walking through basic secure email tasks, replacing technical terms with their analogous paper-mail counterparts. They motivated encryption with scenarios users can easily imagine being relevant to an 18^{th}-century monarch. A study on participant comprehension of these concepts found that while these metaphors sped up the explanation of secure email, they were not necessarily more effective in the end than the original technical language. After implementing these metaphors in a user interface, the authors still found that participants needed extra help to correctly perform encryption and decryption tasks.

There have been recent efforts outside of academia to address the concerns of usability and adoption by deploying and evaluating different approaches to email encryption.

[1] `https://www.hushmail.com/`

Open Tech Fund has compiled a partial list [1] of past and ongoing projects. Existing solutions include browser extensions similar to those discussed in this paper, such as Mailvelope[2] and Google End-to-End[3]. Other solutions, such as Mailpile[4] and Thunderbird's Enigmail[5] are email clients with integrated support for PGP. OpenITP recently released the results of a usability study on new Mailpile features [8]. Their study found that many of Mailpile's features enhanced the usability of key management and distribution, and also identified factors (such as awareness of encryption) that require further investigation. While the development of these tools shows great strides towards the adoption of encrypted email solutions, we have yet to see concrete evidence of the effect of individual features and design principles on usability and user trust. Our work is motivated by this large influx of tools and aims to lay some theoretical groundwork upon which more secure tools can be built in the future.

Our study is based on work done in 2013 by Ruoti et al. [17]. Their proposed system, Pwm, generates symmetric keys on a key escrow server to automate key generation and enable users to encrypt messages to contacts without obtaining public keys. As a result, the system hides almost all of the cryptographic details from the user. They found their system to be significantly more usable than select existing systems and users generally liked the effortless experience. However, they noticed that half of their users preferred their implementation of a standalone encryption application named Message Protector (MP). They inferred that this preference was due to a higher degree of trust that stems from being more involved and aware of the encryption process. However, this conclusion was derived from qualitative analysis as the experiment was not designed to probe such factors. In our work, we modify the experimental design significantly in order to explicitly probe whether this awareness is the real reason for the difference in trust, or if it can be attributed to other factors (such as differences in the design quality of the different applications).

3. DESIGN AND IMPLEMENTATION

We conducted a focus group and two pilot studies to guide the design of our encrypted email tools. We designed three separate tools, each with different levels of integration and transparency. Figure 1 shows the extent to which each of our three tools are integrated in the webmail client (also automated in the sense that the user is not required to perform additional encryption steps) and the extent to which they are transparent (i.e., show the details and results of encryption to both the email sender and email recipient).

We created two integrated encryption tools in the form of a Chrome browser extension with different levels of transparency. The transparent version automatically encrypts and decrypts email messages with the click of a button and deliberately shows the encrypted ciphertext to the user. The opaque version provides exactly the same functionality, but instead hides the ciphertext with a user-friendly overlay.

We created a transparent standalone encryption program that requires users to manually copy and paste messages from and into a separate software program. This tool shows the resulting ciphertext to the user as they step through the encryption process. We chose not to implement an opaque, standalone encryption tool as this would be more difficult to implement, likely not interoperable with other PGP software, and would not aid in investigating the hypotheses we describe in Section 4.

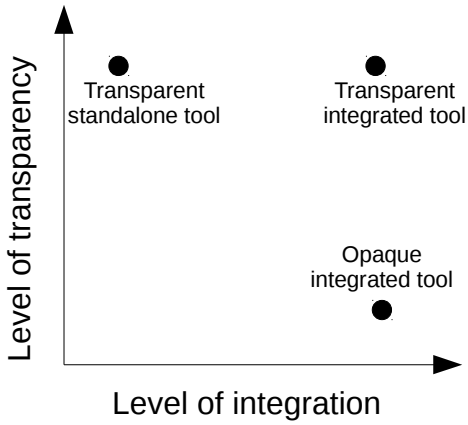


Figure 1: Placement of our tools on the transparency-integration scale

In the following sections, we discuss: how the focus group was conducted and how the results influenced our designs; how the initial pilot study was conducted and briefly, its results; how the final pilot study was conducted; and lastly, how the information and lessons learned from these studies influenced the final design of our software applications.

3.1 Focus Group

Before we began to design our tools, we conducted a focus group on existing encrypted email tools to get a sense of which features were the most desirable and which tasks posed the greatest problem to users. We gathered three participants (F0, F1, and F2) and had them perform a variety of tasks involving key management, encrypting emails before sending them, and decrypting emails upon receiving them. We had them perform the tasks using 1) Pwm, the integrated tool developed by Ruoti et al. [17], 2) Mailvelope, an encrypted email browser extension, and 3) the Enigmail plugin for Thunderbird. We did not include Message Protector, one of the programs developed by Ruoti et al., as it was not available to us at the time. The task order was varied using a 3x3 Latin square. All participants performed the tasks in the same environment using an email account we provided them solely for the focus group. All data analysis from this focus group was qualitative.

F0 and F1 were both computer science graduate students with experience using PGP software at some point in the past. They uncovered a number of issues with all three pieces of software, in both usability and generic software bugs. Their experience proved helpful in spotting a number of potential issues for ordinary users: for example, F1 noted that "it is as easy to send someone your private key as your public key [in Mailvelope]". F2 was a graduate student in Pure Mathematics with little-to-no prior exposure to encrypted email or cryptography.

We will now summarize our findings by software program.

[2] https://www.mailvelope.com/
[3] https://github.com/google/end-to-end
[4] https://www.mailpile.is/
[5] https://www.enigmail.net/

Mailvelope: All three participants had trouble finding the functions they needed (e.g., for generating private keys and importing public keys) in the Options window of Mailvelope. Each participant imported public keys in a slightly different way, including F0, who copy/pasted its contents despite the existence of a direct import-from-email feature in Mailvelope. All three expressed confusion as to whether the key had been imported successfully or not. Participants also expressed an annoyance with the lack of labels on the webmail overlay (F1: "What are these buttons?"). All three participants had difficulty understanding the dialog for selecting which users to encrypt the email for (which were not determined automatically from the list of email recipients). Participants also frequently chose to interact with ciphertext that was not intended to be human-editable: F1 managed to successfully import an incorrect public key by modifying the public key block.

Pwm: Participants were generally impressed with Pwm's simple, integrated interface (F2: "Oh really? No way!"). However, F1 missed the "encrypt" button in the overlay and unknowingly sent an unencrypted email. F2 *almost* sent an unencrypted email, but caught themselves at the last moment and decided to try clicking the lock button. F0 and F2 noted that the icon (an unlocked lock) is ambiguous as to whether a user should click to unlock (thus the message is locked by default) or if they had to click to lock the message. This finding is similar to the issues participants had with Mailvelope's use of unlabeled icons for buttons.

Enigmail: The Enigmail extension for Thunderbird is intended to open the setup wizard on first use, but there was a bug which prevented this from occurring on the current version of Ubuntu we set up for the focus group. We thus had to instruct all three participants to manually start the wizard from a buried menu option. This wizard contains a significant amount of exposition and technical detail, and all three participants expressed exhaustion reading it and eventually gave up in favour of spamming the "next" button. F1 and F2 found the "generating randomness" explanation confusing and thought they had failed to follow the correct instructions. All three found the key management interface confusing, with F2 opening public keys in a text editor frequently instead of importing them. All participants also found the Sign/Encrypt checkboxes difficult to locate, and clicked the menu that reveals them numerous times to convince themselves that they were still checked. F2 forgot the passphrase they set to locally encrypt their private key during the setup phase.

In summary, participants encountered usability issues with all three software applications, but had a far easier time with Pwm. We believe this is due to the design decision of Pwm to prevent the user from interacting with any options panes whatsoever in order to accomplish the software's basic tasks. While the extra tutorial and options provided by Enigmail appear helpful, they in fact turned out to be detailed and verbose to the point that users found them overwhelming and started ignoring them. The results of this focus group provided us with three main design goals to guide the design of our encrypted email tools.

1. Easy setup: A wizard should prompt users for the minimum information needed to start using the system and should not overwhelm them with exposition.

2. Simple to use: Users should not need any interaction with the options screen whatsoever in order to accomplish basic tasks (i.e., encryption, decryption, and importing keys).

3. Clear and explicit: It should be obvious to users what the result of clicking a button will be, and whether or not their email will be sent securely *before* they click the send button.

3.2 Initial Pilot Study

With these goals in mind, we proceeded to design and implement our own integrated and standalone encryption tools. We will present the final versions in Section 3.4. We evaluated them by running an initial study with six participants. We recruited participants with no prior experience with email encryption tools from the Faculty of Mathematics graduate program to participate in our pilot study. Before running the study, we received clearance from our Office of Research Ethics. We used the results of this study to increase the usability of our tools and eliminate confounding factors that would affect their perceived trustworthiness.

We noticed a trend, as in the focus group, of users editing or interacting with text in ways they were not supposed to. Three out of six participants failed to copy the `--- BEGIN MESSAGE ...` header we placed at the top of messages for our standalone system. The purpose of this header was to indicate the beginning of a ciphertext message. In the integrated system, some participants added a plaintext copy of the message into the ciphertext block, below the header. This is a significant problem in the design of both systems, as it poses a threat to the confidentiality and integrity of messages. We alleviated this problem in the standalone system by removing the headers altogether and creating separate buttons for the encryption and decryption tasks.

We observed that four out of six participants encrypted to the wrong recipients using our standalone tool. (This problem did not exist in our integrated tool, which automatically detected email recipients from the Gmail compose window.) Some chose to encrypt to everyone in their address book, or to no one. We partially countered this problem by requiring them to select exactly one recipient to encrypt for, but solutions to this problem are limited by the nature of a standalone approach to encryption tools: there is no way for an unintegrated solution to compare who the message is being *sent to* with who it is being *encrypted to*.

Finally, we observed several isolated instances of confusion over various advanced aspects of the software. One participant got lost in the "Advanced Options" menu, and one had trouble with the manual key management process of the integrated tool. We addressed these issues by reorganizing and simplifying much of the UI.

