Private Webmail 2.0: Simple and Easy-to-Use Secure Email

Scott Ruoti[†*] **, Jeff Andersen**[†]**, Travis Hendershot**[†]**, Daniel Zappala**[†]**, Kent Seamons**[†]
Brigham Young University[†], Sandia National Laboratories[*]
{ruoti, andersen, hendershot} @ isrl.byu.edu, {zappala, seamons} @ cs.byu.edu

ABSTRACT

Private Webmail 2.0 (Pwm 2.0) improves upon the current state of the art by increasing the usability and practical security of secure email for ordinary users. More users are able to send and receive encrypted emails without mistakenly revealing sensitive information. In this paper we describe four user interface traits that positively affect the usability and security of Pwm 2.0. In a user study involving 51 participants we validate that these interface modifications result in high usability, few mistakes, and a strong understanding of the protection provided to secure email messages. We also show that the use of manual encryption has no effect on usability or security.

Author Keywords

Security; Usability; Secure email; Encryption

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces—*user-centered design*; H.1.2. Models and Principles: User/Machine Systems—*human factors*

INTRODUCTION

Recent measurements demonstrate that email is largely insecure [8, 6, 11]. Although adoption of connection-based encryption and authentication is increasing (i.e., STARTTLS, SPF, DKIM, DMARC), these technologies are often configured incorrectly and have weaknesses that render them susceptible to attacks. However, even if transmission of email were properly secured, email at rest is generally unencrypted. This means that an honest-but-curious email provider or a government that coerces an email provider into granting access can potentially scan email messages. End-to-end encryption is a much stronger protection mechanism. With end-to-end encryption, the sender encrypts email messages before transferring them to the email provider, ensuring that no one but the intended recipients is able to read the messages.

Unfortunately, usable secure email is still an open problem more than 15 years after Whitten and Tygar's seminal paper, *Why Johnny Can't Encrypt* [26].[1] Arguably the most secure form of end-to-end encryption is PGP, but PGP has continually proved itself unusable [26, 21, 15]. More recently, research has shown that by relaxing the security requirements of PGP, it is possible to create usable, secure email [10, 16, 1]. While these newer systems have lower *theoretical* security than PGP, they have higher *practical* security—meaining these new tools allow ordinary users to encrypt their email, something they are largely unable to do with PGP-based tools.

Of these usable, secure email systems, Private Webmail (Pwm) is specifically targeted at helping the masses encrypt their existing webmail accounts, such as those provided by Google, Yahoo, and Microsoft. A set of user studies of Pwm demonstrated that it had significantly higher usability than similar secure webmail systems (Encipher.it and Voltage Mail [16]). Still, in this evaluation of Pwm about a tenth of participants accidentally sent email in the clear when they meant to encrypt it. Many more users were unsure of the security provided by Pwm.

In this paper, we describe an improved interface design for Pwm that addresses issues raised in the original studies. First, we added an artificial delay to encryption that enhances users' confidence in the strength of the message encryption while simultaneously instructing users on who can read encrypted messages. Second, we modified the email composition interface to help users understand which emails are encrypted, so that they avoid mistakenly sending plaintext in the clear. Third, we added contextual clues that help users understand how to correctly use secure email, showing how the recipient, subject, and optional greeting fields are used. Finally, we implemented inline, context-sensitive tutorials, which improved view rates for tutorials from less than 10% for Pwm to over 90% in some cases for Pwm 2.0. Through a user study involving 51 participants, we demonstrate

[*] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

[1] We note that S/MIME is widely used in certain organizations (e.g., U.S. government), but this adoption has not spread to the masses.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
UIST'16, October 16–19, 2016, Tokyo, Japan.
ACM ISBN 978-1-4503-4189-9/16/10...$15.00.
DOI: http://dx:doi:org/10:1145/2984511:2984580

that our new design, referred to as Pwm 2.0, is significantly more usable and secure than Pwm.

Our main contributions are as follows:

1. We describe the interface modifications we used to build Pwm 2.0. Our user study demonstrates that these modifications had a significant impact on the usability and security of Pwm 2.0 as compared to the original Pwm system. Moreover, we discuss user reactions to these interface modifications and lessons learned from our study. We believe these principles could also benefit other encryption applications (e.g., instant messaging, social networks, file sharing).

2. Pwm 2.0 scores an 80.0 on the System Usability Scale (SUS), rating in the "excellent" category for usability and receiving an "A" grade. This score is in the 87th percentile for system usability [18] and is the highest SUS score for secure email to date. Moreover, over 80% of users wanted to immediately begin using Pwm 2.0, and over 90% felt that their friends and family could use Pwm 2.0 with relative ease.

3. We show that with Pwm 2.0 users rarely send plaintext messages when they meant to encrypt the message. Out of 306 tasks, only six mistakes were made using Pwm 2.0 (2%), a significant improvement to Pwm's rate of about 10%. In addition, the majority of users understood the confidentiality, authenticity, and integrity Pwm 2.0 provided their messages (84%, 63%, 76%, respectively).

4. As part of our study, we split users into two groups using different versions of Pwm 2.0 supporting either automatic encryption or manual encryption. Our results demonstrate that, contrary to the findings of the original Pwm study [16], requiring a user to view their encrypted message before sending it (manual encryption) has no benefit over encrypting and sending the message in one step (automatic encryption).

BACKGROUND

In this section, we review related work for secure email and automatic versus manual encryption. We then describe the threat model that motivates our work. Finally, we describe the original Private Webmail system.

Related Work

Whitten and Tygar [26] conducted the first formal user study of a secure email system (i.e., PGP 5). They found serious usability issues with key management and users' understanding of the underlying public key cryptography. Subsequent studies of newer PGP-based clients have shown that PGP tools are still unusable for the masses [21, 15].

Garfinkel et al. [10] conducted a usability study of secure email based on S/MIME, and observed that hiding encryption details can impact usability. For example, they found that because the integration of encryption into Outlook Express "was a little too transparent," users were often unsure whether a given email was encrypted. Additionally, they found that some users failed to read the instructions associated with various visual indicators, leading to difficulties in understanding the interface. Still, they found that automating key management increased the overall usability of secure email.

