2014-03-07

Pwm: A Secure Webmail System Designed for Easy Adoption

Benjamin W. Burgon
Brigham Young University - Provo

Pwm (Private WebMail): A Secure Webmail System Designed for Easy Adoption

Benjamin W. Burgon

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

Master of Science

Kent E. Seamons, Chair
Dan Olsen
Quinn Snell

Department of Computer Science

Brigham Young University

March 2014

ABSTRACT

Pwm (Private WebMail): A Secure Webmail System Designed for Easy Adoption

Benjamin W. Burgon
Department of Computer Science, BYU
Master of Science

None of the three largest webmail service providers (serving over 1 billion users) support end-to-end message encryption. Encrypted email has never seen mass adoption because it is prohibitive for non-experts to use. Private WebMail (Pwm) is our extension to popular webmail systems that lets users easily encrypt sensitive messages without having to first contact the recipient and share information. It is designed to spread quickly in a grassroots fashion so that a user receiving their first encrypted message can quickly and easily start using the system. This thesis describes the design and implementation of Pwm, then measures its usability through analysis and a user study.

Keywords: secure email, webmail, end-to-end encryption, usable security

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Kent Seamons, from whom I have learned a great deal, for his guidance and support of my work throughout my graduate program at Brigham Young University. I would also like to acknowledge my fellow students in the Internet Security Research Lab with whom I greatly enjoyed working and collaborating. Finally, I would like to acknowledge my family for the support they provided to me throughout my work on this project.

Table of Contents

Chapter 1

Introduction

As it has traditionally been defined and implemented, email is generally not a secure communications medium. This is not to say that there is no security available but that existing security measures are often inadequate or are not guaranteed to be available. Almost all standard email protocols define processes for authenticating senders or recipients via passwords or other authentication schemes. Most also allow for, but do not require, confidential transmission of messages using cryptographic protocols such as TLS. While these standards can reduce the risk of eavesdropping or interception during transmission, these precautions are not mandated. In fact, specifications for the use of TLS over SMTP dictate that mail servers with public DNS listings must not require TLS encryption for incoming mail [**?**]. Furthermore, there are no requirements for protecting the contents of messages stored on intermediate relay servers or users' server-side mailboxes or local mail clients.

1.1 Historically Poor Usability

In the decades since email's inception, numerous attempts have been made to increase the security of email systems. Some of the most notable among these attempts are Privacy Enhanced Mail (PEM) [**?**], Pretty Good Privacy (PGP) [**?**], and Secure/Multipurpose Internet Mail Extensions (S/MIME) [**?**]. Multiple studies have shown that these systems have significant usability weaknesses:

- Difficulties understanding and using public/private keys [**?**]
- Difficulty and cost of obtaining certificates from a CA [**?**]

- General need for prior communication or coordination between senders and recipients before secure communication can occur:
  - General chicken and egg problem: Senders want to send secure content, but need action from desired recipient
  - Recipients are unmotivated to take necessary action because they have never received an encrypted message

These usability issues often put the use of secure email beyond the limits of most people's technical abilities and patience.

1.2 Identity Based Encryption

At the heart of many of the usability issues documented in prior secure email solutions lies the need for users to establish and reliably exchange public parameters before secure communication can begin. Shamir [**?**] first proposed eliminating this step through an approach known as Identity Based Cryptography. Perhaps the most notable implementation of this concept is in Boneh and Franklin's Identity Based Encryption (IBE) [**?**]. By eliminating the need for *a priori* key creation and distribution, many of the most severe usability issues become irrelevant.

In IBE, each user's public key is derived from the user's email address or another unique, publicly available identifier (e.g., phone number). By using this publicly available information to derive the public key, IBE eliminates the need for users to create and manage public/private key pairs. In fact, it even makes it possible for a sender to encrypt a message for a recipient who has never used IBE. By enabling Alice to encrypt a message using Bob's public key even though Bob has never created or published a public key, IBE has eliminated the primary form of prior communication required by other public key cryptography solutions.

In addition to reducing usability problems, eliminating the need for prior communication can help facilitate the adoption of IBE since it frees senders to send encrypted messages

to email users who have not yet adopted IBE. These recipients are thereby made aware of IBE, motivated to get an IBE client to read the message, and, once they have the client, are also enabled to send messages to other potential new users. Ideally, this openness can lead to a grass-roots adoption of IBE as more email users adopt IBE after receiving encrypted messages from prior adopters.

It is important to note that sending uninitiated email users encrypted emails without prior notice can lead to confusion when users receive ciphertext that they are unable to read without an email client that supports IBE. It appears that no prior research has addressed ways to solve this issue, and a primary goal of Pwm is to address this shortcoming.

1.3 Webmail Issues

Today, many individuals and organizations are migrating to webmail-based email systems. A webmail system is an email service in which end users read, compose, send, and receive email messages using a web-based interface accessed via a browser.

Webmail users often rely on hosted service providers for a web-based client to send and receive emails. Since web-based clients rarely provide functionality for encrypting or signing messages, users seeking additional security typically must switch to an entirely new webmail service, possibly changing email addresses in the process.

This dependency on a service provider for client-side security functionality can be disempowering for users who need to communicate private or sensitive information. Several projects have sought to bring existing security solutions to webmail systems, but little has been done to evaluate their usability.

1.4 Thesis Contribution

This thesis presents Pwm (Private WebMail), a system designed to integrate with existing webmail services to provide privacy for webmail users. Pwm addresses the complex setup and key management tasks that have been shown to challenge users of earlier secure email

solutions. By removing these difficulties, Pwm lowers the bar for adoption by new users. Since Pwm completely eliminates the need for *a priori* setup by recipients, Pwm can be adopted in a grassroots manner by new users when they receive their first encrypted message [**?** **?**]. By taking the approach inspired by IBC to eliminate the need for prior setup and communication, Pwm inherits the potential for unprepared first-time users to receive encrypted messages without warning or explanation.

This thesis describes the design and implementation of Pwm with a specific focus on usability for new users. It also includes two user studies to demonstrate Pwm's effectiveness in overcoming usability challenges common to earlier secure email systems. These are the first comprehensive studies to measure the usability of a secure webmail system by non-technical users with no *a priori* setup by recipients.

Chapter 2

Related Work

2.1 Traditional Email Security

2.1.1 Pretty Good Privacy (PGP)

PGP is an approach to encryption originally developed by Phil Zimmerman beginning in 1991 [**?**]. One of PGP's distinguishing features is its communal approach to key verification and management. Rather than relying on a trusted certificate authority (CA) to verify the identity of key holders, PGP relies on a "Web of Trust" model in which existing PGP users sign the public keys of new users after taking measures to verify that the new user's identity is accurate. The means by which new users' identities are to be verified are often left to the discretion of the verifier.

Usability issues with PGP are documented and discussed in Whitten and Tygar's seminal paper "Why Johnny Can't Encrypt" [**?**]. This paper describes a study of users in a hypothetical political campaign where they had to create a keypair for their own use and perform a series of email-based tasks requiring them to encrypt and decrypt several messages. Of the users who succeeded in eventually sending an encrypted message, seven out of eleven encrypted the message they were sending with their own public key (meaning only they, themselves could decrypt it) and three sent the plaintext of their private messages. In summary of the study, Whitten and Tygar stated "We conclude that PGP 5.0 is not usable enough to provide effective security for most computer users, despite its attractive graphical user interface, supporting our hypothesis that user interface design for effective security remains an open problem" [**?**]. Although several new versions of PGP have been released

since the time of the study, the core model, and hence many of the usability challenges, of PGP remains the same.

2.1.2 S/MIME

S/MIME is an asymmetric encryption system in which all users' public keys are contained in certificates signed by trusted certificate authorities. Unfortunately, the task of acquiring and using signed certificates from a certificate authority (CA) can be difficult for users. While some CAs offer free or low cost certificates, others charge fees that are relatively high and can be prohibitive for casual users. The process of acquiring signed certificates from a CA is further complicated by the lengthy application process that some CAs require users to complete [**?**].