3.3 Final Pilot Study

Before commencing the final version of our study, we recruited five participants as pilots (using the recruitment process described in Section 4.1). We used these pilot participants to find any remaining bugs in our implementation, problems with the wording of our instructions, and any final issues with our UI choices. We fixed issues between pilot participants as time permitted, and then fixed all outstanding issues once the five pilot runs were completed; no changes were made in the duration of the user study.

The recurring issue of users interacting with structured ciphertext appeared again, this time in the integrated tool.

To fix this, we locked the contents of the message body once it had been encrypted; users had to decrypt it first in order to edit it. We also found users entering the wrong text in some fields of the standalone system, so we added strict validation and feedback to as many input fields as we could.

3.4 Implementation Details

In this section, we describe the final versions of our three encrypted email tools. Although some example screenshots are given here to highlight key concepts, we have included a more thorough set of screenshots documenting our tools in Appendix C. Note that we used the branding of "Mailvelope" for our integrated tool and "Message Protector" for the standalone tool in order to give the impression of a completed product to the participants. We attempted to make the tools as similar as possible, limiting the differences to those required to achieve different levels of transparency and integration. We used the same UI widgets, consistent terminology, and similar messages for each tool.

3.4.1 Key Management

One of the most challenging aspects of designing an encrypted email client for everyday computer users is the concept of key management. In order to use PGP encryption, users need to first (securely) obtain a key from each intended recipient. Current PGP implementations [19] use a Web of Trust model in which users signed each others' keys to establish paths of trust, but this idea has failed to catch on with non-technical users [14]. Later specifications (such as S/MIME) proposed using trusted authorities to verify keys' authenticity, but this defeats the decentralization ideal offered by end-to-end encryption. Newer proposals such as verified keyservers (e.g., Keybase.io[6]) and Google End-to-End [9] (similar to certificate transparency [13]) suggest a middle ground, in which lesser security than direct out-of-band verification is obtained but less trust is placed in central authorities (such as key escrow servers). We believe these proposals represent a good *default* level of security for the everyday user: all users are protected from passive adversaries, but more complex options (such as out-of-band verification or trust paths) can still be applied within the same system, allowing advanced users to gain some protection against active attackers.

To this end, we implement key management using a simulated version of Keybase.io. The core principle of Keybase.io is that (unlike PGP's HTTP Key Protocol servers to which anyone can post a public key) the person uploading a public key for an email address must be able to *receive* email at that address, preventing dishonest users from uploading keys for arbitrary other users. This means the keyserver can be polled for a recipient key with high probability of the actual recipient being the owner of said key, as long as there are no active adversaries present in the system. With this capability, encryption software is able to transparently perform key management on behalf of the user for any correspondents that have similar software installed. The user interface in the failure case of this scenario simply needs to explain to the user that the recipient does not have compatible software installed. We believe this concept is easily grasped by most everyday users, as they experience it already with segregated IM and social networking applications.

[6] `https://keybase.io/`

3.4.2 Integrated Tool

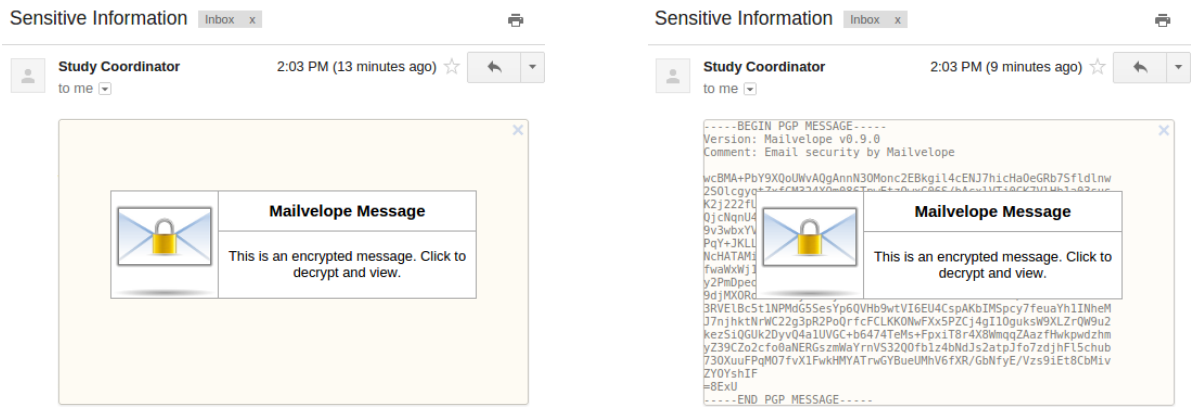
We chose to implement our encrypted email overlay by modifying Mailvelope, the browser extension we used in our focus group. We chose Mailvelope because it is open source, implements the OpenPGP standard, works with the current versions of Chrome, Firefox and Gmail, and also has a feature to support arbitrary webmail providers. Due to time constraints we chose to limit our development efforts to supporting Gmail accessed via Chrome on Ubuntu, although there would be minimal development effort required to expand our changes to include the full set of platforms and browsers. A Chrome extension package (CRX) for our modified browser extension is available [2].

In support of our *easy setup* goal above, we implemented a wizard that opens the moment the extension is installed. The wizard asks the user for only their name and email address, which is then used to generate a PGP keypair for them using sane default settings. Similar to Ruoti et al., we removed the requirement to set a local password protecting the private key and set it to be empty by default.

We successfully met our second goal of preventing the user from interacting with the options pane by adding mechanisms to the Gmail overlay that accomplished all required tasks. This included automatically setting sane default options. We implemented the key management mechanism described previously by emulating tie-in with Keybase.io. When the user checks "Encrypt this email", our emulated Keybase.io server is polled for keys corresponding to the recipients in the To: field. If a key for all recipients is found, the message is simply encrypted on the spot. If there are recipients for which no key is found, a prompt appears informing the user why the message cannot be encrypted ("the recipient does not have [software] installed"), and gives them the option to automatically send an invitation to the recipient to install it themselves. When the recipient accepts the invitation, a message is sent back to the user informing them they can now encrypt messages to that recipient.

Our third goal of being *clear and explicit* was met by replacing all of the UI widgets in the webmail overlay with labeled components, and providing explicit feedback to the user in strategic places. For example, we replaced the text of the "Send" button with "Send Unencrypted" with a checkbox labeled "Encrypt this email" placed right next to it. Once checked, the extension checks to see if there are public keys available for the typed recipients in the Compose window. If there are, the email is encrypted with PGP and the Send button is reverted to its normal text. We also replaced the iconography used to represent special messages (encrypted emails, key requests, etc.) with a box stating that it was an encrypted email object and a brief description of what would happen if the user clicks on it.

To facilitate investigation of our research questions in Section 4, we created two versions of our integrated tool ("transparent" and "opaque") with slight differences in the UI. In the opaque version, no ciphertext is ever displayed to the user. In the transparent version, the overlay on received encrypted messages is partially transparent, so the user can see the ciphertext behind it. Figure 2 shows these differences side by side. We also hide the ciphertext in the Compose Message window in the opaque version, replacing it with a message stating "This message is now encrypted".



(a) Receiving an encrypted message in the opaque version (b) Receiving an encrypted message in the transparent version

Figure 2: Comparison of opaque and transparent versions of the integrated tool

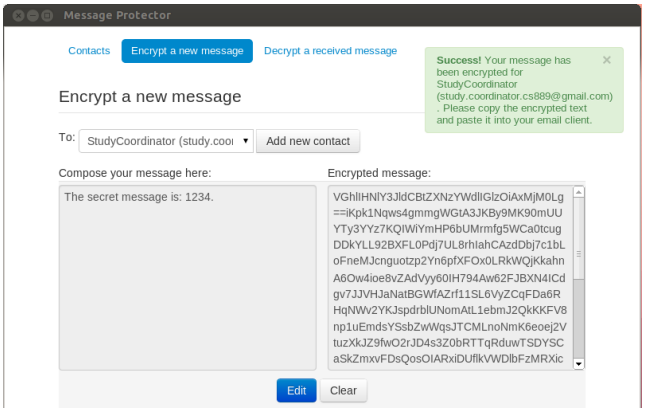


Figure 3: Encrypting a message using the standalone tool

3.4.3 *Standalone Tool*

We wrote our standalone program to resemble the Message Protector (MP) program from the original study as closely as possible. As Ruoti et al. made a point of MP being a standalone software application and not a web app, we created a desktop app using the Chromium Embedded Framework, which allowed us to implement it as a web app using Javascript and HTML5 and then install it locally to a user's desktop. Unfortunately, we had only a single screenshot and a description of MP's functionality to go from, and so were forced to infer some details of its UI. We also made a number of changes motivated by our focus and pilot studies, in order to make a best-effort approach to implementing MP's user experience. The code for the final application is available on our website [2].

Figure 3 shows Alice in the process of encrypting a message with sensitive information to Bob. After encryption, users must copy and paste the ciphertext from the output field into a webmail client of their choice. When receiving an encrypted message, they copy and paste the received ciphertext into the Decrypt pane of the standalone software. We implemented key management to more closely resemble the functionality of our integrated tool. When the user attempts to add a new entry to their contact list, the Keybase.io server is polled. If a key is found, the entry is simply added to the contact list and encryption to that contact can be done immediately. If no key exists, a message is displayed informing the user and asking them to copy/paste an invitation to the recipient, instructing them to install the tool. The contact entry is added to a pending contacts list, and the Keybase server is then polled periodically in order to see if the recipient has joined the system yet. Once they have, a notification is displayed in the software, and encrypted messages can then be sent to the contact.

4. METHODOLOGY

The goals of our experiment were two-fold; we wanted to demonstrate the usability of our integrated email encryption tool, and to investigate which software features are most related to user trust. We designed our experiment with the following hypotheses in mind:

H01: The integrated client is a usable encrypted email tool, as determined by the System Usability Scale (SUS).

H02: Standalone encryption tools provide a less desirable user experience than integrated encryption tools.

H03: The extent to which encryption software is automatic does not have a significant effect on user trust.