Fahl et al. [7] explored manual and automatic encryption in the design of a Facebook Chat system. Here, automatic encryption refers to encrypting and sending a message without explicitly demonstrating that encryption has occurred, whereas manual encryption shows the user ciphertext and requires them to manually send this ciphertext. Their user study demonstrated that users reported preferring automatic key management, but found no significant difference between automatic and manual encryption. Nevertheless, they raised the issue that automatic encryption could impact users' feelings of trust and recommended the issue be addressed in future work.

Ruoti et al. [16] conducted a series of user studies with Private Webmail (Pwm), a secure email prototype that tightly integrates with Gmail. Even though results showed the system to be quite usable, they found that some users made mistakes and were hesitant to trust the system since the automatic encryption was too transparent. They also conducted a study comparing Pwm to a desktop application (MP) that supported manual encryption. They found that participants using MP's manual encryption made fewer mistakes and answered more questions correctly on a quiz that tested their understanding of the system. However, there are significant differences between Pwm and MP, which are confounding factors in Ruoti et al.'s results. In this paper, we also examine the differences between automatic and manual encryption, but do so using two variants of our Pwm 2.0 system that only differ in regards to the use of automatic and manual encryption.

Atwater et al. [1] have also examined the question of automatic and manual encryption brought up by Ruoti et al.'s study. They created two version of a tool based on Mailvelope, and these versions only differed in their usage of manual and automatic encryption. They found that manual encryption had no statistically significant effect on participants' trust of their secure email system. Their work differs from ours in that they examined the question of trust whereas we are looking at understanding and mistakes, the two pivotal points from the original Pwm study. In addition, though Atwater et al. demonstrate usability with a PGP-based client, the system they simulated does not include several practical key management steps that users would be required to perform, limiting the applicability of its results.

Threat Model

In our threat model there are four entities:

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

2. **Webmail provider.** We consider the webmail provider as an honest-but-curious party.[2] The webmail provider has access to the users' encrypted messages, but not to the keys that encrypt those messages.

3. **IBE Key server.** Key management in Pwm uses Identity-Based Encryption (IBE) [20]. In IBE, a user's public key can be computed using public parameters from the IBE key server and the user's identity (e.g., email address). To retrieve the private key for a given identity, a user proves ownership of the appropriate identity, and then the key server will generate and send the private key to the user.[3]

   We consider the IBE key server as an honest-but-curious party. While the key server is responsible for generating a user's private key, it does not have access to the messages encrypted with those keys.

4. **Adversary.** The adversary is free to eavesdrop on any communication between users, webmail providers, and key servers.[4] Additionally, the adversary can attempt to compromise the webmail provider or key server. The adversary wins if she is able to use these resources to access the plaintext contents of the encrypted email body.

We do not consider attacks directly against the user or trusted computing base (i.e., phishing credentials, installing malicious software). Similarly, we do not consider an attacker who can compromise fundamental networking primitives (i.e., TLS, DNS). While these are valid concerns, if the attacker can accomplish these types of attacks, then they can already do far more damage than they could by breaking the secure email system. We also note that data needed by the webmail provider to transmit email (e.g., recipient addresses) cannot be encrypted, and may be available to the adversary (e.g., this information is passed over an unencrypted channel). Our threat model instead focuses on ensuring the data in the encrypted body is safe from an attacker.

To steal the user's sensitive data, the adversary must obtain both the encrypted email and the key material needed to decrypt the email. The former can be accomplished by either compromising the webmail provider, or intercepting an encrypted email that is not transmitted using TLS. The latter can be accomplished by compromising the IBE key server. Just as the adversary must collect the data from both the webmail provider and key server, neither of these parties alone has enough information to unilaterally steal the user's sensitive data. While as honest-but-curious parties, these two entities will not collude, a government could subpoena both organizations and compromise a user's sensitive data. If this is a significant concern, it is possible to apply a thresholding scheme to the IBE key server and place the various key servers in different localities that for political reasons will not cooperate.[5]

While our threat model is more permissive than the traditional PGP threat model, it is nonetheless useful in a variety of situations. For example, it satisfies the typical case where a small business or a university has outsourced its email to a third-party service provider such as Google, but does not trust Google with sensitive data. The business or university does, however, trust a browser extension that it has vetted and a key server that it operates. Alternatively, our threat model allows everyday users to easily encrypt sensitive email, preventing their webmail providers from storing, scanning, or accidentally leaking sensitive information. Regardless, the model provides greater assurance than currently available through vanilla SMTP [6].

Private Webmail (Pwm)

Pwm implements secure email through tight integration with Gmail and leverages automatic key management. When users read or compose secure email, Pwm replaces portions of Gmail's interface with Pwm's own secure interface. Pwm's interfaces are styled differently than Gmail's, providing a clear demarcation of which information is being protected by Pwm. Pwm's interfaces are implemented using *security overlays* [25]. Security overlays allow Pwm's interfaces to be visually integrated within Gmail's interface, while still protecting the content of Pwm's overlays from Gmail.[6]

All secure email sent using Pwm includes instructions on how to setup Pwm for first-time users. The setup instructions direct new users to the Pwm website, where they are able to add a bookmarklet to their browser's bookmark storage.[7] The new user then returns to Gmail and clicks on the Pwm bookmarklet to run Pwm. The benefits of using a bookmarklet to run Pwm include not requiring installation permission on the machine and also avoiding any user worries related to installing extensions [16]. The drawback is that users are required to click on the Pwm bookmarklet each time they reopen Gmail.

When Pwm is running, secure email messages are automatically decrypted for users. The decrypted contents of the message are displayed in place of the instructions

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

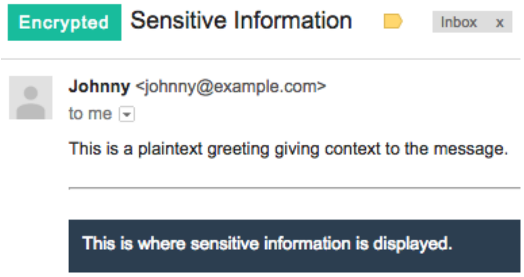
[3] Private keys are not stored on the key server, but rather generated on demand based on a master secret. For added security, this secret can be stored inside of a crypto-card.

[4] In nearly all cases, this communication will be encrypted using TLS, and the adversary only has access to the encrypted packets.

[5] Alternatively, the IBE key server could roll over its keys on a regular (e.g., daily) basis, limiting the amount of time an adversary or a government has to steal the user's sensitive data.

[6] This prevents Gmail from being curious and scanning the contents of encrypted emails.