One significant usability improvement S/MIME offers is in public key distribution. When a message is signed with S/MIME, the signer's public certificate is included as a message attachment. This makes the certificate available for recipients to immediately verify the contents of the message. In addition to being transparent to users, including the sender's public certificate eliminates the need for prior communication before sending signed messages. However, a prior exchange is still necessary for encrypted messages since the sender must have the recipient's public certificate in order to encrypt the message. For example, if Alice wanted to encrypt a message so that only Bob could read it, she would first need to obtain Bob's public certificate either by requesting that he send it to her, or by acquiring a message that he had previously signed.

The practical usability of S/MIME has been studied and found to be lacking for many users. A 2005 study inspired by Whitten and Tygar's 1999 study found that S/MIME addressed the core difficulties of key generation faced by PGP users but did not improve user ability to properly use these credentials to achieve S/MIME's potential for reliable message security [**?**]. Additionally, while this study showed that S/MIME assisted users in verifying message signatures by including the sender's public certificate with the message it did not

address the inability for a user to obtain public certificates to encrypt a messages to recipients from whom they had not previously received a public certificate.

2.2 Identity Based Encryption

The introduction of IBE [**? ?**] has inspired a significant amount of research into various possible applications and extensions of the technology. Even among this research, little work has been done to assess the usability of IBE in protecting email. At the core of IBE's appeal for use in encrypting email is its elimination of the need for prior communication between senders and recipients before messages can be encrypted. As the name implies, derivation of keys is based upon each user's identity and an assortment of public and private parameters. The universally unique nature of email addresses makes them an ideal choice for use as user identifiers. To facilitate the derivation of public keys using the correct parameters, IBE utilizes a key escrow server. The escrow server will deliver public parameters, allowing derivation of public keys by any user. However, the escrow server requires authentication to verify a party's identity before deriving and delivering a private key.

The contributions offered by IBE show promise in lifting the burden of key generation and distribution from secure email users by making the process essentially transparent. The posibility for users to encrypt a message without any prior communication also defers the requirement for any action on the part of the recient until the encrypted message has already been delivered.

To date, no other usability studies of IBE implementations for secure email have been published.

2.3 Secure Webmail

Following the growing popularity of webmail in recent years, attempts have been made at providing security for webmail. These solutions fall within three categories: depot, dedicated, or integrated systems. Depot systems provide an interface through which recipients of

encrypted messages can authenticate themselves and read the message contents. Dedicated services provide a complete webmail service with the option to enable encryption for specific messages. Integrated systems are secure webmail tools that integrate with an existing webmail service provider's client interface. Systems providing dedicated webmail functionality for registered users and depot-like functionality for unregistered recipients are quite common.

2.3.1 Dedicated / Depot Services

HushMail

HushMail[1] is a PGP-based dedicated secure webmail system made freely available to individual users for non-commercial use. Messages sent between HushMail users are encrypted in an automatic process abstracted from the view of both the sender and recipient(s). Messages sent to recipients without HushMail accounts can also be encrypted, but require that the sender and recipient already possess a shared secret, or can establish one through some other communications channel. In such a case, the sender is asked to "[type] a secret question and answer that can only be answered by the recipients". In an instructional reference, the question "where did we go to dinner last night?" is provided as an example. While this requirement may be simple for senders and recipients who already posess shared secrets, senders who do not already share such a relationship with their intended recipients are left to their own devices to communicate the answer to their chosen secret question through some other means.

Once a message has been encrypted and sent, non-HushMail users will receive an email notifying them that they had been sent a message encrypted with HushMail and providing them with a URL or link to the HushMail message depot where they can retrieve the message. When the recipient attempts to access this URL, they will be prompted to enter the shared secret they should have established with the sender though some means

[1] http://www.hushmail.com/

beyond those provided by HushMail. Once this shared secret is authenticated, the plaintext contents of the message will be presented to the recipient.

Voltage

Voltage[2] is an implementation of IBE exhibiting characteristics of both dedicated and depot systems. Voltage is available to users as a plug-in for Outlook or through a dedicated webmail interface named Voltage SecureMail Cloud. Users of the Outlook plug-in experience a relatively seamless integration of IBE functionality with the standard Outlook interface. Webmail and other non-Outlook users can access Voltage through the dedicated Voltage SecureMail Cloud webmail interface. When a Voltage user sends an encrypted message to a recipient without Voltage, the message can be encrypted immediately since the recipient's public key is derived from their email address. Once the message has been encrypted and stored on the Voltage servers, a new message to the recipient is generated and sent. This new message contains an explanation that the recipient has been sent an encrypted message that can be retrieved at a URL hosted by Voltage. When the recipient visits the provided URL, they will be prompted to create an account with Voltage. Once this account has been created and verified, the recipient is given access to the message using a version of the dedicated webmail interface that allows them to read the message and securely reply to the sender, but not compose new messages.

2.3.2 Integrated Services

Penango (Formerly Gmail S/MIME)

Formerly called Gmail S/MIME, Penango [3] is a web browser extension that adds S/MIME support to the Gmail and Zimbra webmail interfaces, allowing functionality similar to client-side e-mail clients with S/MIME support. Penagno aims to make the tasks of signing,

[2]http://www.voltage.com/
[3]http://www.penango.com/

encrypting, verifying, and decrypting messages nearly transparent to the user. As with other S/MIME client implementations, users are left to their own devices to obtain a signed certificate from a CA which must then be imported into Penango. At the time of this writing, the latest build of the Penango extension supported Firefox and Internet Explorer 8 and 9 and was available under both paid commercial or free personal use licenses. Of the existing webmail encryption tools surveyed, Penango most closely replicates the features and user experience available to non-web-based email users in that it provides functionality for signing, encryption, decryption, and signature verification for messages so long as the user has properly created, configured and shared all necessary security certificates before attempting to use Penango.

2.4 User Study Design

Prior work has focused on issues related to effectively designing and conducting user studies such as the ones we conducted to validate Pwm's success in providing a usable solution for secure webmail.

Most notably, Sotirakopoulos et al. [**?**] discussed results of a follow-up to an earlier study of SSL warning messages. They found that past studies, including their own, had been significantly impacted by the environment in which the experiment was conducted. A notable example is that 33% of subjects reported that they ignored security warnings because they felt safe in the experimental environment. The results and suggestions in this paper influenced the design and execution of our own user study with the goal of avoiding many of the problems documented in this paper, primarily the assumption that participants in a lab-based study can be assumed to make the same risk assessments in an environment they know to be protected, as they would on their own in the real world.

Another aspect of our user studies that was influenced by prior work was our approach to quantifying user's evaluation of Pwm's usability. Brooke [**?**] developed the System Usability Scale (SUS) as a means of evaluating user's perceptions of the usability and

learnability of software systems. A detailed discussion of the nature of SUS and how we applied it to our own user studies is provided in chapter 5 of this thesis.

Chapter 3

Pwm

3.1 Design

Pwm's objective is to enable users to increase the security of their existing webmail communication tools without requiring any complicated setup or *a priori* configuration. The increased security protects the user from disclosure of private information if the user's email messages are accessed by an unauthorized third party. We have termed this level of protection to be "good enough" security as the user is protected from most threats during message delivery and storage, but may still be vulnerable to advanced attackers such as government agencies.

The following are the primary design features of Pwm that help meet its objective.