H04: Users trust transparent encryption software more than opaque software.

We conducted a mixed measures user study to evaluate these hypotheses. We used a between-subjects variable to evaluate the effect of transparency on user trust, and a within-subjects variable to evaluate the preference of an integrated, automatic encryption tool over a standalone, manual encryption tool. We will accept or reject the last three hypotheses based on participant responses to probing questions on tool preference and trust.

4.1 Subjects

We asked 36 participants (15 male, 21 female) to perform a sequence of email encryption tasks on two different systems (integrated and standalone). 18 participants were given the transparent version of the integrated client, wherein the ciphertext was visible after encryption and before decryption. The other half were given the opaque version of the client,

with hidden ciphertext. We also alternated between making participants perform the integrated tasks first and second. As with our focus and pilot studies, this study received clearance from our Office of Research Ethics.

We required all participants to be active webmail users (Gmail or Hotmail) to reduce the effect of differing experiences with email clients on the results. To help avoid bias from technical expertise, we excluded all current or former computer science students, and those with any background or knowledge of cryptography. We advertised our study on Craigslist and Kijiji, two popular local online classifieds websites. We also advertised our study to an introductory computing course for non-majors, a university-wide graduate student mailing list, and via posters put up around campus. We ended up with 33 students and 3 non-students. About 17% of participants were engineering students, 42% were science students (chemistry, biology, environment, health, and physics), 25% were studying mathematics (statistics, pure mathematics, and actuarial sciences), two were business students, and one was studying political science. Our non-student participants were a mechanic, a civil engineer, and a cashier. Participants knew from the recruitment materials and information letter that they were participating in a study on the usability of email privacy tools. They had no knowledge of the tools beforehand, and were not told that we were measuring trust.

The self-rated level of computer expertise in our participant pool ranged from minimal to expert. Of the 36 participants, 67% reported having previously sent sensitive information over webmail or Facebook. No participants had ever used email encryption, though a few reported taking precautionary measures when sending sensitive information. Some of these measures were effective: P06 and P09 reported previously sending sensitive information in password protected zip files, and P22 used the university's secure file transfer service. Others practiced mildly effective techniques: P02 made an attempt at concealing information by writing it in Chinese, and P03 sent protected information over mobile voice calls to increase the difficulty of interception. Others still exhibited a misunderstanding of features. P27 reported sending sensitive information over Gmail with Chrome's incognito mode, and P01 sent sensitive information over emails addressed with BCC. The majority of participants who sent sensitive information reported taking no precautionary measures (67%). Despite this trend, all but one participant considered maintaining the privacy of messages containing sensitive information to be "important" or "very important".

4.2 Tasks

We asked each participant to perform a set of tasks with both the integrated and standalone clients. These tasks included setting up the system, sending secure messages, and receiving secure messages. After each set of tasks, the participants then filled out a questionnaire. They answered questions about usability, suggestions for improvement, and trust. We used an interactive online survey for participant instructions and questionnaires.

We conducted the study in an empty room, with one computer for the participant and one for the study coordinator. They had access to a Jane Doe Gmail account, created for the purpose of the study. To protect the privacy of our participants, we asked them not to use their own personal email account. The participants worked on a virtual machine running Ubuntu 12.04LTS. We carefully explained the location and purpose of each icon at the beginning of the study. The launcher only contained icons for the programs they were required to use (Chromium web browser version 34, and the standalone client). We freely answered questions about the UI that were unrelated to the encryption software. We ran a desktop recording program in the background to review their actions and movements after they had completed the study. We also collected audio recordings during their completion of the study tasks and the post-study interview.

Participants had no knowledge or introduction to the tools before being asked to complete the tasks. We instructed them to complete the tasks to the best of their ability, and asked them to explore the interface if they were unsure about what to do next. We intervened or notified them of the correct actions only when they became stuck for an extended period of time. The purpose was to get a clear idea of how inexperienced users would interact with the systems, and what natural tendencies led to encryption mistakes.

In what follows, we describe the tasks performed using the integrated and standalone clients, as well as the questionnaire and interview process given to the participants.

4.2.1 Integrated Client Tasks

Setup: At the beginning of the integrated portion of the study, participants were asked to log into Jane Doe's email account to see a single message with a request to install a secure webmail extension. The text of this email contained a link from which participants could install and set up the Chrome extension. Normally users could be directed to the Chrome Extension Store to obtain the extension, but we used a separate download website for our study; some of the consequences of this are discussed in Section 5.

Email Decryption: After setup, participants received an encrypted email message from the study coordinator. To decrypt, they followed the instructions on the overlay and copy and pasted the decrypted text into the interactive survey.

Email Encryption (Part 1): Participants were asked to perform two different encryption tasks: in the first task, we asked them to reply to the message they received from the study coordinator in the previous task.

Email Encryption (Part 2): In the second encryption task, we asked participants to send a secure message to a new contact, for whom they did not yet possess an encryption key. The participant first had to email the contact with plaintext instructions on how to install and setup the integrated browser extension. After they received confirmation that their contact had signed up (in the form of an encrypted email), they were able to send an encrypted message.

4.2.2 Standalone Client Tasks

Setup: Our standalone tool was set up as a pre-installed standalone desktop application. We assume users are familiar enough with installing software to leave this part out of the usability study (and we did not want to influence them with the unfamiliar process of installing software in Ubuntu). When opening the tool for the first time, participants were asked to enter their email information and to add the contacts randomFriend@hotmail.com, mom@familyWebsite.com, and study.coordinator.cs889@gmail.com.

Email Encryption (Part 1): We then asked participants to use the standalone client to send a secure message to study.coordinator.cs889@gmail.com.

Email Decryption: Next, they received a reply from the study coordinator with an encrypted message. They were then asked to decrypt this message using the standalone client and paste the contents back into the interactive survey.

Email Encryption (Part 2): As in the integrated client tasks, we concluded by asking participants to use the standalone client to send a secure message to a new contact for whom they did not possess a public key. Participants were first instructed to send plaintext instructions to this contact. Again, the client simulated polling Keybase.io. When it found an added contact's key, it displayed a notification that the participant could now communicate with the contact securely.

4.3 Questionnaires

After completing the tasks for each system, participants answered a few survey questions about their experience with and trust level of the system (full text given in Appendix A). They first provided feedback on the usability of the system in the form of ten Likert scale statements. We used the results of this feedback to calculate the System Usability Scale rating of each system [5].

The remaining questions asked users what they liked and disliked about each system, how often they envisioned using the system, and quizzed their understanding of who could read the messages sent between them and the study coordinator. These were used to collect qualitative feedback on how usable and trustworthy users found the systems.

We concluded the questionnaire by asking participants to choose which system they preferred and their thought process behind that decision. The purpose of these questions were to gather more qualitative feedback on the usability of our systems, and to find out which aspects of the system were most important to users in terms of usability or trust.

4.4 Interview

To gain a deeper understanding of the answers to the last part of the questionnaire, we finished the study by asking each participant a series of increasingly probing questions about which system they preferred. We began by asking them to restate their answers at the end of the questionnaire, and then asked them explicitly about trust if they did not factor that into the reasoning behind their decision. Finally, we asked them to imagine a scenario in which they managed a business that required employees to send sensitive information to clients. We asked them which tool they would prefer their employees to use. The purpose of this question was to gain insight into what problems they considered important and how they assumed the general public would interact with encryption tools.

5. RESULTS

5.1 System Usability Scale

The System Usability Scale (SUS) was originally proposed in 1996 by John Brook as a means of evaluating the usability of products in an industrial setting [5]. SUS is based on a Likert scale measurement in which the evaluator indicates their agreement on a 5 point scale with 5 positive and 5 negative statements about the product. We use this scale to evaluate the usability of our tools in order to compare our work with the results of the similar experiments conducted by Ruoti et al. [17].

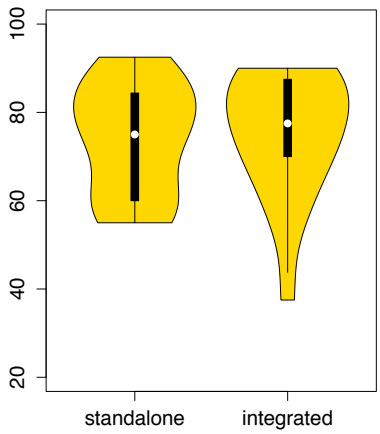


Figure 4: Absolute SUS scores for our encryption tools

We first calculated the SUS scores for the integrated and standalone clients, using only the half of participants who used each tool first. This was to obtain an absolute usability rating of each tool and avoid bias from participants comparing them to the tool they used before. The results were a SUS score of 75 ± 14 for the integrated client and 74 ± 13 for the standalone. These tools both receive an adjective rating of "good" according to Bangor's adjective ratings [3]. A summary of the SUS scores for each group of 18 participants are shown as violin plots in Figure 4. These enhanced box plots show the distribution of the calculated SUS scores from each participant. We found no significant difference ($p > 0.5$) using the Mann-Whitney U test for non-parametric data in the absolute usability of these two tools. The similarity in user experience ratings between these two tools gives a strong indication that the differences in qualitative feedback are due to our experimental factors (i.e. transparency and integration), and not confounding design factors. Our SUS scores for both tools are comparable to those reported by Ruoti et al. (76 for the integrated tool and 74 for the standalone).

We then calculated the SUS scores for each tool from participants that had performed tasks with the other tool previously. The usability rating for the integrated client, among participants who had already seen the standalone client was 78 ± 16, and the usability rating for the standalone client among participants who had previously seen the integrated client was 52 ± 21. This latter result showed a statistically significant difference ($p < 0.01$) in usability ratings for the standalone client between participants who were seeing it for the first time and those who had already tried out the integrated client. Figure 5 shows the interaction of ordering with the SUS scores for the standalone and integrated tools. Each plot represents 18 people; the absolute plot contains SUS scores from the participants that used the tool first, and the comparative plot shows the SUS scores from the participants that used it second.