[7] A bookmarklet is a browser bookmark that contains executable JavaScript. This JavaScript is run on the page that users are viewing when the bookmark is clicked.



Encrypted Sensitive Information

Johnny <johnny@example.com>

to me

This is a plaintext greeting giving context to the message.

This is where sensitive information is displayed.

Figure 1. The read interface for an encrypted email.

and ciphertext that were in the email message's body (Figure 1).[8] Encrypted email messages in the user's inbox are marked as *Encrypted* (Figure 2). Pwm allows users to add encryption on a per-message basis by clicking a lock icon inserted next to the "Send" button at the bottom of the compose interface.

In Pwm, users are authenticated to the IBE key server using Simple Authentication for the Web (SAW) [24], a form of email-based authentication [9]. Since Pwm runs inside the user's Gmail session, it is able to complete authentication automatically and does not need to prompt the user for input. It should be noted that because Pwm authenticates to the IBE key server using email, if an adversary were to compromise the user's webmail account they would also be able to authenticate to the key server. This means that messages encrypted with Pwm are only as secure as the user's webmail account. Ruoti et al. took this approach in order to maximize usability and because it still provides greater security than currently available to webmail users [8, 6, 11]. If greater security is needed, the IBE key server could be modified to authenticate users by some other means (e.g., separate password, two-factor authentication).

USABILITY IMPROVEMENTS

We used an iterative design methodology to address problems with Pwm's original interface. For each problem, we brainstormed several potential solutions and evaluated each solution using cognitive walkthroughs and heuristic evaluations. The most promising solutions were then implemented in prototypes, and these prototypes were evaluated with pilot usability studies conducted with a convenience sample from friends, family, and students. Those solutions that were rated as helpful in our pilot usability studies were then implemented in our Pwm 2.0 system. The source for Pwm 2.0 can be found at `https://bitbucket.org/isrlemail/pwm`.

In the remainder of this section, we describe the most relevant solutions that were added to Pwm 2.0.

Delayed Encryption

One concern expressed by a significant portion of participants in the original Pwm study [16] was that Pwm encrypted email so quickly that participants were unsure if Pwm had really done anything. Also, participants indicated the encryption process was so invisible that they were unsure who could read their encrypted email message.

To address both of these problems we added an artificial delay after users click the *Send encrypted* or *Encrypt* buttons. For each email recipient, users are shown a message lasting 0.75 seconds that states the email is being encrypted for that user (e.g., "Encrypting for `bob@gmail.com`"). This helps users understand who will be able to read the encrypted email and also lets them feel that something substantial has happened during the encryption process.[9] Because most email messages have a small number of recipients, this artificial delay does not significantly impact the overall email experience.

Compose Interface

In the original studies, about 10% of users accidentally sent a sensitive email without first encrypting it. In each case, the user recognized that they had made a mistake immediately after hitting the send button, but unfortunately their sensitive information had already been transmitted in the clear. Moreover, many users initially composed their sensitive emails in plaintext, only encrypting them right before sending the message, allowing Gmail to scan their message as it was being typed.

To address both of these issues, we revised Pwm 2.0's compose interface to better conform to the flow of composing email messages in Gmail (i.e., moving from top to bottom). As part of this process, we added informative text at the top of the compose interface (before the data entry sections) that indicates to users whether their message is being encrypted, then allows them to enable encryption before they begin composing their message (Figure 3). When encryption is enabled, we modify the informative text to make it clear that the message is now being encrypted (Figure 4). Furthermore, we modified Gmail's *Send* button to read *Send unencrypted*, to make it clear to users that by default their messages are sent without encryption.

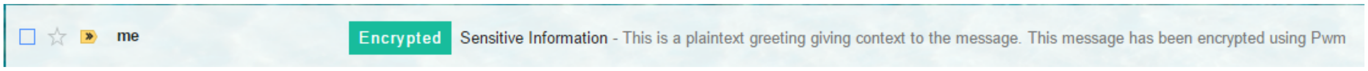
To help users better understand how Pwm 2.0 protects their emails, we modified the *placeholder* text for the `Recipients` and `Subject` fields, explaining how Pwm 2.0 uses these two fields (Figure 4). Also, we added functionality allowing users to compose plaintext greetings that are included with the encrypted email (Figure 1 and Figure 4).[10] The greeting can help engender trust in a new user that receives an encrypted email, giving them confidence to setup and use Pwm 2.0. It also provides context for encrypted email messages before they are decrypted.

Tutorials

[8] Screenshots in this section are of Pwm 2.0.

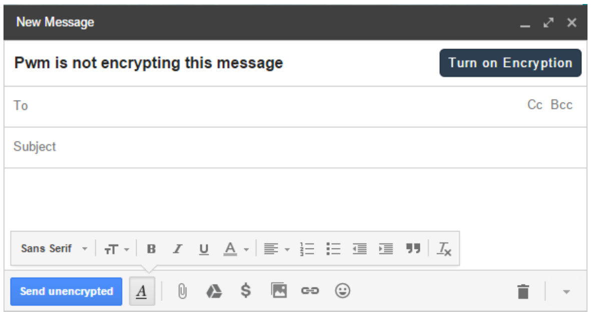
[9] Future work could examine the effect of removing this delay for experienced users who already have a correct understanding of the system's functionality. Additionally, a maximum delay length could be set for large recipient lists.

[10] This feature was first suggested by Ruoti et al. [16], but was not actually implemented.



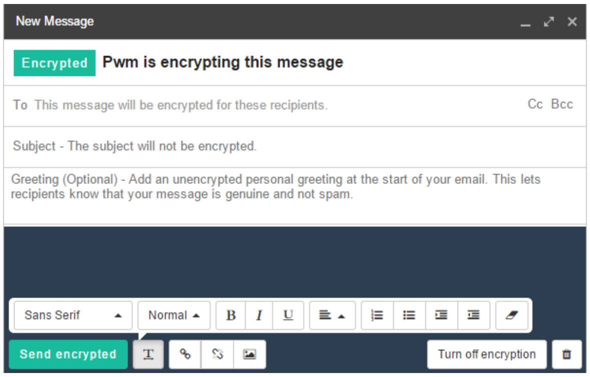
Sensitive Information - This is a plaintext greeting giving context to the message. This message has been encrypted using Pwm

Figure 2. An encrypted email in the inbox



Pwm is not encrypting this message

Figure 3. Gmail compose interface before enabling encryption



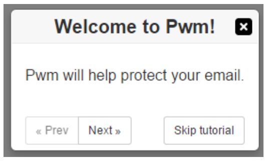
Pwm is encrypting this message

To This message will be encrypted for these recipients.