- Tight integration with existing systems by means of security overlays that augment existing webmail accounts and interfaces to provide end-to-end encryption in an environment already familiar to the user.
- Easy setup for first time recipients of an encrypted email
- Transparent key management using key escrow
- "Good Enough" protection of sensitive message contents

3.1.1 Integration with Existing Systems

Pwm uses security overlays to tightly integrate new security features into existing webmail interfaces. A security overlay is a window in which users can view and interact with secure content. The overlay is superimposed directly over the portions of the webmail provider's

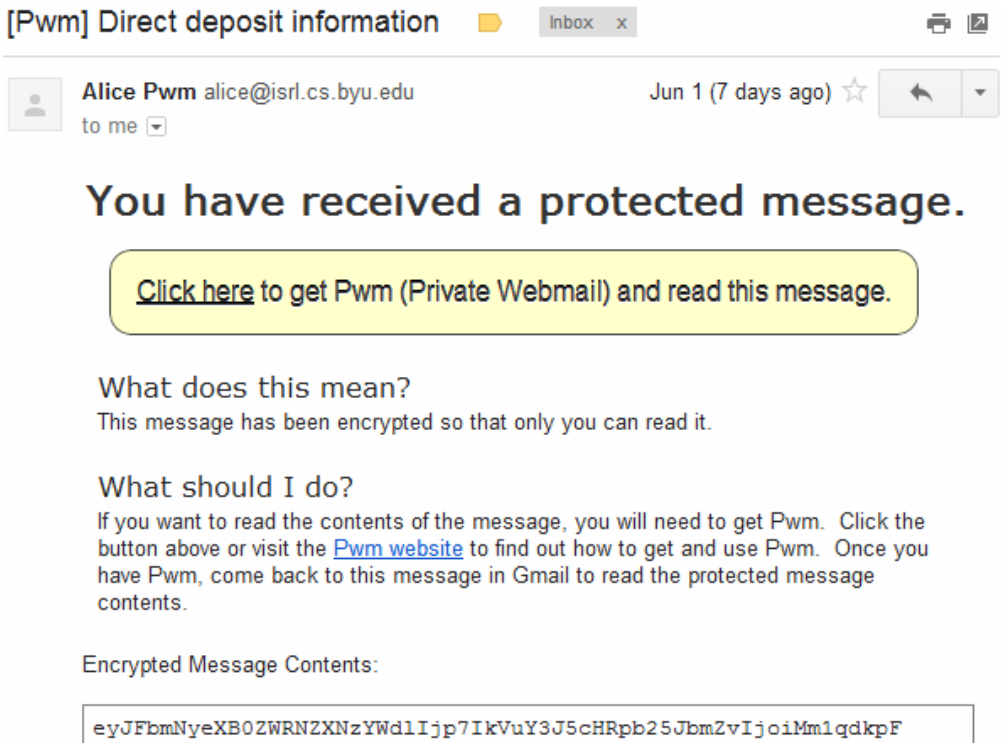

Figure 3.1: An encrypted message without a Pwm overlay

user interface that need to be secured. Users then interact with the security interface in lieu of the occluded portion of the webmail provider's interface.

Security overlays are designed to be functionally transparent to users, matching the functionality that exists in the overlaid portion of the webmail provider's interface as closely as possible. This functional transparency allows users to complete tasks in the way to which they are accustomed, lowering the chance that users will forgo using the secure system in preference of their existing, less secure webmail interface. While the overlays are functionally transparent, Pwm security overlays are designed to be visually distinctive from the webmail provider's interface. This distinction assists users in determining whether they are using a security overlay or the webmail provider's original interface and highlights features unique to the security overlay.

For example, Figure 3.1 is an encrypted Pwm email as it appears in the standard webmail view without Pwm and Figure 3.2 is that same message after it has been decrypted. The security overlay has been positioned in the page where users expect to read email.

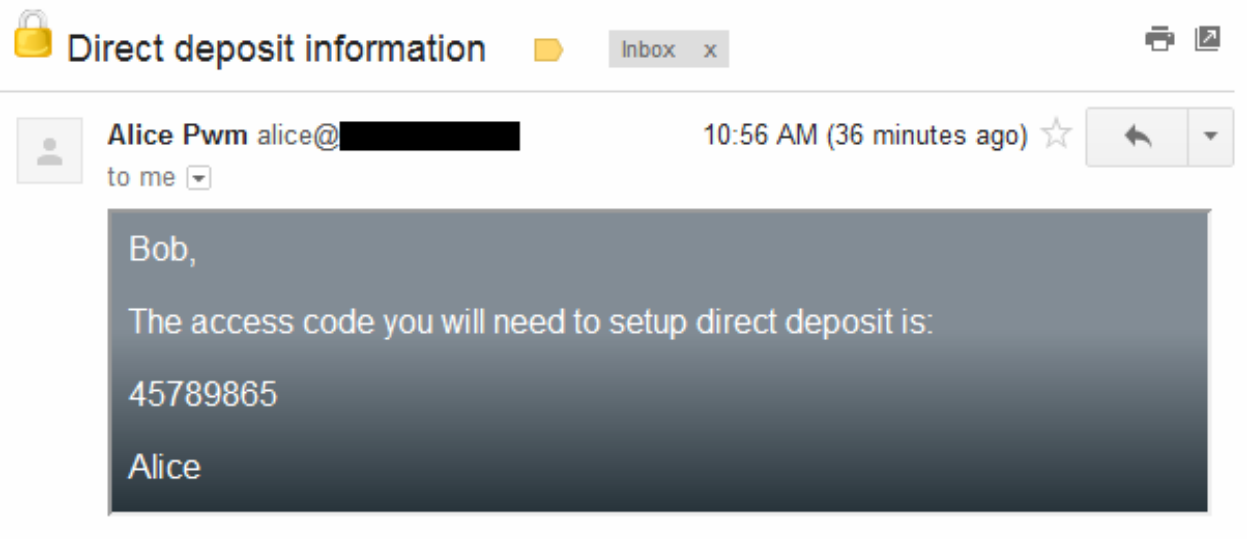

Figure 3.2: The decrypted contents of a Pwm message displayed in a security overlay

Functionally, this is identical to reading any other message, but visually it is distinctive and allows users to quickly identify when they are reading encrypted emails.

Pwm avoids visual transparency as that would prevent users from determining when the system is in use and reduce trust in the system [**?**].

3.1.2 Easy Setup

Pwm can be installed and run as either a browser extension or a bookmarklet. A browser extension is a well-known method for adding functionality to the browser. Bookmarklets are a newer and increasingly popular method for accomplishing similar goals within the context of only the currently active web page, such as the webmail provider. A bookmarklet is simply a browser bookmark that contains JavaScript instead of a URL. The bookmarklet and the browser extension both function by inserting the Pwm in-page services script onto the webmail provider's web page. The key functional difference between the two is that the browser extension is always running while the bookmarklet must be clicked each time the user visits Gmail.

Bookmarklets have several advantages in comparison to browser extensions, the most important being ease of setup. On the prototype Pwm website, the bookmarklet is represented

by a large button with the text "Secure My Email." Installation is as simple as dragging this bookmarklet button to the bookmarks bar and dropping it, creating a button with the same label. Bookmarklets are also quick and easy to use: whenever Bob wants to run Pwm, he only needs to click the "Secure My Email" bookmarklet button in his bookmarks bar.

As demonstrated by the success of Pinterest[1], typical web browser users are able to set up and use bookmarklets with little difficulty. Since this installation process requires no modification of system files, Bob does not need administrative privileges to use it. Furthermore, the prototype can be set up and run on any computer where Bob accesses webmail.

Initiating New Users

Because Pwm makes it possible for messages to be encrypted and sent to a user who has never used Pwm before it is important that Pwm enables such recipients to quickly and easily gain an understanding of what the message is and what they need to do to be able to read it. When users attempt to view an encrypted message using their webmail client, they would typically only see a block of ciphertext. By nature, such a block of ciphertext does not provide any useful information to a new user, and it cannot be completely hidden due to security restrictions enforced by most webmail providers. To overcome this problem, Pwm prepends a plaintext heading to the message ciphertext, as shown in Figure Figure 3.1. This provides information about Pwm and directions to the Pwm website containing the bookmarklet to be installed.

3.1.3 Transparent Key Management

Pwm veils most security details and tasks from users with the intent of reducing the learning curve for the first time user. This includes all key related functionality as well as authentication.

[1]Pinterest is a popular social website that makes heavy use of bookmarklets. `http://pinterest.com/`

Key Derivation

A key escrow server handles key management. The key escrow server follows the principles of identity-based cryptography (IBC) introduced by Shamir [**?**] in that cryptographic keys are generated based on users' identities (i.e., email address). This model allows users to send encrypted email to recipients who are currently outside the system since the only information required for encryption is the recipient's identity. Unlike IBC, the Pwm key escrow system uses symmetric key cryptography and key derivation [**?** **?**] instead of public key cryptography. The advantages of key escrow are (1) key management can be fully automated, (2) users can never lose their encryption keys, and (3) keys can be automatically ported to new devices. The disadvantage of key escrow is that the key escrow server has access to users' keys, which is a recognized trade-off to get the other usability benefits [**?**].