When comparing the integrated and standalone usability scores of all participants, we found that the usability score of our integrated tool (73 ± 15) was significantly higher than the standalone counterpart (63 ± 20, $p < 0.01$). We did not see a statistically significant difference between the SUS scores of the transparent and opaque versions of the integrated client ($p > 0.05$). The respective scores were 73 ± 15 and 80 ± 14.

	Behaviour ID	Occurrences	Description
Integrated tasks:	i-1-3	5	Asked if they were supposed to install extension after seeing warning
	i-3-5	7	Confused as to how to send a secure message
	i-3-4	3	Tried to use the standalone client to send a secure email
	i-4-6	14	Sent encrypted email before knowing the new contact had joined
	i-x-2	3	Sent a plaintext message
Standalone tasks:	s-1-1	6	Forgot to add contacts
	s-4-1	18	Sent encrypted email before knowing the new contact had joined
	s-4-4	4	Confused by invitation process
	s-4-7	3	Added new contact twice

Table 1: Common behaviours encountered in the performance of study tasks

5.2 Observational Results

We noticed several common behaviours our participants exhibited while performing the study tasks. The most common of these are summarized in Table 1 and discussed in more detail below. See Appendix B for a full table of issues and behaviours. Our numbering scheme identifies issues by tool and task number in the following manner: {i/s}-{task #}-{id} where the id is arbitrary.

i-1-3: The integrated setup task asked participants to install a browser extension. We did not have our extension on the Chrome Application store, and this triggered several browser warnings (see Figure 6). Five of our 36 participants showed hesitation in completing the installation process and asked us whether it was okay for them to proceed. It is possible that these warnings affected their trust of the system.

i-3-5: The third study task required participants to send an encrypted email to the study coordinator for the first time. As we withheld tutorial information about how to do so (for reasons explained in the previous section), we noticed some confusion as to how the encryption process worked. Several participants asked us how to send a message with the installed software instead of Gmail. We responded that they should experiment with the user interface and proceed with sending an email as they normally would. After this prompting, most participants noticed the existence of the encryption checkbox upon opening a new compose window.

i-3-4: Other participants exhibited a behaviour similar to i-3-5, in which they were confused about how to send a secure message with the integrated tool, but instead of asking for direction, they opened the standalone application visible in the application launcher of their desktop. We stepped in at this point and told them to proceed with sending an email through Gmail as they normally would.

i-4-6: The last study task for the integrated client asked participants to send an encrypted email to a contact for whom they did not possess a public key. The purpose of this was to observe how participants interact with, understand, and trust our simplified key management scheme. After participants sent an invitation email to the new contact, the study coordinator accepted the invitation right away and uploaded a public key to our Keybase.io server. The majority of participants waited for a confirmation email from the contact, explaining that they had installed the system and could now communicate securely. However, 39% of the participants sent the encrypted email without waiting for the confirmation. This was possible because key management was done by automatically connecting to a key server in the background. This behaviour may suggest that participants are not aware of the steps that are normally involved in key management. It also suggests that they are willing to take advantage of transparent key management schemes.

i-x-2: Only three of the 36 participants (8%) sent "sensitive information" in plaintext during the encryption tasks. Two of these participants had not noticed the encryption checkbox or "Send Unencrypted" button and expressed surprise when we pointed out their existence at the bottom of the compose window. The other participant did manage to encrypt the messages, but appended a plaintext copy before sending it to the study coordinator.

s-1-1: Six participants skipped the majority of the first study task, not bothering to add contacts to the system. They realized this fact upon reaching the encryption phase of the study, since they were unable to encrypt a message without first selecting a contact to send the message to.

s-4-4: As in the integrated client tasks, we asked participants to send an encrypted email to someone for whom they did not possess a public key. Some participants asked us how to "send something through" the standalone client. We explained that they had to add the contact and then copy and paste the instructions into their webmail client.

s-4-1: After participants invited the new contact to install the standalone application, many then proceeded to encrypt a message to the contact without seeing or noticing confirmation that the contact had installed the system. Our behind-the-scenes key management process made it possible to use the software without an understanding or awareness of the exchange of public keys. This willingness to proceed without confirmation or feedback is the same behaviour we witnessed during the integrated tasks (behaviour i-4-6).

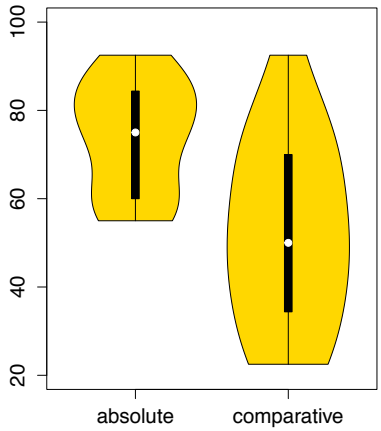
s-4-7: After receiving confirmation that the new contact had installed the standalone application and could now receive encrypted emails, three participants tried to add the contact to the system again. This could be a further indication of confusion surrounding key distribution, or it could be a failing of the user interface.

5.3 Qualitative Results

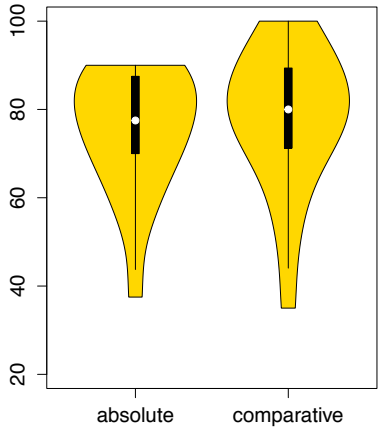
Five general themes surrounding usability and trust emerged from the questionnaire and post-study interview. We now discuss these themes in detail.

5.3.1 Preference for Integration

81% of participants preferred the integrated client over the standalone. The majority of them cited the integration into Gmail as the reason for their preference:



(a) Standalone tool usability scores



(b) Integrated tool usability scores

Figure 5: Interaction of ordering and SUS scores. "Absolute" refers to participants who used the respective tool first, whereas "comparative" refers to participants who had already used the other tool beforehand.

I find it more convenient... I don't have to open up another program to send the encrypted message, I can just choose whether or not to encrypt it when sending an email... (P07)

[The integrated tool] is much more convenient to use, to the point where I wouldn't mind encrypting even everyday, non-sensitive emails (P24)

I liked being able to use something that complemented the email system I currently use instead of having to learn something new then apply that to what I already use. (P30)

Others pointed out specific aspects of the standalone tool that were cumbersome or tedious:

I would have to go into a whole other program, open it up, encrypt it and if I'm . . . sending 50-60 messages a day it'd be difficult to do so and always have to go back and forth. (P22)

With [the standalone client] I had to go through this whole process... when you send many emails eventually that gets tedious... it's definitely something I would want to avoid. (P34)

This sentiment was especially apparent in the interview when we asked participants which tool they would prefer if they were in a managerial position.

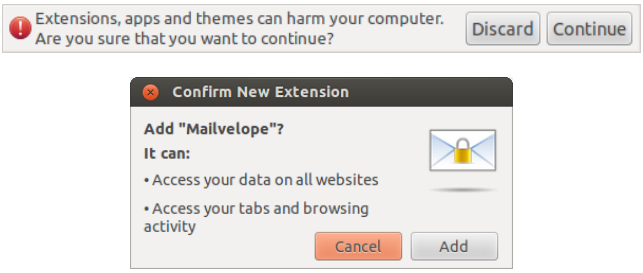


Figure 6: Two warnings that appear when installing extensions in the Chrome browser. The first appears only when installing from sources other than the Chrome Web Store.

I would rather have them use [the integrated tool] because if I give them a more complex system like [the standalone tool], there might have to be some kind of training going into it. If it's much more complicated, I might not even use it because it's so, well, tedious and I feel my employees might not even be using it. There's always that chance so I feel that [the integrated tool] is a much better option. (P01)

Some participants were also aware that the introduction of extra steps also introduced opportunities for human error, with P16 saying they could "see people accidentally not copying the entire message by missing a character and the message may become incomplete".

[The standalone tool] requires an additional program to be open, and copy and pasting to occur which I could mess up on. (P22)

Many participants also added that this tool would be easy to set up to encrypt messages by default. They expressed during the interview that in a business environment, they would prefer to have encryption always on. One participant expressed concerns that employees could forget to encrypt emails before sending even with the integrated client.

... it would be easy to train [employees] to check the box before they send it, but I suppose that would also make it easy for them to forget... you asked about the possibility of having it always on, so that would make it a good thing I think to use it that way. (P09)

5.3.2 Lack of Trust Preference

The overwhelming majority of participants (69%) did not trust one system over the other, even when prompted to think about trust. Most participants addressed the question of whether they trusted one system more with a simple "no", but others explicitly stated that they felt the two tools "did the same job" (P14).

Some participants with no trust preference did not consider email an appropriate method of secure communication.

Currently, I do not feel there is a need to encrypt my sensitive information. I typically do not share sensitive information via email... email encryption is too troublesome, especially when no one around me has used it before. (P21)

...for some extremely important information, I would like to talk face to face or through cellphone. (P04)

5.3.3 Distrust of Integration

Of the 31% of participants who did have a trust preference, most (10 out of 11) trusted the integrated client less than the standalone one because of its integration with the browser:

[The standalone tool] acts as a different entity that is on your desktop and not integrated online. So it feels more secure to encrypt and decrypt messages separately (P08).

Another participant pointed out that the seamless integration "...might give the sense that [this tool] is 'less safe' and more intrusive" (P26).

While the idea of using physical security measures to protect sensitive information (e.g., keeping a password in a locked drawer as opposed to a text file in a home directory) is correct, the notion that our standalone client and programs like it are offline and prevented from leaking information to third parties is incorrect. In fact, browser extensions can be considered "more secure" than separate applications due to the sandboxing they are subjected to. Only one participant mentioned that the integrated client "...seemed safer since it is an extension on the browser compared to [the standalone program] which is a software on my computer" (P32).