Subject - The subject will not be encrypted.

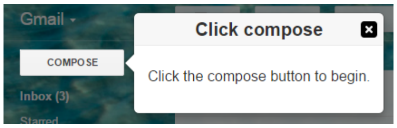
Greeting (Optional) - Add an unencrypted personal greeting at the start of your email. This lets recipients know that your message is genuine and not spam.

Figure 4. Pwm's compose interface



Pwm will help protect your email.

Figure 5. Style of the tutorial window



Click the compose button to begin.

Figure 6. Tutorial waiting for action from the user

Several of the problems encountered by participants in the original studies could have been avoided if they had viewed the included video tutorial. As participants were clearly uninterested in reading the instructions or watching a video on the Pwm website, we replaced these with integrated, context-sensitive tutorials in Pwm 2.0. These tutorials provide step-by-step instructions on how to use Pwm 2.0, are displayed the first time a user performs a new action, and are shown directly to the side of the interface the user is currently using. Each step in the tutorials uses simple language and instructs the user about a single feature of Pwm 2.0 (Figure 5). Some tutorial steps require explicit action from users before they can move on to the next step, helping users internalize correct behavior (Figure 6). The tutorials are as follows:

1. **Introduction.** This tutorial is shown to users the first time they run Pwm 2.0. It informs users that Pwm 2.0 will help protect their email and tells them how to identify whether Pwm 2.0 is running.

2. **Read.** The first time users read an encrypted email message, they are shown this tutorial. The tutorial shows users how to identify an encrypted email, explains the plaintext greeting, and identifies which portions of the email message were encrypted. Additionally, it informs them that email messages encrypted with Pwm 2.0 are provided authenticity and confidentiality (even from Gmail).

3. **Compose.** The first time users compose an email message while Pwm 2.0 is running, they are given the option to watch a tutorial describing how to compose encrypted email. This tutorial teaches users the correct order of operations for composing an encrypted email. It also informs them about who can read their message, the purpose of the optional greeting, and where to type sensitive information.

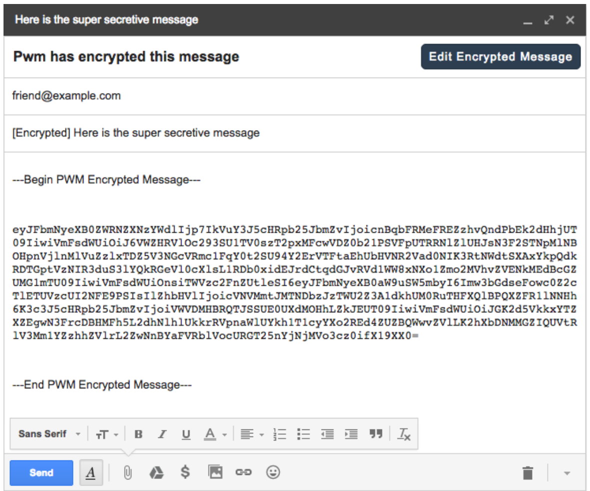
Look and Feel

We designed a new website for helping users install Pwm 2.0.[11] This website has a more professional look and feel than the original Pwm website. This change was made in reaction to user comments that they decide how much they trust a tool based on how professional the tool's website looks. To make Pwm 2.0 also feel more professional, we standardized its look and feel to use the same colors and styles as the website.

Automatic and Manual Encryption

The original version of Pwm automatically encrypts and sends email as soon as the user clicks *Send encrypted* (Figure 4). We created a manual encryption version that splits this operation into two distinct steps. In the first step, instead of clicking the *Send encrypted* button, the user instead clicks the *Encrypt* button. Upon doing so, the user's email message is encrypted, and both the instructions for decrypting the email message and the encrypted ciphertext are shown to the user inside Gmail's compose interface (Figure 7). This allows the user to confirm that encryption has actually taken place. In the second step, the user then clicks Gmail's *Send* button to send the encrypted email.

[11] `https://pwm.byu.edu`



Here is the super secretive message

Pwm has encrypted this message

friend@example.com

[Encrypted] Here is the super secretive message

---Begin PWM Encrypted Message---

```
eyJFbmNyeXB0ZWRNZXNzYWdlIjp7IkVuY3J5cHRpb25JbmZvIjoicnBqbFRMeFREZzhvQndPbEk2dHhjUT
09IiwiVmFsdWUiOiJ6VWZHRVlOc293SU1TV0szT2pxMFcwVDZ0b21PSVFpUTRRNlZlUHJsN3F2STNpMlNB
OHpnVjlnMlVuZzlxTDZ5V3NGcVRmc1FqY0t2SU94Y2ErVTFtaEhUbHVNR2Vad0NIK3RtNWdtSXAxYkpQdk
RDTGptVzNIR3duS3lYQkRGeVl0cXlsL1RDb0xidEJrdCtqdGJvRVdlWW8xNXo1Zmo2MVhvZVENkMEdBcGZ
UMGlmTU09IiwiVmFsdWUiOnsiTWVzc2FnZUtleSI6eyJFbmNyeXB0aW9uSW5mbyI6Imw3bGdseFowc0Z2c
TlETUVzcUI2NFE9PSIsIlZhbHVlIjoicVNVMmtJMTNDbzJzTWU2Z3AldkhUM0RuTHFXQlBPQXZFRllNNHh
6K3c3J5cHRpb25JbmZvIjoiVWVDMHBRQTJSSUE0UXdMOHhLZkJEUT09IiwiVmFsdWUiOiJGK2d5VkkxYTZ
XZEgwN3FrcDBHMFh5L2dhNlhlUkkrRVpnaWlUYkh1T1cyYXo2REd4ZUZBQWwvZVlLK2hXbDNMMGZIQUVtR
lV3MmlYZzhhZVlrL2ZwNnBYaFVRblVocURGT25nYjNjMVo3cz0ifX19XX0=
```

---End PWM Encrypted Message---

Figure 7. Ciphertext shown to users after manual encryption

METHODOLOGY

This section gives an overview of our IRB-approved user study that evaluated Pwm 2.0. This study was also used to run a between-subjects A/B test comparing automatic and manual encryption. The data from this study is available at `https://uist2016.isrl.byu.edu`.