The Pwm client interacts with the key escrow server using an additional invisible key management security overlay. This security overlay handles all key management operations (e.g., obtaining and storing keys, authentication).

Authentication

Authentication is performed using Simple Authentication for the Web (SAW [**?**]), a form of email-based identification and authentication (EBIA [**?**]). When Pwm requires authentication to the key escrow server, it contacts the SAW server via a TLS encrypted HTTP request. The SAW server then generates an authentication token. This authentication token is then split in half. One half is returned to the requester in the TLS encrypted response to the HTTP request, and the other half is sent by SMTP to the email account for which authentication has been requested. Pwm retrieves this email and recombines the authentication token without requiring user input. The combination of key escrow and SAW allows for transparent and automatic key management, removing many of the difficulties users faced with traditional secure email solutions.

3.2 Implementation

While the server-side components of Pwm (e.g., key escrow server and authentication server) are web services run on BYU servers, the Pwm client, which implements the majority of Pwm's functionality, is a JavaScript program, which runs in the end user's browser environment. This Javascript client, whether initialized by the Pwm bookmarklet or the Pwm browser extension, is responsible for the creation and placement of secure overlays as well as all of the cryptographic and business logic required to implement Pwm. Due to the compartmentalization of the security overlays, it is simplest to describe the Pwm client as being composed of two top level components: the security overlays themselves and the rest of the client. Because the modules in this latter component run in the context of the webmail provider's page, they are collectively reffered to as the In-Page Services.

3.2.1 Security Overlays

Security overlays are implemented using iFrames, a well known mechanism for providing isolation in the browser [**?** **?**]. Because the content of security overlays is not hosted by the webmail provider the browser's same origin policies prevent the webmail provider from directly accessing their contents [**?**]. While direct access of a security overlay's contents is not desirable, it is still necessary for the security overlays to receive encrypted content from the webmail provider for decryption and to send encrypted content to the webmail provider for transmission. This is done using the JavaScript Web Messaging API [**?**]. To ensure that no sensitive information is leaked to the webmail provider, security overlays encrypt all such information before transmission.

Security overlays also need to communicate with each other to coordinate their efforts. To enable communication, the security overlays use JavaScript's *sessionStorage* object as a communication pipe. This object is shared between all frames from the same domain, but it is inaccessible from any other domain (i.e., webmail provider) [**?**]. The *sessionStorage* object

also has the advantage that all contents assoiciated with a given domain are automatically deleted once no more pages from that domain are loaded.

Security overlays use the JavaScript *localStorage* object to persist information over longer periods of time. This object has the same domain-based protections as *sessionStorage*, but it does not clear data between sessions.

Key Management Overlay

In addition to the standard security overlays, Pwm also uses an invisible key management security overlay that is responsible for obtaining and storing all cryptographic keys. When another security overlay needs a cryptographic key, it uses the *sessionStorage* communication pipe to request it from this invisible security overlay. The key management security overlay will then obtain the key and return it to the requesting security overlay. The key management security overlay also handles any authentication necessary for obtaining cryptographic keys. The obtaining, storing, and authenticating operations are done within this security overlay to prevent scripts in any other domain context, including that of the webmail provider, from ever having access to cryptographic keys or authentication tokens.

3.2.2 In-page Services

In addition to security overlays, Pwm uses a script known as the in-page services. At runtime, this script is injected into the webmail provider's web page. It then identifies which portions of the provider's interface need to be enhanced and creates appropriate security overlays. It also acts as a bridge between security overlays and between the security overlays and the webmail provider's web page. The in-page services also scan the user's inbox for incoming halves of SAW authentication tokens and passes them to the key management overlay to be recombined and used. The in-page services are written in JavaScript and run on all major browsers.

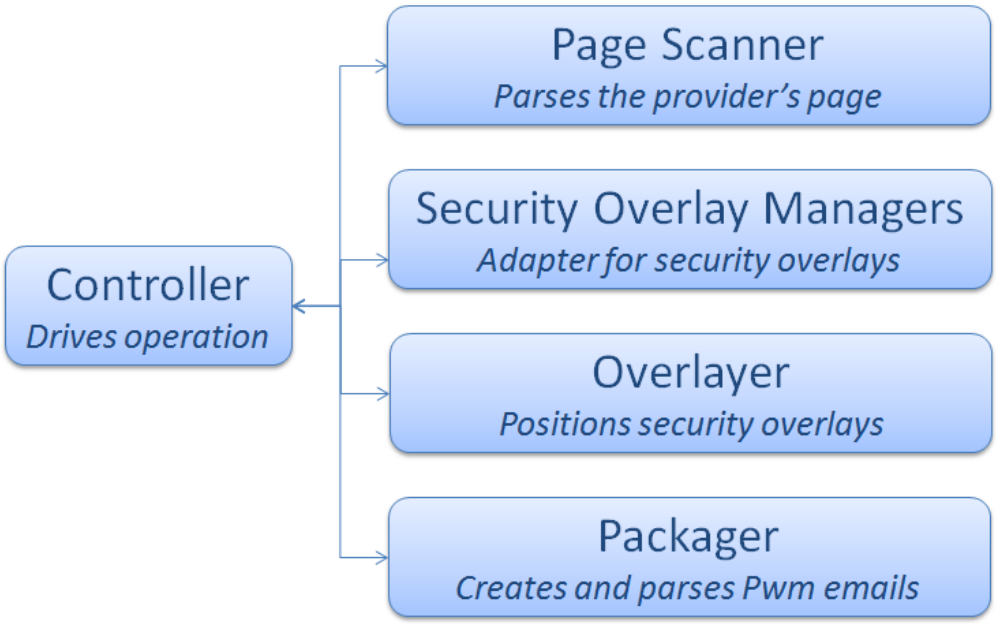

Figure 3.3: Pwm In-Page Services Modules

The functionality of the in-page services is split into modules (Figure 3.3), each of which is self-contained and handles one distinct aspect of Pwm's underlying framework. This allows components to be updated, extend or replaced as needed to support future research.

Page Scanner

The page scanner contains all of the code that is specific to the webmail provider. By moving all the provider-specific code into one place and wrapping it in a more generic interface, the remaining modules can be written in a fashion that is agnostic to a specific webmail provider. The page scanner's primary purpose is to find DOM elements on the webmail provider's web page and wrap those elements in easy-to-use JavaScript objects. The other modules use these objects to access the webmail provider's web page without needing to understand the page's DOM.

While each webmail provider requires its own page scanner, once this module is created for a specific provider, no other modules need to be modified to support that webmail provider

in Pwm. Pwm includes a page scanner for Gmail that wraps inbox items, read areas, and compose forms. The page scanner also scans the user's account for new email.

Security Overlay Managers

Security overlay managers act as an adapter for security overlays. A security overlay manager contains a reference to the security overlay's iFrame, as well as the JavaScript object from the page scanner that wraps the overlaid element. The manager provides a simple API for sending and receiving messages to and from the security overlay. When a new type of security overlay is created, an associated security overlay manager class should also be built.

Overlayer

The overlayer ensures that security overlays are correctly positioned and sized at all times. This renders the original portion of the webmail provider's interface inaccessible to users, preventing them from inadvertently interacting with it and sending sensitive information without encryption.

Packager

Pwm allows users to receive encrypted content within webmail. If users are presented with ciphertext without any explanation, they will be confused and likely treat it as spam. To assist users in recognizing and properly handling encrypted content, the framework includes a packager. The packager wraps ciphertext in a visually appealing manner, and it includes additional plaintext information intended to educate and guide users in decrypting the ciphertext. The packager is also responsible for unwrapping ciphertext it has wrapped.