5.3.4 Reputation-Based Trust

Of the 69% of participants who had no trust preference, many (9 out of 25) mentioned that they would prefer to do research on the companies producing the software, or to hear about the tools' word-of-mouth reputation. One participant, without prompting, added that "actually I don't know whether I should trust these two systems" (P17) while stating their preference of the integrated client over the standalone. Another stated that their trust depended on "basically how well I know the company and if I don't know I'll research it and see if it's professional-looking." (P13)

5.3.5 Misunderstanding of Key Management

Several participants demonstrated a misunderstanding of who could read messages encrypted with the integrated tool. One participant thought that the standalone client was safer because it encrypted only for the recipient they selected, as opposed to the integrated client which had the same "encryption" (key) for every contact.

[The standalone client] had a different encryption for each contact, like it allowed different encryption routes for my privacy. (P27)

...with [the standalone tool], only certain people can receive the email and decrypt it. You have to add the contact. (P02)

Another participant did not like going through the invitation process and wanted to be able to send encrypted messages immediately. They wished for the invitation to include the ciphertext for the first message, so the recipient could read it as soon as they installed either piece of software.

5.4 Discussion

The quantitative data we collected for the evaluation of the usability of our encrypted email tools on the System Usability Scale, together with the qualitative feedback we received in the questionnaire and interview portion of our study, provide strong evidence in support of our hypothesis that our integrated client is a usable encrypted email tool (H01). It had a SUS score of 75, receiving an adjective rating of "good", and was comparable to the encryption tools proposed by Ruoti et al. (which received SUS scores of 76 and 74). The qualitative feedback on the integrated tool, summarized in the previous section, was positive. Most users stressed its positive aspects and integration when describing their preference for our system over the standalone system.

Our second hypothesis (H02), that standalone, manual encryption tools provide a less desirable user experience than integrated, automatic tools is also supported by the results from our quantitative and qualitative evaluation of the usability of both systems. The preference for integration was the strongest theme we encountered in participant feedback, with 81% of participants citing this as the primary reason they preferred the integrated client over the standalone one.

In our third hypothesis (H03), we expected to see no preferences in user trust between systems that were integrated (automatic) and those that were standalone applications (manual). Although the majority of our participants did not trust standalone software more than the integrated browser extension, 28% expressed a belief that the standalone system was more secure. Most reasoned that this was because it was not integrated into the browser. This is an interesting trend that perhaps mimics other systems in which diversification prevents losses due to a single point of failure (e.g. financial investments, nutrition, etc.). It is also possible that these few participants expressed this belief because of the browser warnings that were displayed when installing the integrated system. Five participants stopped the study to ask whether they were allowed to install the extension after seeing such a warning. Several cited this as the reason they trusted the standalone software more.

I would think that [the standalone tool] would be the safer one. That's just the feeling I got from it... the browser extension asks you for permissions when you go in, it asks to see what tabs you have and all that stuff. (P31)

We expected the differences in user trust to instead stem from the degree to which they saw the internal workings of the software, and the result of encrypting or decrypting a message (H04). Only one out of 36 participants expressed concern about using an opaque version of the extension.

I guess the [standalone] one ... it looks very encoded. The [integrated] one, it's black, but it doesn't actually... (P15)

More participants judged user error to be an important factor in trust than software transparency.

We did not create a hypothesis for the effect of transparency on usability, but we did find a noticeable, yet not statistically significant difference in the usability scores ($p > 0.05$) of the transparent and opaque versions of our integrated tool. One user specifically mentioned their preference for opacity from a usability perspective, which we recommend exploring in future work:

After you encrypted it, it shows a bunch of letters which is pretty long (P35)

We have strong evidence for our first two hypotheses (H01 and H02): that it is possible to make usable integrated encrypted email tools, and that they are preferred over similar standalone versions. Our replication of Ruoti et al.'s study contradicted the lack of preference they witnessed between integrated and standalone solutions. While roughly equal numbers of participants preferred each of their tools, we saw an overwhelming preference for an integrated solution.

Our evidence contradicts hypotheses H03 and H04 on the basis of trust preferences in different types of privacy software. Our methodology separated the two aspects of Ruoti et al.'s original tools—the level of transparency and integration—to find the design features that contribute to user feelings of trust. We found that it was the level of integration, and not the involvement and awareness of the encryption process that led some users to believe their information

was more or less secure. This finding is contradictory to the hypothesis of Ruoti et al. We also found that the majority of participants did not feel a different level of trust towards either of our tools. Instead, they either did not think about trust at all or based trust on company reputation and tool popularity rather than on software features.

5.5 Limitations

We conducted extensive focus group and pilot study sessions in an attempt to make all three versions of our encrypted email tools as usable as possible. This was to eliminate confounding factors due to tool design, and allow us to focus on the factors of transparency and integration in determining trust and usability. We aimed to make the integrated and standalone clients as similar as possible, within the constraints imposed by the level of integration. The System Usability Scale ratings discussed in Section 5.1 provide evidence for the accomplishment of this goal. However, it is still possible that minor differences affected the level of trust and quality of user experience that participants expressed.

It is also likely, given that we received the majority of our participants through our on-campus recruiting efforts (92% of our participants were students), that our participant pool was not representative of the general public. The age and tech-savvy bias of our participants may give a skewed vision of the overall usability of our tool, and could have an additional effect in the trust themes we discuss in Section 5.3.

There was a minor difference in the task order between the integrated and standalone portions of the study. As discussed in Section 4, the integrated tasks had participants decrypt a received message before encrypting a reply. In contrast, the standalone tasks had participants first send an encrypted message and then decrypt a reply. Our reasoning behind these differences was to emulate a normal workflow for each tool, given the differences in the setup procedures. We think it is unlikely that these differences would invalidate our results. As mentioned in Section 5.4, it is possible that the installation of the browser extension fed into the lack of trust a few users expressed in the integrated tool.

6. DESIGN IMPLICATIONS

The strongest result we observed was an overwhelming preference for the integrated, automatic encryption tool. This preference was apparent from both the comparative usability scores and the feedback in the questionnaires and post-study interview. Users liked the seamless integration of the encryption functions with a system they already use and are familiar with. They also recognized the tediousness of frequently copy-pasting ciphertext from a standalone client. At the same time, participants reported that they considered trustworthiness as a factor when choosing software. Some based this trust on intrinsic properties of the tools (such as their degree of integration), while others based trust on the reputation and popularity of the software developers.

This gives us more insight into designing usable encryption tools that also engender trust from users. With widespread adoption and ongoing use being the eventual goals, our results suggest we should focus on making integrated solutions more straightforward and trustworthy, rather than making standalone systems more usable. Furthermore, the usability of standalone systems is limited by their nature: key management will continue to be a manual and tedious task if standalone tools are unable to interact with users' contact lists. While integrated systems can automatically select encryption keys to match an email's recipients, a standalone tool requires users take care to select the correct recipient twice—once in the standalone encryption tool, and once in the email client itself. The other main usability complaint we received regarded the continual need to copy and paste ciphertext; this, again, is solved only by integrating the encryption software with the webmail client.

There are two main avenues we can explore to effect user trust of integrated systems. The first is to re-enforce the notion of sandboxing—several participants cited sandboxing as their reason for placing higher trust on the standalone system. This trust, however, overlooks the fact that browser extensions and plugins are generally placed under significantly more restrictions than desktop applications. Both systems polled servers for keys, and therefore had the ability to send and receive metadata without the explicit knowledge of the user (although it could be inferred indirectly from the invitation task). Through design, we may be able to indicate to users that their keys are stored locally with the integrated system as well, and re-enforce the notion that private information they enter will not be sent to their email provider, browser, or software developers. The second method of inspiring trust is to publish the integrated tool under a reputable developer's name, and provide the download from a trustworthy source (such as the major app stores). It is also important to permit, and even encourage, reviews from users of the software in a place where they can be viewed by other users. Users frequently put their trust in word-of-mouth opinions, and some would also search the Internet for impressions of the software from reputable authorities (such as reviews in online news outlets).

7. CONCLUSION

In this work, we showed that encrypted email clients based on the OpenPGP standard can be both usable and well-liked by ordinary webmail users. We showed that users have a strong preference for encryption tools that integrate tightly with their existing email client, as opposed to standalone encryption software that must be used separately. Our results demonstrate that, contrary to previous research findings, such standalone software does not inspire trust by forcing users to interact with encrypted objects. Rather, we found that a fraction of users believe desktop applications are more likely to operate in a purely offline manner, refraining from sending user data back to the developers, as compared to browser extensions. The majority of users, however, felt unable to make a distinction in trustworthiness between the two types of software alone, and would rather defer to popular opinion by way of online reviews or company reputation.

This work shows methods that could be applied to improve the usability of existing PGP tools. When working with unfamiliar concepts such as encryption, explicit UI labels help users feel more confident versus unlabelled icons. Common mistakes (such as sending plaintext email unknowingly) can be partially defended against with prominent indicators. Semi-automated key distribution allows users to send secure email without an understanding of how PGP works. We hope that successful integration with the popular webmail service Gmail will encourage these services to move towards making E2E encryption the default for all users.

8. ACKNOWLEDGMENTS

This work was made possible with funding from the Natural Sciences and Engineering Research Council of Canada and the Ontario Research Fund.

9. REFERENCES

[1] Overview of projects working on next-generation secure email. `https://github.com/OpenTechFund/secure-email`. Accessed Feb 2015.

[2] E. Atwater, C. Bocovich, U. Hengartner, E. Lank, and I. Goldberg. Paper companion website. `https://crysp.uwaterloo.ca/software/leadingjohnny/`, 2015.

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

[4] P. Bright and D. Goodin. Encrypted e-mail: How much annoyance will you tolerate to keep the NSA away? *Ars Technica*, June 2013.

[5] J. Brooke. SUS-a quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.

[6] 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 2005 Symposium on Usable Privacy and Security*, SOUPS '05, pages 13–24, New York, NY, USA, 2005. ACM.

[7] S. Gaw, E. W. Felten, and P. Fernandez-Kelly. Secrecy, flagging, and paranoia: Adoption criteria in encrypted email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 591–600, New York, NY, USA, 2006. ACM.