We were careful to design the study to test a generic secure email system, rather than specifically testing Pwm 2.0. This approach helps us to feel more confident that participants were interacting with Pwm 2.0 in a realistic fashion. To help with this goal, we included researchers who are not involved in our secure email research in designing the scenarios and tasks. After creating scenarios and tasks that were acceptable to all parties, we then conducted a pilot study with three participants. We did not identify any significant flaws during the pilot study.

Study Setup

The study ran for two weeks (February 17, 2015 through March 3, 2015) and included 52 participants that were randomly assigned to test either the automatic or manual encryption version of Pwm 2.0.[12] Participants took 30 to 60 minutes to complete the study and were compensated $10 USD for their efforts.

Studies were conducted in a room dedicated for this study. When participants first entered the room, they were read a brief introduction to the study. All subsequent instructions were written and provided either via a printed information sheet or via email. Participants completed all tasks on a virtual machine, ensuring that the computer started in the exact same state for all participants. We also recorded participants' screens.

Two study coordinators were involved in the study. One coordinator sat in the same room as the participant, and was instructed to avoid prompting participants and to only assist participants if they had not made any progress within five minutes. The second coordinator sat in another room and corresponded with the participant over email as part of the study tasks.

Quality Control

One participant was unable to use Pwm 2.0 with their Gmail account. This participant had a special Gmail configuration that was not compatible with our tool. While this participant did complete the study using a temporary email account we provided them, we were worried that their increased interactions with the study coordinator might bias their responses, and so we discarded their results. For the remainder of this work, we report results based on the 51 remaining participants.[13]

Participant Demographics

We recruited Gmail users for our study at a local university. Participants were evenly split between genders: male (25; 49%), female (26; 51%). Participants skewed young: 18–24 years old (45; 88%), 25–34 years old (3; 6%), 35–44 years old (2; 4%), 55 years or older (1; 2%).

We distributed posters broadly across campus to avoid biasing our results by any particular major. All participants were affiliated with the university,[14] with the overwhelming majority being undergraduate students: undergraduates (44; 86%), graduates (2; 4%), faculty and staff (5; 10%). Participants had a variety of majors, including both technical and non-technical majors. No major was represented by more than three participants, with the vast majority having only a single participant.

Most participants indicated they logged into Gmail on the web to check their email frequently: many times a day (39; 76%), once a day (3; 6%), 2–3 times a week (2; 4%), once a week (2; 4%), 2–3 times a month (2, 4%).

Scenario and Task Design

During the course of the study, participants were given two scenarios to complete: (1) being hired for a new job, and (2) sending credit card information to a spouse. Prior to beginning each scenario, participants were provided with a written description of the scenario and information that participants should send in place of their own personal information.[15] Participants were asked to treat this sensitive information with the same care as they would treat their own information.

[12] Our power analysis ($\alpha = 0.05$, $\beta = 0.80$) indicated that we would need 23 participants in each treatment group to reasonably detect differences in SUS scores and task completion time. In contrast, our power analysis revealed that detecting even medium sized (10%) differences in our proportional measures would require 387 participants in each treatment group (far beyond our group's ability to recruit participants). As such, the proportional measures should only be evaluated as trends, and not as statistically significant.

[13] After analyzing the remaining data, we reviewed the removed participant's results and found that they were in line with the results we analyzed.

[14] We did not require this affiliation.

[15] We took this approach to avoid situations where sensitive information might have been leaked to Gmail.

For each scenario we designed realistic tasks that used as many email features as possible. Participants completed tasks in the order shown below. Tasks were completed entirely using email, and participants used their own email accounts. If participants accidentally sent sensitive information without encryption, they were notified of their mistake in an email and asked to resend the information using encryption.

Being hired for a new job

Participants were told that they had recently returned from an interview at National Citadel, a fictitious company created for this study.

Task 1. Participants received an email from National Citadel containing instructions on how to be reimbursed for expenses from their recent interview. Participants were told to send their Social Security number and a picture of their receipts. They were informed that this information must be encrypted as per National Citadel's policies. This email also instructed users to set up and use Pwm 2.0 to encrypt their email messages.

This task was designed to test whether participants were able to set up and use Pwm 2.0 using only instructions that might be reasonably expected from a company requesting information to be encrypted.

Task 2. Participants were asked to simulate several weeks passing after the completion of task by closing their browser and then reopening Gmail. Participants were then sent an encrypted email that contained an offer letter. They were asked to reply with their acceptance and to *CC* their new manager.

This task was designed to test whether participants would remember how to enable Pwm 2.0. It also tested whether they could use Pwm 2.0 to *CC* a new recipient.

Task 3. Participants were sent an email instructing them to email information to a background check company. They were instructed to encrypt the information.

This task was designed to test whether users could enable encryption, either by forwarding the request for information or by composing a new email message.

Task 4. Participants were instructed to send bank account information to National Citadel's payroll department. They were *not* reminded to encrypt this information.

This task was designed to test whether users would remember to encrypt information if they were not explicitly prompted to do so. Unlike the preceding tasks, if they sent information without encryption, we still considered this task complete.

Sending credit card information to a spouse

Participants were told that their spouse had texted them asking for login information to a credit card website.

Task 5. Participants were instructed to send login information to their "spouse" using encrypted email. Participants were told that their spouse had never before used secure email, so they should take whatever steps they felt necessary to ensure their spouse could read the encrypted email.

This task was designed to see how participants would induct a new person into using secure email. Regardless of what instructions were sent, we considered this task complete when the information had been sent.[16]

Task 6. Participants received another email from their spouse asking them for additional credit card information. This request was *not* encrypted and did *not* instruct the participant to send the additional credit card information encrypted.

This task was designed to test whether users would remember to encrypt information if they were not explicitly prompted to do so. Unlike most of the preceding tasks, if they sent information without encryption, we still considered this task complete.

Using the video recording of participants' screens, we measured how long they took completing each task and how often they sent sensitive email without encryption (i.e., made a mistake).

Study Questionnaire

We administered our study using the Qualtrics web-based survey software. Before beginning the survey, participants were read an introduction by the study coordinator and asked to answer a set of demographic questions. Participants then completed the six study tasks, following which they were asked to complete a questionnaire regarding their experience.