Controller

The controller contains the main execution loop for the in-page services. It is responsible for driving and coordinating the functionality of all the other modules. It uses the page

scanner to find areas of the page it would like to enhance. It then creates security overlays with the associated security overlay managers for those portions of the page. The security overlay managers are then passed to the overlayer for positioning. The controller listens to the security overlay managers for messages from the security overlays, and it ensures that appropriate actions occur. Finally, the controller removes security overlays that are no longer needed.

3.3 Threat Analysis

In accordance with Pwm's design goal of providing "good enough" security, Pwm is able to protect users from a variety of different types of threats and attacks. This section provides an overview of various types of attacks against which Pwm protects messages as well as a discussion of types of attacks to which Pwm is potentially vulnerable.

At the highest level, Pwm faces three categories of potential adversaries: honest but curious webmail providers, network eavesdroppers, and active attackers. Pwm provides full protection against the first two categories and partial protection against the third.

3.3.1 Honest But Curious Providers

An honest but curious provider as a service provider who provides best-effort service in fulfilling its message or data delivery duties but who may parse, log, store or otherwise interact with user's data either as part of fulfilling its duties or for other benign purposes. In the context of typical webmail use cases, both the user's ISP and the webmail provider can be viewed as honest but curious adversaries. Such honest but curious providers can be categorized as passive adversaries who are disincentivised from carrying out active attacks since doing so would have negative effects on their reputations and legal standings.

For example, the webmail provider may follow advertising practices that involve scanning messages for key terms to be used in ad targeting. Webmail providers may also store messages in ways that users do not expect. To a lay user, it may be reasonable to

assume that the contents of a message are destroyed once a message is deleted. However, this may not always be the case. Many factors such as distributed backups or data retention requirements in some jurisdictions could result in users messages being stored by the webmail provider for periods far longer than the user might expect.

In cases where Pwm is used, either the Pwm authentication server or the Pwm key escrow server could also be viewed as s potential honest but curious provider.

Pwm protects users from such honest but curious providers by performing all encryption locally on the user's client. By encrypting all data locally, no plaintext is transmitted and is therefore never visible to an ISP or webmail provider since the decryption key and authentication token would be unavailable to such parties. When a recipient of an encrypted message decrypts the message to read it, the message is also decrypted locally, and the decrypted plaintext is displayed in a browser IFrame that is isolated from the webmail provider's message display page, and any other web pages, by the browser's same origin policies. Because message plaintexts never leave the user's client, any honest but curious provider would be completely unable to access the plaintext of any Pwm protected messages.

This same protection also holds against Pwm's own servers. While the authentication server is able to generate authentication tokens that could be used to request keys for any user, this does not make the authentication server a threat to message security because no message data (either plaintext or ciphertext) is visible to the authentication server at any time. The only possibility of information disclosure to the Pwm authentication server is that it is aware of all email addresses for which authentication is requested. This information however, is not logged or stored by the authentication server.

Likewise, while the Pwm key escrow server is capable of deriving keys for any user, no information about any messages (except for sender and recipient addresses) is exposed to the key server. This leaves an honest but curious key escrow server unable to observe any message contents. While the key escrow server does require sender and recipient email

addresses to create the appropriate keys, these addresses are not logged or stored by the key escrow server.

3.3.2 Network Eavesdroppers

We define a network eavesdropper as any party capable of passively viewing data as it is transferred over a network connection in any stage of the message delivery process. Network eavesdroppers are not capable of modifying data or viewing data encrypted and transmitted using TLS or SSL. Attackers with such capabilities are addressed later as active attackers.

Pwm provides full protection against network eavesdroppers. While the message plaintext is never transmitted over the network, an eavesdropper may be able to read the ciphertext. The ciphertext, however, is useless to the eavesdropper. The eavesdropper is unable to retrieve the creator or viewer key as these are only ever transferred in a TLS session. Likewise, the eavesdropper cannot access a user's authentication token since it is never transferred in full over an unsecured HTTP session.

3.3.3 Active Attacks

We define an active attacker as an adversary that is capable of compromising the security of the user's webmail account, the webmail provider, ISP, or Pwm authentication or key servers. An active attacker may also be capable of performing man-in-the middle attacks against TLS sessions. There are numerous possible avenues an active attacker could take in attacking a Pwm protected message.

Active attacks against the user's client computer outside of the web browser environment (e.g., key loggers, screen capture software or other such malware) are outside the environment that Pwm is capable of protecting and are therefore outside the scope of threats Pwm addresses. Likewise, Pwm does not protect against similar attacks carried out through malicious browser extensions that may be granted access permissions allowing them to access resources Pwm is unable to protect.

One notable type of active attacker Pwm aims to defend against is the malicious insider. A malicious insider is a party who has been entrusted with access to resources supporting the operation of any of the services involved with message delivery, but who abuses this access. Pwm's primary defenses against malicious insiders are based on the segregation of data between mail delivery, encryption and authentication provided by the SAW authentication scheme. Because of this separation, malicious insiders are defined as having access to only one of the necessary services: key escrow server, authentication server, webmail provider or ISP.

Key Escrow Server A malicious insider with access to the key escrow server would have the capability to generate creator and viewer keys for any arbitrary sender/recipient pair. This would enable the malicious insider to encrypt a message as if it had been created by another person, but lack of access to the victim sender's webmail account would prevent the attacker from thoroughly imitating the identity of another sender. While the malicious insider at the key escrow server would also have the keys necessary to decrypt any message, there would be no opportunity to do so because message ciphertext is never exposed to the key escrow server.

Authentication Server A malicious insider with access to the authentication server would be capable of generating authentic authentication tokens and requesting any desired keys from the key escrow server. As with the attacker at the key escrow server, the malicious insider at the authentication server would then be able to encrypt messages with any creator key but would be unable to thoroughly imitate the identity of the sender. Likewise, the malicious insider would potentially be able to decrypt any message ciphertext she obtained but has no opportunity to obtain any ciphertext.

Webmail Provider If we assume that a malicious insider at the webmail provider is able to access the contents of a victim's mailbox, the malicious insider could authenticate as the

victim, obtain keys, and potentially decipher message plaintexts or fully impersonate the victim when encrypting and sending messages. Such an attack could be carried out through the following steps:

- Initiate a request to the authentication server for authorization as the victim
- Record the portion of the authentication token returned in response to the request
- Access the victim's mailbox and retrieve the second portion of the authentication token
- Recombine both portions to derive the authentication token using the same algorithm used by the Pwm client
- Use the authentication token to request any creator or viewer keys pertaining to the victim from the key escrow server
- Use the acquired keys to decrypt cipher text in the victim's mailbox or encrypt new messages and fully impersonate the victim by sending the message from their own mailbox

If the malicious insider is in a position at the webmail provider in which he can log in as the intended victim, as opposed to accessing the contents of the victim's mailbox through some other means, all these steps could be carried out by using Pwm once logged in as the victim. This attack puts a malicious insider at the webmail provider in a position to potentially attack Pwm users. Fortunately, this attack can be thwarted by augmenting SAW with the addition of another authentication factor such as a password. This protective step is not implemented in the version of Pwm discussed in this thesis because it would not be in keeping with the goal of testing email encryption with minimal complexity for the end user.

Internet Service Provider A malicious insider with access to the internet service provider (or intermediate network nodes such as routers or gateways) is also potentially in a position for carrying out attacks against Pwm. The capacity for such an attacker to successfully attack Pwm users depends on the attacker's ability to carry out man-in-the-middle attacks

against Pwm users' TLS or SSL sessions. All possible attacks by a malicious insider at the ISP are time-dependent; that is, the attacker must be monitoring a users session at the time Pwm is being used.

If a prospective attacker is not able to carry out man-in-the-middle attacks against TLS and SSL sessions, the attacker will be less able to obtain sufficient information to carry out an attack. Such an attacker could successfully defeat Pwm security if the user authenticated to their webmail provider over a connection unsecured with TLS or SSL. In that case, the attacker could sniff packets sent from their target's computer and record their authentication credentials. The likelihood of this attack is reduced by the current practice by most webmail providers of automatically enabling TLS sessions for all users.