[8] Gillian Andrews. Usability Report: Proposed Mailpile Features. `https://openitp.org/sup/field-notes/`, December 2014.

[9] Google End-to-End Wiki. Key Distribution. `https://github.com/google/end-to-end/wiki/Key-Distribution`. Accessed Feb 2015.

[10] M. Green. The Daunting Challenge of Secure E-mail. *The New Yorker*, November 2013.

[11] G. Greenwald, E. MacAskill, and L. Poitras. Edward Snowden: the whistleblower behind the NSA surveillance revelations. *The Guardian*, 2013.

[12] P. Gutmann. Why isn't the Internet secure yet, dammit. In *AusCERT Asia Pacific Information Technology Security Conference 2004; Computer Security: Are we there yet?*, May 2004.

[13] B. Laurie, A. Langley, and E. Kasper. Certificate transparency. RFC 6962, RFC Editor, June 2013.

[14] M. Lee. Ed Snowden Taught Me To Smuggle Secrets Past Incredible Danger. Now I Teach You. *The Intercept*, October 2014.

[15] C. T. Moecke and M. Volkamer. Usable secure email communications: criteria and evaluation of existing approaches. *Inf. Manag. Comput. Security*, 21(1):41–52, 2013.

[16] K. Renaud, M. Volkamer, and A. Renkema-Padmos. Why doesn't Jane protect her privacy? In E. De Cristofaro and S. Murdoch, editors, *Privacy Enhancing Technologies*, volume 8555 of *Lecture Notes in Computer Science*, pages 244–262. Springer International Publishing, 2014.

[17] S. Ruoti, N. Kim, B. Burgon, T. van der Horst, and K. Seamons. Confused Johnny: When automatic encryption leads to confusion and mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 5:1–5:12, New York, NY, USA, 2013. ACM.

[18] S. Sheng, L. Broderick, C. Koranda, and J. Hyland. Why Johnny still can't encrypt: evaluating the usability of email encryption software. In *2006 Symposium On Usable Privacy and Security - Poster Session*, 2006.

[19] The Free Software Foundation. The GNU Privacy Handbook. `https://www.gnupg.org/gph/en/manual.html`, 1999.

[20] W. Tong, S. Gold, S. Gichohi, M. Roman, and J. Frankle. Why King George III can encrypt. `http://randomwalker.info/teaching/spring-2014-privacy-technologies/king-george-iii-encrypt.pdf`, 2014.

[21] A. Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, SSYM'99, pages 14–14, Berkeley, CA, USA, 1999. USENIX Association.

APPENDIX

A. USER STUDY INSTRUCTIONS

These are the questionnaires and instructions given to participants for the pilot study and main study. These questions are available in HTML format from our website [2], in addition to the script used to record answers and the instructions for the focus group described in Section 3.1.

A.1 Pre-study questionnaire

Welcome to our study!

Thank you for your participation. During this study, you will be asked to perform certain tasks using Gmail and then provide feedback to help us improve our software. During the course of this study, all acts taking place on the screen will be recorded along with audio of anything we discuss. This will help us learn whether or not our software is easy to use.

You will have access to a temporary Gmail account for use in completing tasks during this study. You will not be asked to use your own Gmail login name or password at any time. Do not enter or access any of your own personal data during the study since everything on the screen will be recorded.

Are you a student? (Yes; No)

What is your occupation or area of study?

What is your gender? (Male; Female; Other)

What is your approximate age? (18-23; 24-30; 31-40; 41-50; 51-65; 66+)

How long have you been a Gmail user? (I don't use Gmail; <1 Year; 1-2 Years; 3+ Years)

Approximately how often do you use webmail (Gmail)?

How would you rate your level of computer expertise? (Minimal; Minimal to Average; Average; Advanced; Expert)

Have you ever sent private or sensitive information via Web email or Facebook? (Yes; No)

If so, how did you send that information? (briefly explain):

How important is maintaining the privacy of your messages containing sensitive information? (Very Important; Important; Neither important nor unimportant; Unimportant; Very Unimportant)

Have you ever encrypted an email or Facebook message? (Yes; No)

If you answered yes to the previous question, please briefly explain how you did so.

A.2 Standalone client tasks

Message Protector Tasks

Message Protector (MP) is a computer program that allows users to protect Internet messages (e.g., email, Facebook private messages) via encryption. In this portion of the study, you will execute various tasks that comprise the primary functionality of MP and answer a few related questions.

Task 1: Set up MP

Open Message Protector from the toolbar on the left-hand side of your screen.

[image] This is what the MP icon looks like.

MP requires an email address and the email account password to allow the user's contacts to be able to read their protected messages. For this study, we have created the following test account for you to use:

Email Address: `jane.doe.cs889@gmail.com`

Password: `xxxxx`

Allow the following contacts to read your protected messages: `randomFriend@hotmail.com`, `mom@familyWebsite.com`, and `study.coordinator.cs889@gmail.com`.

In this scenario, you will encrypt and decrypt email messages with MP. The encrypted emails are sent and received, however, using Gmail. Click here to go to Gmail and log in with the credentials above.

Task 2: MP Email Encryption

Use MP to encrypt an email and send it to `study.coordinator.cs889@gmail.com` (note that you will need to copy and paste the message contents from the MP window into a new Gmail message). Include the phrase "The last four digits of my SSN are 6789" in the message.

Task 3: MP Email Decryption

After completing the previous task, you will receive a protected reply email from `study.coordinator.cs889@gmail.com`. Use MP to decrypt the message.

[image] It may take a few minutes for the email to arrive.

Type the decrypted message below:

Task 4: Adding contacts

Try sending a **secure** message to

`study.coordinator2.cs889@gmail.com` *(notice the "2" in "coordinator2")*

using MP. You will notice they do not have MP installed yet, so follow the instructions that appear to send them an invitation.

[image] It will take a few minutes for the study coordinator to receive and accept your invitation.

Once they have accepted the invitation, send the secure message. Include the phrase "The vault combination is 1234" in the message.

A.3 Standalone post-study questionnaire

SUS Questions:

Please answer the following question about MP. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the N/A column if you don't have a response to a particular statement. *Choose from 1 (strongly disagree) to 5 (strongly agree)*

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 found the system very cumbersome to use
8. I would imagine that most people would learn to use this system very quickly
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
11. My level of understanding of MP directly affects whether I would use it to protect my email messages.

Remaining Questions:

Who can read messages that you protect with MP? (Anyone that has MP installed, receives the message, and that I have selected to communicate with securely; Anyone who receives the message and who I have selected to communicate with securely; Anyone who receives the message; Anyone who has MP installed; I don't know)

After MP is installed, what actions must recipients take to read MP protected messages? (Access the MP software; Copy the message and paste to MP; Copy the message, paste to MP, click the Encrypt button; Copy the message, paste to MP, click the Decrypt button; I don't know)

How often would you use MP to protect your email messages? (Always; Very Often; Occasionally; Rarely; Very Rarely; Never)

What did you like about MP?

How could MP be improved?

A.4 Integrated client tasks

Mailvelope Tasks

Mailvelope is a computer program that allows users to protect email messages via encryption. In this portion of the study, you will execute various tasks that comprise the primary functionality of Mailvelope and answer a few related questions.

Task 1: Set up Mailvelope

Please login to our test Gmail account with the login name and password shown below. Read the first message and follow the instructions given in the message.

Click here to open Gmail

Username: `jane.doe.cs889@gmail.com`

Password: `xxxxx`

At some point, you will be prompted for your name and email. Please use the information below:

Name: `Jane Doe`

Email: `jane.doe.cs889@gmail.com`

Task 2: Mailvelope Email Decryption

Wait for a new message from the study coordinator.

[image] It may take a few minutes for the email to arrive.

Read the message and enter the secret phrase below:

Proceed to the next task.

Task 3: Mailvelope Email Encryption

Send a **secure** message to

`study.coordinator.cs889@gmail.com`

using Mailvelope. Include the secret phrase you were given in your message.

Task 4: Adding new Mailvelope Contacts

Try sending a **secure** message to

`study.coordinator2.cs889@gmail.com` *(notice the "2" in "coordinator2")*

using Mailvelope. You will notice they do not have Mailvelope installed yet, so follow the instructions that appear to send them an invitation.

[image] It will take a few minutes for the study coordinator to receive and accept your invitation.

Once they have accepted the invitation, send the secure message. Include the phrase "The vault combination is 1234" in the message.

A.5 Integrated post-study questionnaire

(Same SUS questions from A.3.)

Remaining Questions:

What did you like about Mailvelope?

What did you dislike about Mailvelope and how would you like it to be changed?

If you started using Mailvelope on your own, would you prefer protection for new messages to be? (Always on; Only on for the messages I decide are private; Usually off, unless I click a separate button on the Gmail page)

A.6 Post-study questionnaire

Post Study Survey

Please answer the following questions. Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the N/A column if you don't have a response to a particular statement.*Choose from 1 (strongly disagree) to 5 (strongly agree)*

1. I trust Gmail employees to not disclose, misuse, or abuse my email messages
2. I am concerned about Gmail scanning my messages
3. I worry that some messages aren't really from who they say they are from
4. I feel safe sending important information through email
5. I feel safe creating accounts with usernames and passwords on new sites
6. I feel safe installing browser extensions or plugins
7. Creating accounts for new websites is easy
8. Installing browser extensions is easy
9. I feel safe clicking on links in email messages
10. I feel safe clicking on links in email messages from people I know
11. I never click on links in email messages
12. I would trust a company other than Facebook or Gmail (e.g., MP, Mailvelope) to protect my email messages
13. I feel that it is important to encrypt my emails and messages that contain sensitive or private information
14. I would use a different Internet Encryption tool for every website that I store or share sensitive information

Have you installed browser extensions, add-ons or plugins before today? (Yes; No)

What has prevented you from installing browser extensions, add-ons or plugins in the past?

When deciding whether you will trust a browser extension, add-on or plugin, what influences your decision?