To measure perceived usability, we had participants complete the ten System Usability Scale (SUS) questions [4, 5]. Answers to these questions are used to derive each version's SUS score, a single numeric score from 0, the least usable, to 100, the most usable, that provides a rough estimate of the version's overall usability. Recent research has shown that SUS scores are effective for comparing systems across different study populations and that SUS gives more reliable results than other usability metrics [23, 19, 17].

After completing the SUS questions, participants were asked several questions regarding whether they would want to use Pwm 2.0 in their day-to-day lives. We then asked participants questions to examine how well they understood the cryptographic properties of Pwm 2.0. To test understanding of confidentiality, they were asked to indicate which parties could read a message encrypted with Pwm 2.0. Similarly, participants were asked whether Pwm 2.0 provided authenticity and integrity for secure email messages composed with Pwm 2.0. Each question was asked in language that would be approachable to users and did not require a technical

[16] Pwm 2.0 includes instructions by default, and since they were sufficient to help the participant start using Pwm 2.0, it was reasonable to assume that the participant believed they would be sufficient to help their spouse start using Pwm 2.0.

	Count	Mean	Standard Deviation	Confidence Interval ($\alpha = 0.05$)
Automatic	27	79.1	9.6	±3.69
Manual	24	81.1	9.0	±3.60
Overall	51	80.0	9.2	±2.52
Original Pwm Study	72	74.2	13.7	±3.16

Table 1. SUS scores

background. Participants were given the option to indicate that they were unsure whether a given property was provided by Pwm 2.0.

Limitations

While our study included students with a diverse set of majors and technical expertise, it may not be representative of non-student populations. Likewise, Gmail users may also not be representative of the general population's preferences regarding secure email. Our study is short-term and is not necessarily representative of how participants would use secure email over a longer period of time. Further studies could address these limitations.

Our study is a lab study and has limitations common to all studies run in a trusted environment [13, 22]. While there are indications that some participants treated the provided sensitive information as they would their own (e.g., refusing to email the provided social security number), there is no guarantee that participants' reactions mimic their real life behaviors. Additionally, our studies did not test participants' ability to resist attacks.

RESULTS

In this section we report the quantitative results from our user study.

Usability

We evaluated Pwm 2.0 using the System Usability Scale (SUS). The automatic encryption version of Pwm 2.0 had a SUS score of 79.1 and the manual encryption version had a SUS score of 81.1. Further breakdown of the SUS scores, along with SUS scores from the original Pwm study [16], can be found in Table 1. The difference between manual encryption was not statistically significant (two tailed student t-test, unequal variance—$p = 0.43$). The difference between Pwm 2.0's overall SUS score (80.0) and Pwm's SUS score (74.2) is statistically significant (two tailed student t-test, unequal variance—$p < 0.01$).

To give greater context for Pwm 2.0's SUS score, we leverage the work of several researchers. Bangor et al. [3] analyzed 2,324 SUS surveys, and derived a set of acceptability ranges that describe whether a system with a given score is acceptable to users in terms of usability. Bangor et al. also associated specific SUS scores with adjective descriptions of the system's usability. Using this data, we generated ranges for these adjective ratings, such that a score is correlated with the adjective it is closest to in terms of standard deviations. Sauro et al. [18] also analyzed SUS scores from Bangor et al. [2], Tullis et al. [23], and their own data. They calculate the percentile values for SUS scores and assign letter grades based on percentile ranges.

	Count	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Total
Automatic	27	4:06	4:00	3:01	1:20	4:01	0:54	17:22
Manual	24	5:10	3:48	3:03	1:08	3:44	0:50	17:43
Overall	51	4:37	3:54	3:02	1:14	3:52	0:52	17:32

Table 2. Average task completion times (min:sec)

Using these contextual clues, Pwm 2.0's SUS score of 80.0 is rated as having "excellent" usability and given an "A" grade. It is in the 87th percentile for system usability.

Task Completion Times

For each task, we calculated the average time between when the user could start a task and when they finished that task. These are shown in Table 2. For Tasks 2–6, and for the study as a whole, any differences between manual and automatic encryption were negligible. Task 1 had a statistically significant difference (two tailed student t-test, unequal variance—$p = 0.04$), but we caution against overemphasizing this result. First, while the difference is significant, the 95% confidence interval indicates that the actual effect size might be small (64 ± 60 seconds). Second, any differences in task completion times are amortized over the remaining tasks. Finally, a single statistically significant difference could arguably be the result of alpha inflation [19].

Understanding

We asked three questions to determine whether each participant understood the confidentiality, authenticity, and integrity properties provided by Pwm 2.0. The responses indicated whether each participant correctly understood the principle, had some misunderstanding, or was unsure. Overall, our data suggests that over 50% of users would likely understand the security provided by Pwm 2.0 (Adjusted-Wald binomial confidence interval [19], $\alpha = 0.05$—Confidentiality–83% ± 10%, Authenticity–62% ± 13%, Integrity–75% ± 12%). In all cases, the differences between manual and automatic encryption were not statistically significant.

The proportion of users who understood how Pwm 2.0 was protecting their email was much higher than the rate reported in the original Pwm study.[17] Still, it is interesting that even with tutorials that explicitly instruct par-

[17] We do not calculate statistical significance for these two proportions as they were derived from different questions. Note that the understanding questions for Pwm 2.0 were stricter

	Count	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Total
Automatic	27	0	0	1	0	0	0	1
Manual	24	1	0	0	1	0	0	2
Overall	51	3	0	2	1	0	0	6

Table 3. Number of participants who sent sensitive information without encryption

ticipants on these three cryptographic properties, there are still a small number of participants who were unsure how their messages were protected. This indicates that more work needs to be done to determine how to effectively teach users about these properties.

Avoiding Mistakes

During the study, we recorded all instances of a participant taking an action that deviated from the study parameters. Using this data, we identified several instances where a participant's actions leaked sensitive information. These results are reported by task in Table 3.

During the study, two participants sent sensitive information without ever running Pwm 2.0, and another refreshed the Gmail page but did not restart Pwm 2.0 before sending sensitive information. In all three cases, the mistakes were caused by the misuse of the tool and occurred before participants had ever seen any differences in the two versions of Pwm 2.0. For this reason, they are reported in the overall number of mistakes, but not under automatic or manual encryption. The differences between mistakes related to manual and automatic encryption are negligible.