If, however, an attacker at the ISP is able to perform a man-in-the-middle attack against a Pwm user, there are multiple potential attack vectors. First, the attacker could capture the target user's webmail authentication credentials as described for unsecured connections above despite employment of TLS or SSL on communications between the target user and the webmail provider. Second, a man-in-the-middle attacker could intercept TLS protected connections between the target user and the Pwm authentication server. This would allow the attacker to retrieve the first portion of an authentication token. The attacker could then wait for the user's Pwm client to access the SAW email and retrieve the second portion of the authentication token as it is downloaded by Pwm. Third, the man-in-the-middle attacker could eavesdrop on the connection between the target client and the Pwm key escrow server. This would allow the attacker to retrieve keys as they are transferred to the client as well as to determine with whom the client is communicating based on what creator/viewer keypairs are requested.

3.3.4 Active External Attackers

Any active external attacker who successfully compromises any of the above services or servers would be capable of accomplishing the attacks available to malicious insiders described above.

Chapter 4

Cognitive Walkthroughs

The cognitive walkthrough is a user interface analysis technique originally developed by Polson et al. [**?**] based on skill acquisition theory. The primary goal of the cognitive walkthrough is to evaluate the capability of an user interface design to guide untrained users in learning how to use the interface to perform a defined set of actions.

The design goal of making Pwm easily adoptable by untrained users makes cognitive walkthoughs a particularly appropriate tool for its evaluation. This chapter presents a cognitive walkthrough of the Pwm interface based on the criteria Polson et al. originally presented. We also took further direction from suggestions for performing cognitive walkthoughs which the same authors later presented in the form of a practitioner's guide [**?**].

We performed the initial cognitive walkthrough on the initial bookmarklet implementation of Pwm. We also repeated the cognitive walkthrough with the browser extension version of Pwm which we built to improve the system based on the initial cognitive walkthrough and first user study. To accurately represent my work and document the improvements we made to the initial implementation of Pwm we present both cognitive walkthroughs in this chapter. The definitions of the inputs to the walkthroughs are identical for both of the cognitive walkthoughs and we define them once before presenting the *Correct Action Sequences* and *Action Walkthroughs* for the respective implementations.

4.1 Inputs to the Walkthroughs

Users of the System Users of the system are webmail users who are already familiar with Gmail and are presumed to know how to read, compose, and reply to normal messages using the standard Gmail interface. As a prerequisite, users are also assumed to be comfortable with basic tasks involved in operating a web browser including following links, switching between tabs or windows, and refreshing pages. Users are also presumed to be familiar with common graphical user interface (GUI) widgets such as dialog boxes, buttons, menus, tool bars, and hypertext links as well as common actions associated with GUI widgets such as clicking, scrolling and drag and drop actions. Additionally, users are presumed to be aware of the risks of exposing sensitive information but are not presumed to have any knowledge of encryption, secure email or other computer security concepts.

Tasks to Analyze Pwm allows users to begin using the system after receiving an encrypted message for the first time. This walkthrough will analyze the task of setting up Pwm to read a first encrypted message followed by the tasks of sending an encrypted reply and composing a new encrypted message.

User Interface Definition The Pwm User Interface to be evaluated in these walkthroughs consists of several components: the standard Gmail webmail interface (Gmail), the plaintext portion of an encrypted message (Initiation Message), the Pwm installation website (Website), the Pwm Bookmarklet or browser extension, and the Gmail webmail interface which Pwm overlays (Pwm).

Additional Considerations Because users are assumed, in the context of this exercise, to already be familiar with the basic use of a web browser, actions specific to the browser or existing Gmail interface are described in less detail than elements and actions introduced by Pwm.

4.2 Bookmarklet Walkthrough

4.2.1 Actions Required for Installing Pwm and Reading the First Message

Correct Action Sequence for the Task

- (Gmail) Click on encrypted message in the inbox
- (Initiation Message) Click on the link to the Pwm website
- (Website) Install bookmarklet
  - Drag link to bookmarks bar
  - Return to Gmail tab
- (Gmail) Click the bookmarklet
- (Pwm) Read decrypted message

Walking Through the Actions

The user begins in the familiar Gmail inbox interface in which they see a new message. The message subject line begins with the string [Pwm]. Whether or not the user notices this prefix will not affect the user's behavior in this action.

The user then clicks on the message in the inbox as they are accustomed to doing in Gmail. The Pwm Initiation Message is displayed in the section of the Gmail page where the user normally sees message contents (figure 4.1.)

A large, bold heading at the top left of the content area indicates to the user that they are viewing a protected message. Immediately below this heading a box contains a link to the Pwm bookmarklet installation website with simple instructions to click the link. The box is a light yellow color to attract the eye of the reader. The user may click anywhere on the box to activate the link.

Upon arriving at the bookmarklet installation page (Figure 4.2,) the user is presented with a brief set of steps to install the bookmarklet followed by a yellow box similar to the