Have you ever been asked to send sensitive information you were not comfortable sending through email? (Yes; No)

What type of sensitive information were you asked to send?

Did you send the requested information? (Yes; No)

Have you ever received information you were not comfortable receiving through email? (Yes; No)

What type of sensitive information did you receive?

Which system would you prefer to use? (MP; Mailvelope; None of the above)

Please explain your answer to the previous question:

Thank you for completing our study! Before you go, please let us know if there is any additional information you would like us to have.

Additional Info:

B. FULL STUDY DATA

Tables 2 and 3 list the frequency with which various issues were encountered by participants using the integrated and standalone tools, respectively. Table 4 shows which tool each participant preferred, and their stated reason for their choice.

This data, plus the answers provided to our other survey questions described above, are available in spreadsheet form from the paper's companion website [2].

Behaviour ID	Occurrences	Description
i-1-1	2	Asked how to install the browser extension
i-1-2	1	Incorrectly enterred data during setup
i-1-3	5	Asked if they were supposed to install extension after seeing warning
i-1-4	6	Didn't click "Finish" button at the end of the setup phase
i-1-5	2	Opened advanced setup options and then closed them
i-1-7	1	Didn't know they had successfully installed extension
i-2-1	1	Immediately clicked overlay without reading it
i-2-2	1	Was unsure about clicking the "Encrypt message checkbox
i-3-1	1	Almost sent an unencrypted message, but encrypted at the last minute
i-3-2	1	Didn't see encrypte checkbox at first
i-3-3	1	Tried clicking encrypt checkbox first, and then composing the message
i-3-4	3	Tried to use the standalone client to send a secure email
i-3-5	7	Confused as to how to send a secure message
i-3-6	1	Closed compose window to modify an encrypted message instead of unchecking the checkbox
i-3-7	5	"This message has been encrypted" didn't show up (bug)
i-3-8	2	Were unsure they had sent the right message
i-4-1	2	Encrypted message only for Jane Doe (bug)
i-4-2	10	The "Encrypt message checkbox failed to appear (bug)
i-4-3	1	Sent two emails instead of one
i-4-4	1	Confused about how to get someone else to install the system
i-4-5	1	Didn't know how to modify the encrypted message
i-4-6	14	Sent encrypted email before knowing the new contact had joined
i-x-1	1	Skipped a task
i-x-2	3	Sent a plaintext message

Table 2: Behaviours encountered in the performance of integrated tool tasks.

Behaviour ID	Occurrences	Description
s-1-1	6	Forgot to add contacts
s-1-2	1	Added contact didn't show (user error)
s-1-3	2	Didn't enter contact names
s-1-4	2	Unsure how to add a contact
s-1-5	3	Expressed uncertainty as to how contacts were loaded/stored in MP
s-1-6	1	Asked whether tool would read read protected messages
s-2-1	1	Thought they had made a mistake while encrypting (but had not)
s-2-2	1	Did not select the correct contact to encrypt to
s-2-3	1	Appended plaintext to ciphertext after encrypting
s-2-4	1	Put plaintext in Gmail draft before encrypting
s-2-5	2	Did not know to copy encrypted message into Gmail
s-3-1	1	Incorrectly copied ciphertext
s-3-2	1	Tried to encrypt before adding contacts
s-4-1	18	Sent encrypted email before knowing the new contact had joined
s-4-2	1	Did not see or understand added contact notification
s-4-3	1	Encrypted invitation instead of sending it in plaintext
s-4-4	4	Confused by invitation process
s-4-5	2	Sent invitation but didn't add the contact to the system
s-4-6	1	Sent the invitation to the wrong person
s-4-7	3	Added new contact twice
s-x-1	1	Sent only plaintext

Table 3: Behaviours encountered in the performance of standalone tool tasks.

ID#	Preference	Reason for choice
P07	Integrated	I find it more convenient in that I don't have to open up another program to send the encrypted message, I can just choose whether or not to encrypt it when sending an email using gmail.
P11	Integrated	the ease of encryption options
P15	Integrated	It seems easier to use because it is integrated in the Gmail interface. However, I'd want to know a bit more about it before starting to use it...
P19	Integrated	more integrated into the user interface and makes sending an encrypted, secure message less cumbersome.
P23	Integrated	Easier to use because it can be sent right through my email.
P27	Integrated	It is very convenient as it is one go and less cumbersome than MP.
P31	Integrated	It's easier since it's integrated into gmail. I don't know if it's safer though
P35	Integrated	Easier to use
P39	Integrated	I found it more convenient because you don't need another software on. You don't need to copy and paste back and forth
P09	Integrated	integrated in gmail, not a seperate software and no need of extra copy+paste
P13	Integrated	[stand. tool] involves installing a new program whereas [int. tool] is directly through google.
P17	Integrated	The plugin is much easier to use for me, actually I don't know whether I should trust these two systems. If I don't know these systems are developed by students in our school, I will doubt the security level of these systems.
P21	Neither	Currently, I do not feel there is a need to encrypt my sensitive information. I typically do not share sensitive information via email. Also, email encryption is too troublesome, especially when no one around me has used it before (or send me invitations to use such a service to decrypt their messages).
P25	Standalone	Although it is not directly integrated into gmail, it was simpler to use and very straightforward. As well, I don't think having an 'encrypt' button at the bottom of every email I send is necessary, and might cause some confusion if accidentally pressed.
P29	Standalone	Even though it requires more steps for the encryption and decryption process, I feel as if it's more user friendly and still provides the same security.
P33	Integrated	Seems easier.
P37	Integrated	easy and simple getting protected without doing extra steps
P41	Integrated	seems a lot easier to use and add contacts
P12	Integrated	simple to use
P16	Integrated	wasn't its own program so it required less window switching. It also required less copy and pasting of the encrypted messages so would be quicker and easier to use, as I can see people accidentally not copying the entire message by missing a character and the message may become incomplete.
P20	Integrated	has a direct link in gmail. It makes it easier and less cumbersome. In terms of adding new contacts both systems require you to send an invitation. Both do not have any extra password requirement for decrypting the message. However, if [stand. tool] software has the feature that it can be downloaded by invitation only, it might make a difference. Otherwise, I don't see much difference between the two.
P24	Integrated	much more convenient to use, to the point where I wouldn't mind encrypting even everyday, non-sensitive emails with it.
P28	Integrated	seemed easier to use and less tedious
P32	Integrated	I liked the fact that it is really integrated into the browser, instead of in [stand. tool], where it is really inconvenient to switch windows just to encrypt and decrypt messages. It also seemed safer since it is an extension on the browser, compared to software on my computer.
P36	Integrated	It is a simpler process and doesn't require the constant switching between two different apps.
P40	Integrated	easier, for extremely important information, I would talk face to face or through cellphone.
P08	Standalone	I know I stated earlier that having a system like this integrated into your email would help but it acts as a different entity that is on your desktop and not integrated online. So it feels more secure to encrypt and decrypt messages separately.
P10	Integrated	simpler and faster to use as a browser extension and messages can be encrypted within Gmail
P14	Integrated	I find it is more easy to use and does the same job.
P18	Integrated	I don't have to access a separate window in order to use it
P22	Standalone	I would like both to use as one program. I like that [int. tool] is intregrated in gmail and makes it VERY easy to encrypt. But I also like that [stand. tool] could send private information over facebook, or even text messages. I think it is complex because it requires an additional program to be open, and copy and pasting to occur which I could mess up on.
P26	Standalone	I enjoyed its ease of use. It was similar to a "translation program" you would use like Google translate. This would make it easier for users that aren't familiar with using computers and foreign programs. [int. tool] was easier in the sense that it was integrated into Google. Users may not be comfortable with that since it's integrated into gmail, where [stand. tool] was a separate program not linked to gmail. This might give the sense that [int. tool] is "less safe" and more intrusive. [Int. tool] also didn't have a decryption function
P30	Integrated	I liked being able to use something that complimented the email system I currently use instead of having to learn something new then apply that to what I already use.
P34	Integrated	Like I stated before, it felt much more comfortable to use because it was better integrated into the email system. I didn't have to open up another program and do a bunch of other functions before sending an encrypted message.
P38	Integrated	It is less complicated and it encrypt it right away in the email

Table 4: The preference chosen by each participant, and the reason supplied for their choice.

C. SCREENSHOTS

This section contains screenshots supplementing Figures 2 and 3 for documenting the tools built for our study. Figures 7, 8, and 10 show various screens and prompts for our integrated encryption tool. Figures 9 and 11 show screenshots of our standalone tool. Note that we kept the branding of the tools during the user study (the integrated tool was branded as Mailvelope and the standalone tool was branded as Message Protector). We did this to identify the tools to the participants and give them memorable names to refer to during the questionnaire and interview.