Overall the rate of mistakes was extremely low (2%, 6 tasks out of 306). This is lower than the mistake rate reported in the original Pwm study of 10%, and the difference is statistically significant ($N-1$ two-proportion test for comparing two independent proportions–$p = 0.001$).

Tutorials

We noted the number of participants that completed each tutorial. The video recording for one participant's session was corrupted, and we were unable to determine if they had completed the tutorials. The data from the remaining 50 participants indicate that participants completed a significant number of tutorials: introductory tutorial (46; 92%), tutorial on reading secure email (46; 92%), tutorial on composing secure email (27; 54%). This is in stark contrast to the original Pwm study where nearly all participants ignored both the setup instructions and a video tutorial included on the Pwm website [16]. This result indicates that participants are willing to watch tutorials, though two criteria seem to be crucial: they appear in-page as participants need them and they contain simple and direct wording.[18]

Acceptability of Pwm 2.0

Overall, participants responded very positively to the prospect of using Pwm 2.0 in their own lives: 82% of participants agreed with the statement "I want to start using Pwm"[19] (81% ± 11%).[20] 73% of participants agree that "I would use Pwm with my friends and family" (71% ± 12%). 92% of participants agreed that "My friends and family could easily start using Pwm" (90% ± 8%). There were no significant differences between participants who used the automatic or manual versions of Pwm 2.0.

Due to the nature of the study, it is likely that participant responses were overly positive, and that the proportions are somewhat high. Nevertheless, this data suggests that if introduced to Pwm 2.0, a non-negligible number of participants would want to continue using it, and would feel comfortable using it to send encrypted email to their friends and family.

DISCUSSION

In this section we discuss noteworthy items, including participant experiences, opinions, and preferences regarding secure email and lessons learned from improving Pwm. We refer to participants here as R1–R52, with the number corresponding to the order in which they participated in our study.

Automatic and Manual Encryption

Our results indicate no significant differences between automatic and manual encryption. As such, we hypothesize that the effect observed in the original Pwm study was due to confounding factors between the two systems being studied (i.e., automatic encryption–Pwm, manual encryption–MessageProtector).[21] Alternatively, it is possible that manual encryption would have benefited Pwm's original implementation, but that the modifications we made while creating Pwm 2.0 were sufficient to provide those same benefits.

Pwm 2.0's High Usability

Pwm 2.0's SUS score of 80 falls in the 87th percentile for system usability [19], and is the highest score for secure email systems [16, 14]—Mailvelope (35, 4th percentile),

than those used in the original Pwm study, emphasizing how much better understanding in Pwm 2.0 is likely to be.

[18] We believe fewer participants watched the compose tutorial, which appeared the first time participants clicked on Gmail's "Compose" button, because it was not clear they needed to see it at this point and because the tutorial drew attention to the option that they could skip it. Better tutorial design could possibly address this issue.

[19] In the study, Pwm 2.0 was referred to as just Pwm.

[20] The confidence intervals reported here were calculated using the Adjusted-Wald binomial confidence interval, $\alpha = 0.05$.

[21] This hypothesis is supported by similar results in Atwater et al.'s work [1].

Tutanota (52, ~15$^{\text{th}}$), Encipher.it (61, ~30$^{\text{th}}$), Voltage (62, ~32$^{\text{nd}}$), Virtru (72, 63$^{\text{rd}}$), original Pwm (74, 70$^{\text{th}}$).[22]

Additionally, most participants understood how Pwm 2.0 was protecting their messages, only rarely made mistakes, and were interested in using Pwm 2.0 with their friends and family. These positive opinions were also reflected in numerous positive qualitative responses. For example, R26, R39, and R42 expressed, respectively,

> *"It was very concise and user friendly and did not require esoteric knowledge to operate. I would definitely feel more comfortable sending sensitive information over email if I were using Pwm versus just sending it via an email provider."*

> *"I liked how I could encrypt sensitive information like bank account information, credit card, and other things. It wasn't that hard to use. I didn't have to download anything; all I had to do was just save a bookmark and then click on it. It was really easy to use."*

> *"I liked how it made encrypting important information so easy. The tutorial was fast and easy. I like that it is easy and convenient to use with day to day emails. I liked that the background was blue so I knew when it was encrypting."*

Delayed Encryption

While we did not specifically collect data regarding users' opinions toward our delayed encryption mechanism, we note that not a single participant complained about encryption being too fast. This is especially significant when compared to the original Pwm study, where over 33% of participants self-reported that encryption was too fast and therefore untrustworthy. The difference between these two proportions is statistically significant ($N-1$ two-proportion test for comparing two independent proportions–$p < 0.001$). While this may not mean that users' concerns regarding the speed of encryption were eliminated, it is strongly suggestive that they were reduced enough not to be a distraction.

Mixed mode email

The optional plaintext greeting that can accompany email encrypted by Pwm 2.0 is intended to help email senders give confidence to recipients who have never used Pwm 2.0 that the email is authentic and not spam. Surprisingly, during our study we noted that a small, but significant number of users would include a plaintext greeting in many of their encrypted email messages. For example, in Task 4 eight participants (8; 16%) including a greeting stating that they were sending their direct deposit information. Interestingly, this was actually a feature that seven participants (7; 14%) listed as one of their favorite features of Pwm 2.0. R12 and R51 stated, respectively,

> *"It wasn't rigid, I could write part of a message and have the other parts encrypted if I wanted. It was very clear what was encrypted and what wasn't [...]"*

> *"That it lets me chose when to encrypt and when not to. It's also nice to have the option of writing a message before the encrypted part so others know it's not spam."*

Look and Feel

Participants indicated that having a color-scheme that was distinct from Gmail helped them more easily understand what information was encrypted and what information was in the clear. This intuitive understanding potentially helped participants avoid mistakenly entering sensitive information where it would not be protected. Additionally, participants mentioned that it helped them feel more confident in the system. Thus it is clear that when designing tightly integrated systems, it is important to have a distinct look and feel.

Instructing New Users

Even though Pwm 2.0 automatically inserts instructions on how to setup and use Pwm 2.0 in every encrypted message, most participants were unaware of this. During Task 5, participants would often spend several minutes trying to open an old Pwm 2.0 message to grab the instructions, only to have the message immediately decrypted and the instructions disappear. It would be helpful to make it more clear that these instructions are always included or to add an option that allows users to explicitly add these instructions.