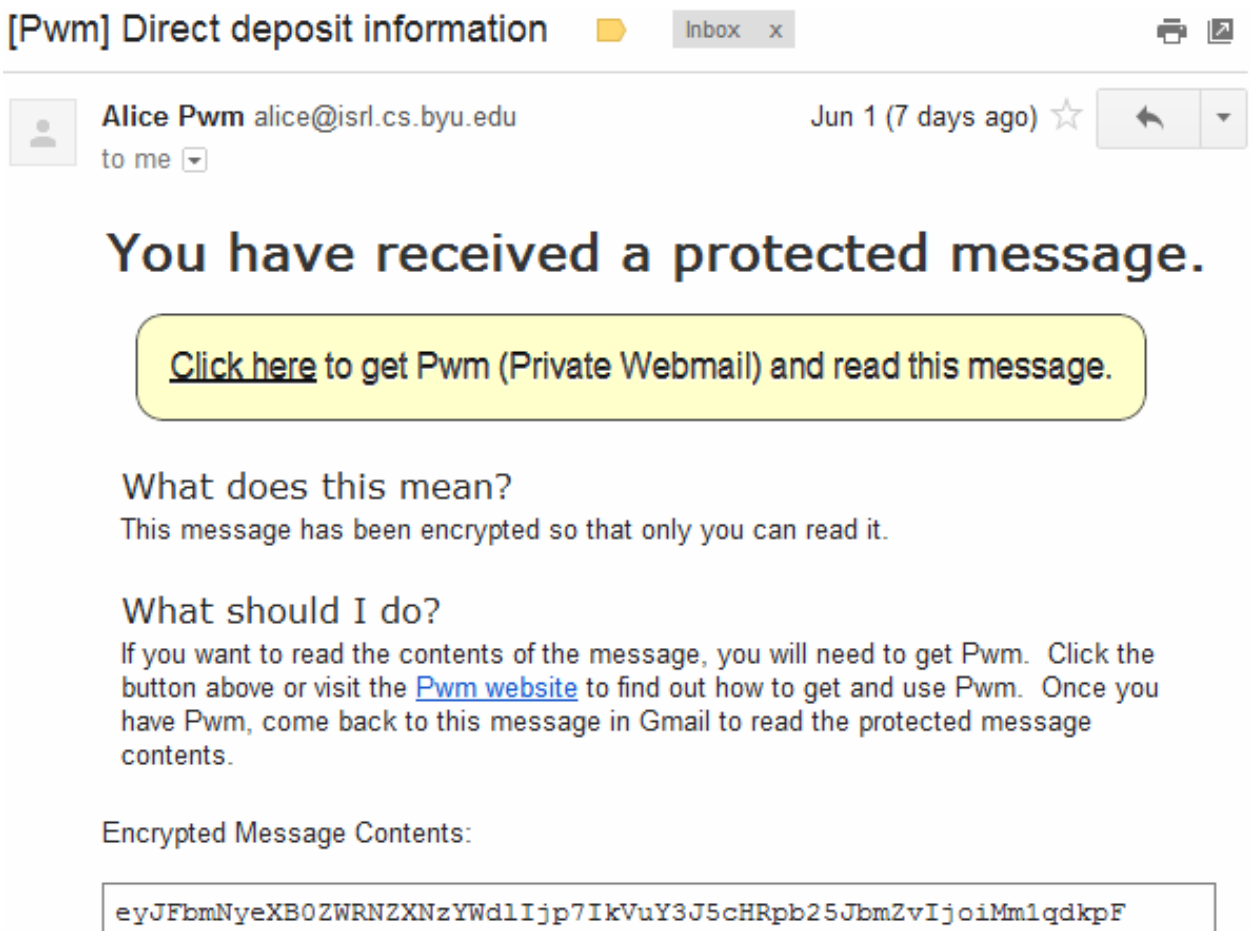

Figure 4.1: Plaintext portion of a Pwm encrypted message

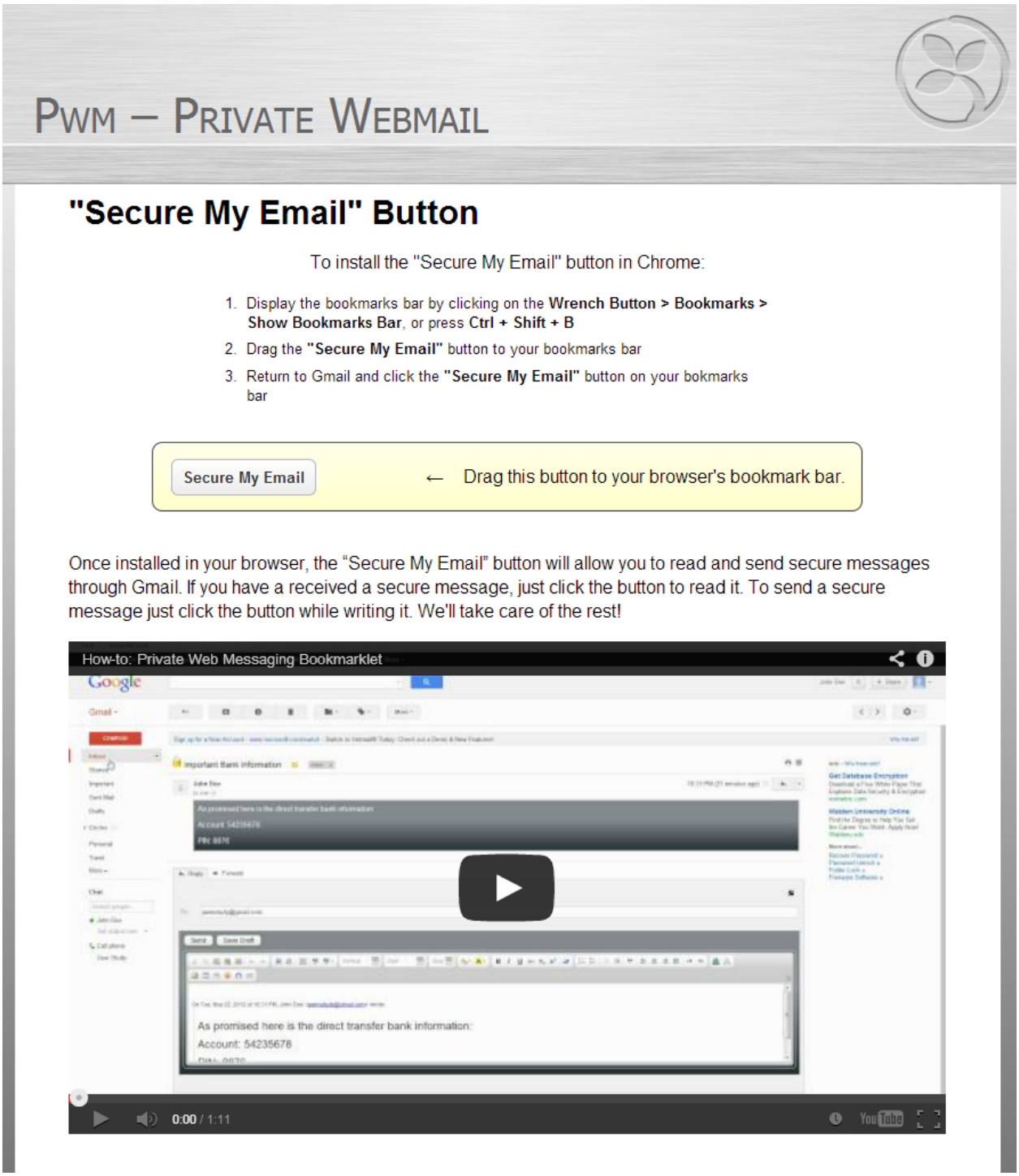

Figure 4.2: The bookmarklet installation page of the Pwm website

one in the initiation message with a button to drag to their bookmarks bar to install the bookmarklet. The box was also followed by a paragraph of text explaining what to do after installing the bookmarklet as well as an embeded YouTube video demonstrating the entire process.

We designed this page with the intent that the user's gaze would be guided from the top of the page down through the step-by-step instructions and on to the button that would then be dragged to the browser's bookmarks bar. Unfortunately, the visually prominent installation box dominated the viewer's gaze potentially causing the step-by-step directions to be skipped. The tendency for users to do this was confirmed in the usability studies. Skipping the step-by-step directions proves problematic unless a user already has the browser bookmarks bar visible. We address this problem in designing the extension installation page described in section 4.3.

The YouTube demonstration video features prominently on the page and naturally draws in the user's gaze. This was intended to help users who may have had difficulty with the text directions and desired visual guidance. Despite the visual prominence of the video the usability study showed that only one participant played the video and did not watch it to completion.

The final step in the step-by-step directions and the last paragraph on the page instruct the user to return to Gmail and click the bookmarklet. Without this instruction, users would not know to click the bookmarklet once they were back in Gmail. Although the bookmarklet was in the browser chrome and outside the context of the Gmail page, we expected that the short temporal gap between installation and use would overcome this context barrier.

Once the user has returned to Gmail and clicked the bookmarklet, the message is automatically decrypted and displayed on a gray background indicating to the user that it is separate from the base Gmail interface.

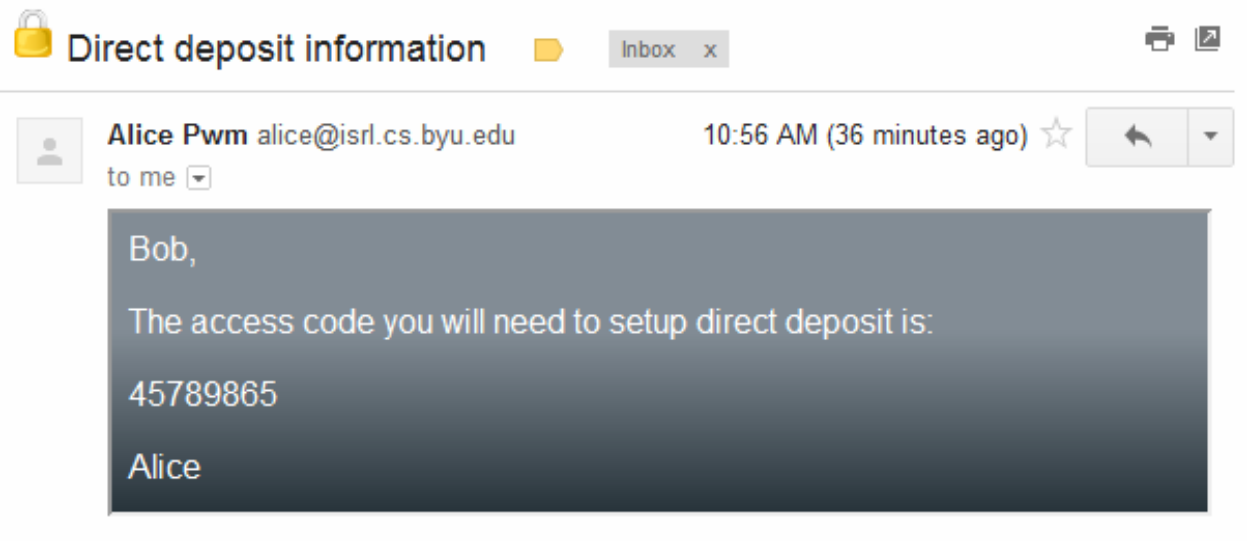

Figure 4.3: The plaintext contens of a Pwm protected message displayed after successful decryption

4.2.2 Actions Required for Reading Subsequent Messages

Correct Action Sequence for the Task

- (Gmail) Open message from the inbox
- (Gmail) Click the bookmarklet to launch Pwm (if not already running)
- (Pwm) Read decrypted message

Walking Through the Actions

Users are assumed to open emails as they normally would. If the user has already clicked the Pwm bookmarklet any secure messages will be marked with a lock icon in the inbox view and will be automatically decrypted and displayed when opened. If the user has not already run the bookmarklet in the current session they will see the initiation message again. Seeing the installation message should remind the user of what they did to install the bookmarklet resulting in the user clicking the bookmarklet. One case that arose in the user study was that if the user does not remember the bookmarklet they will be directed back to the installation page where they will either remember the bookmarklet or may install it again. Once the

user has remembered or reinstalled the bookmarklet, clicking it will decrypt the message and display the plaintext.