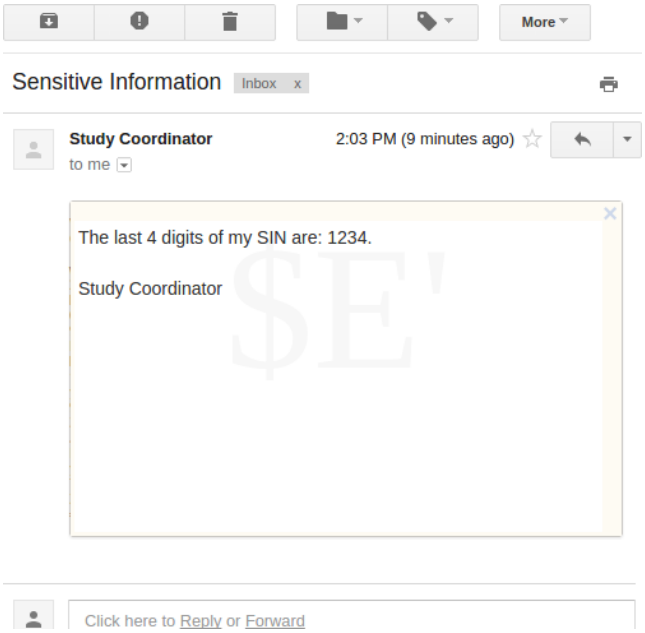


Figure 7: Screenshot of a decrypted message overlay in our integrated tool interface (the watermark is a security feature of Mailvelope, the underlying source code upon which we built our integrated tools)

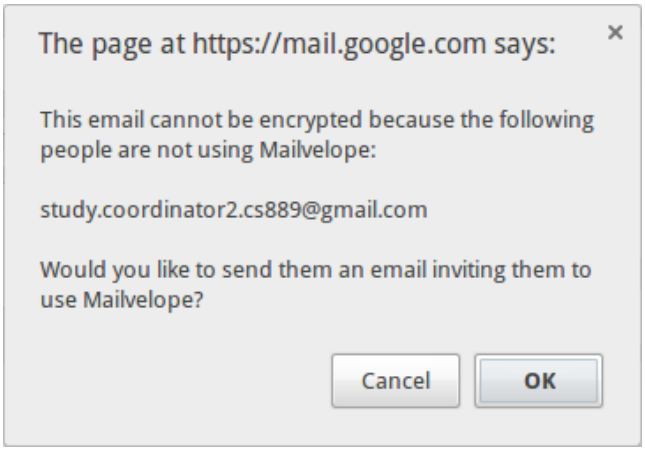


Figure 8: Prompt displayed by the integrated tool when no recipient key is found

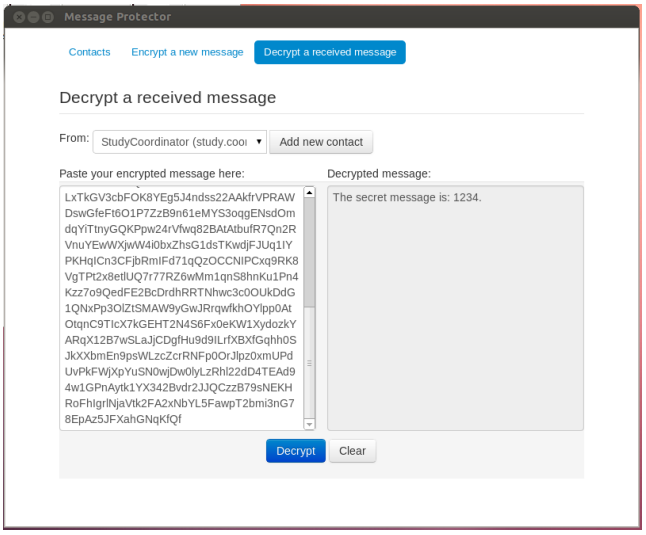
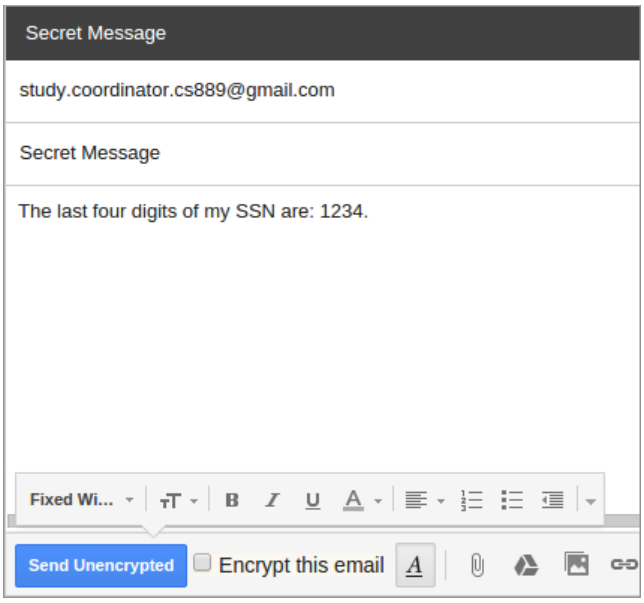
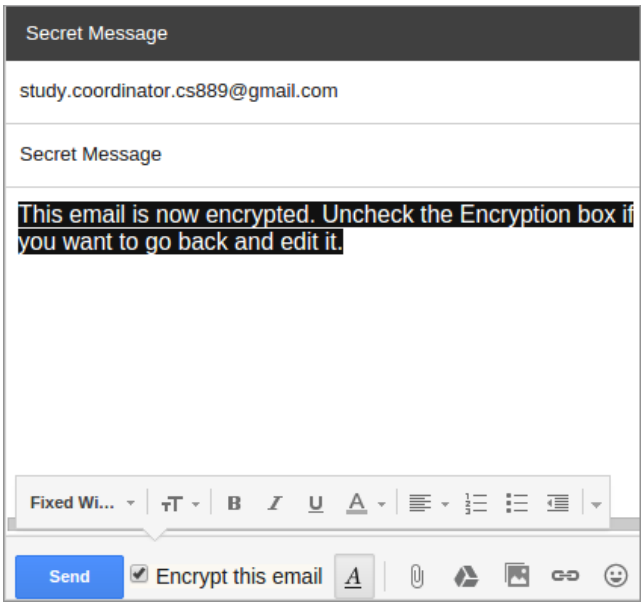


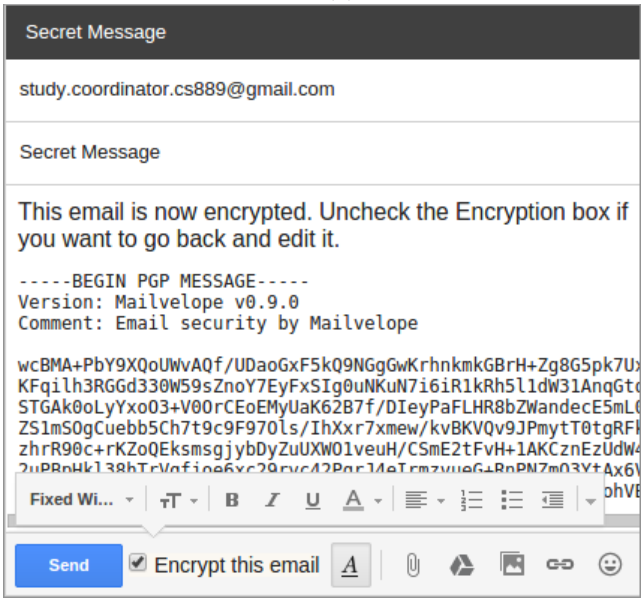
Figure 9: Decrypting a message using our standalone interface



(a) Compose window before encrypting a message

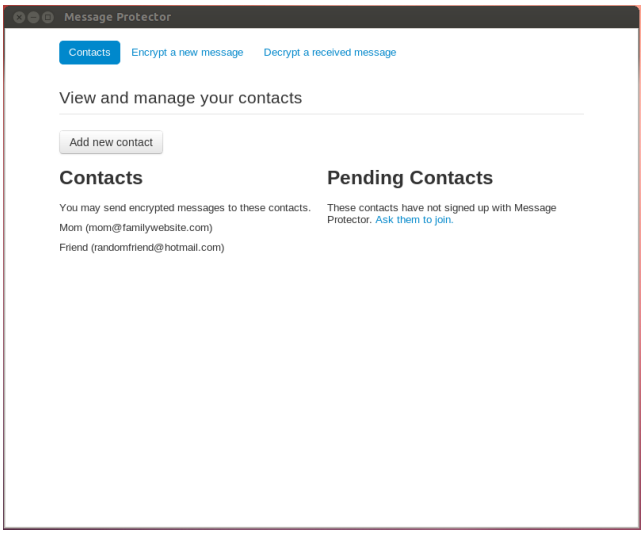


(b) After successfully encrypting a message in the opaque version

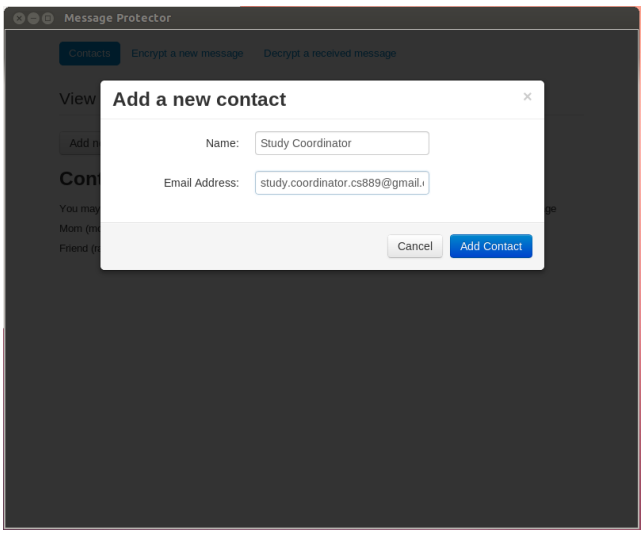


(c) After successfully encrypting a message in the transparent version

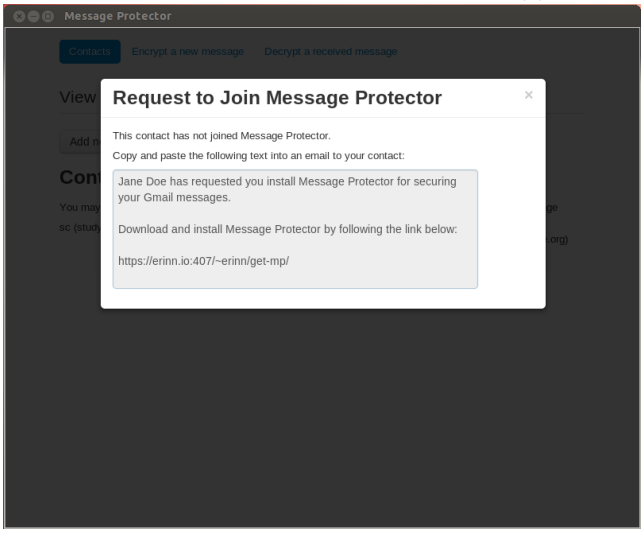
Figure 10: Screenshots of encryption in our integrated interface



(a) Contact list



(b) Adding a new contact



(c) Prompt displayed when no recipient key is found

Figure 11: Screenshots of our standalone interface