CONCLUSION

Our modifications to Pwm's interface have significantly increased its usability and security. In our user study, Pwm 2.0 is rated with an 80.0 SUS score, the highest reported SUS score for secure email systems. Over 80% of participants expressed a desire to begin using Pwm, and over 90% of participants believed that their friends and family could easily start using Pwm. Pwm 2.0 also helped participants avoid mistakenly sending their sensitive data in the clear.

This work represents an important step toward providing usable, secure email encryption. While Pwm 2.0 has lower theoretical security than PGP-based systems, it provides higher practical security for an ordinary user. In studies of PGP-based systems, most users were unable to encrypt their emails [26, 21, 15], but in Pwm 2.0 all users are able to encrypt their emails. While there is little evidence that PGP-based systems will ever be sufficiently usable for the masses [12], techniques from Pwm 2.0 could be applied to PGP-based systems to raise their relative usability.

ACKNOWLEDGMENT

We would like to thank Mike Jones and Dan Olsen for providing feedback on this paper.

[22] Neither Message Protector [16] or Atwater et al.'s system [1] had functioning key management, so their scores are not reported here.

REFERENCES

1. Atwater, E., Bocovich, C., Hengartner, U., Lank, E., and Goldberg, I. Leading Johnny to water: Designing for usability and trust. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, USENIX Association (2015), 69–88.
2. Bangor, A., Kortum, P., and Miller, J. An empirical evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction 24*, 6 (2008), 574–594.
3. Bangor, A., Kortum, P., and Miller, J. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies 4*, 3 (2009), 114–123.
4. Brooke, J. SUS—a quick and dirty usability scale. In *Usability Evaluation in Industry*. CRC Press, 1996.
5. Brooke, J. SUS: A retrospective. *Journal of Usability Studies 8*, 2 (2013), 29–40.
6. Durumeric, Z., Adrian, D., Mirian, A., Kasten, J., Bursztein, E., Lidzborski, N., Thomas, K., Eranti, V., Bailey, M., and Halderman, J. A. Neither snow nor rain nor MITM. . . : An empirical analysis of email delivery security. In *Fifteenth ACM Internet Measurement Conference (IMC 2015)*, ACM (2015), 27–39.
7. Fahl, S., Harbach, M., Muders, T., Smith, M., and Sander, U. Helping Johnny 2.0 to encrypt his Facebook conversations. In *Eighth Symposium on Usable Privacy and Security (SOUPS 2012)*, ACM (2012), 11.
8. Foster, I. D., Larson, J., Masich, M., Snoeren, A. C., Savage, S., and Levchenko, K. Security by any other name: On the effectiveness of provider based email security. In *Twenty-Second ACM SIGSAC Conference on Computer and Communications Security (CCS 2015)*, ACM (2015), 450–464.
9. Garfinkel, S. L. Email-based identification and authentication: An alternative to PKI? In *Twenty-Fourth IEEE Symposium on Security and Privacy (S&P 2003)*, IEEE Computer Society (2003), 20–26.
10. Garfinkel, S. L., and Miller, R. C. Johnny 2: A user test of key continuity management with S/MIME and Outlook Express. In *First Symposium on Usable Privacy and Security (SOUPS 2005)*, ACM (2005), 13–24.
11. Holz, R., Amann, J., Mehani, O., Wachs, M., and Kaafar, M. A. TLS in the wild: An internet-wide analysis of TLS-based protocols for electronic communication. In *Twenty-Fourth Network and Distributed System Security Symposium (NDSS 2016)*, The Internet Society (2016).
12. Marlinspike, M. Gpg and me, 2015. Accessed Sep 25, 2015. `http://www.thoughtcrime.org/blog/gpg-and-me/`.
13. Milgram, S., and Van den Haag, E. *Obedience to Authority*. Ziff–Davis Publishing Company, 1978.
14. Ruoti, S., Andersen, J., Heidbrink, S., ONeill, M., Vaziripour, E., Wu, J., Zappala, D., and Seamons, K. "We're on the same page": A usability study of secure email using pairs of novice users. In *Thirty-Fourth ACM Conference on Human Factors and Computing Systems (CHI 2016)*, ACM (2016), 4298–4308.
15. Ruoti, S., Andersen, J., Zappala, D., and Seamons, K. Why Johnny still, still can't encrypt: Evaluating the usability of a modern PGP client, 2015. arXiv preprint arXiv:1510.08555.
16. Ruoti, S., Kim, N., Burgon, B., Van Der Horst, T., and Seamons, K. Confused Johnny: When automatic encryption leads to confusion and mistakes. In *Ninth Symposium on Usable Privacy and Security (SOUPS 2013)*, ACM (2013).
17. Ruoti, S., Roberts, B., and Seamons, K. Authentication melee: A usability analysis of seven web authentication systems. In *Twenty-fourth International Conference on World Wide Web (WWW 2015)*, ACM (2015), 916–926.
18. Sauro, J. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. Measuring Usability LLC, 2011.
19. Sauro, J., and Lewis, J. R. *Quantifying the User Experience: Practical Statistics for User Research*. Elsevier, 2012.
20. Shamir, A. Identity-based cryptosystems and signature schemes. In *Fourteenth International Cryptology Conference (Crypto 1984)*, Springer (1984), 47–53.
21. Sheng, S., Broderick, L., Koranda, C., and Hyland, J. Why Johnny still can't encrypt: Evaluating the usability of email encryption software. In *Poster Session at the Symposium On Usable Privacy and Security* (2006).
22. Sotirakopoulos, A., Hawkey, K., and Beznosov, K. "I did it because I trusted you": Challenges with the study environment biasing participant behaviours. In *Usable Security Experiment Reports Workshop at the Symposium On Usable Privacy and Security* (2010).
23. Tullis, T. S., and Stetson, J. N. A comparison of questionnaires for assessing website usability. In *Usability Professional Association Conference*, Usability Professionals Association (2004), 1–12.

24. Van Der Horst, T. W., and Seamons, K. E. Simple authentication for the web. In *Third International Conference on Security and Privacy in Communications Networks (SecureComm 2007)*, IEEE Computer Society (2007), 473–482.

25. van der Horst, T. W., and Seamons, K. E. Encrypted email based upon trusted overlays, March 2009. US Patent 8,521,821.

26. Whitten, A., and Tygar, J. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Eighth USENIX Security Symposium (USENIX Security 1999)*, USENIX Association (1999), 14–28.