4.2.3 Actions Required for Replying to a Pwm Message

Correct Action Sequence for the Task

- (Gmail) Open message from the inbox
- (Gmail) Click the bookmarklet to launch Pwm (if not already running)
- (Gmail) Click Gmail reply button or click on the reply box below the message body
- (Pwm) Enter reply contents into Pwm overlay
- (Pwm) Click the *Send* button

Walking Through the Actions

Once a user has used the bookmarklet to read a Pwm message, any attempt to reply will result in Pwm automatically overlaying the Gmail reply editor with Pwm's secure editor. No action is required on the part of the user to make the reply secure. If the user hasn't launched the bookmarklet they may still reply, but the reply will not be secure and the quoted message in the reply will be the initiation message and ciphertext.

4.2.4 Actions Required for Composing New Secure Messages

Correct Action Sequence for the Task

- (Gmail) Click on the *Compose* button
- (Gmail) Click the bookmarklet to launch Pwm (if not already running)
- (Gmail) Enter sender address and subject
- (Pwm) Enter secure message contents into the Pwm overlay
- (Pwm) Click the *Send* button

Walking Through the Actions

The user may click the bookmarklet at any time before or during the composition process to make the message secure. Users begin composing a message normally by clicking the standard Gmail *Compose* button to which they are already accustomed. If a message body has already been entered, the existing text will automatically be encrypted. When Pwm is activated, the Pwm overlay is placed over the message body area, but not over the recipient or subject fields. This provides a visual distinction to the user that the recipients and subjects are not part of the message that is encrypted. A lock icon on a glass-style button is placed in the corner of the overlay.

The greatest risk of user error in this task is in simply forgetting to click the bookmarklet. Unfortunately the placement of the bookmarklet in the browser chrome means that users have to remember to look outside the context in which they are used to composing messages. This did prove to be problematic for a few users in the usability study and we made changes to the design of the extension in order to reduce this type of user error.

4.3 Browser Extension Walkthrough

Based on the results of the initial cognitive walkthrough and observations of users attempting to use the bookmarklet, we implemented a number of improvements. Most significantly, we implemented Pwm as an extension for the Google Chrome web browser. Doing this eliminated the step of dragging an item to the bookmarks bar which had proved to be the most difficult step for many users. Running as an extension also allows Pwm to launch automatically and eliminates the need for the user to look outside the context of the web page to activate Pwm.

4.3.1 Actions required for installing Pwm and Reading the First Message

Correct Action Sequence for the Task

- (Gmail) Click on encrypted message in the inbox

- (Initiation Message) Click on the link to the Pwm website
- (Website)Install Extension
  - Click *"Add to Chrome"* button
  - Approve extension installation
  - Return to Gmail tab
  - Refresh Gmail tab
- (Pwm) Read decrypted message

Walking Through the Actions

Upon opening a Pwm message, the user sees an initiation message essentially identical to the one used for the bookmarklet. The primary difference is that the "Get Pwm" link takes them to a different website to install a Chrome browser extension. Based on the user tendency to skip text-based step-by-step instructions when installing the bookmarklet, we designed the extension installation page to more explicitly guide the user through each step.

Upon arriving at the installation page (figure 4.4) the user's gaze is drawn to the most prominent element of the page, a gray box containing instructions presented as an ordered list with the first step displayed in bold to draw the user's attention. This step directs them to click the prominent blue button to the right labeled "*Add to Chrome*". This button follows the labeling and styling convention provided by Google to make it consistent with other extension installation pages and therefore familiar to users who have used extensions in the past.

Once the user clicks this button, the first instruction step is grayed-out, the second step is highlighted in bold, and a dialog box is presented (figure 4.5) requesting their approval to install the extension. The text in the second step directs the user to approve the installation.

Once the user approves the installation the extension will be installed invisibly and a browser message appears in the upper right-hand corner of the window (figure 4.6) informing

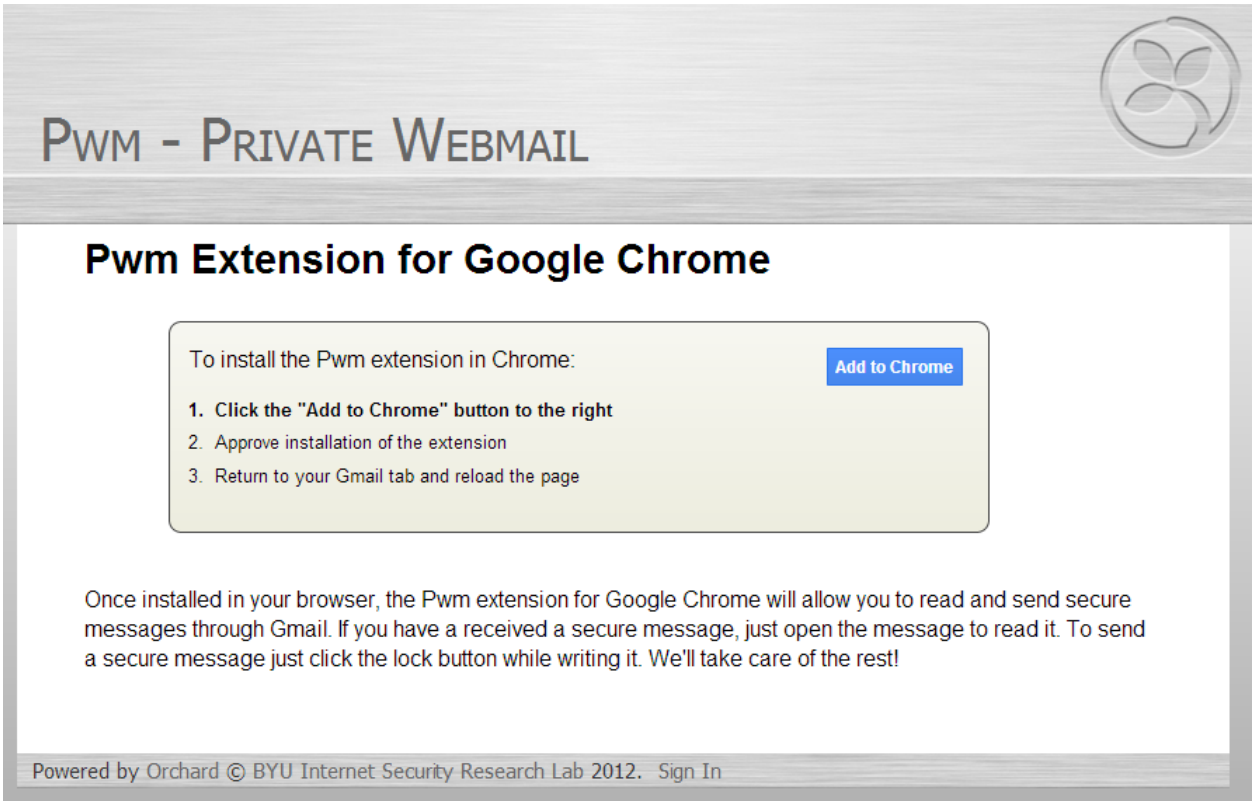

Figure 4.4: Pwm extension installation page

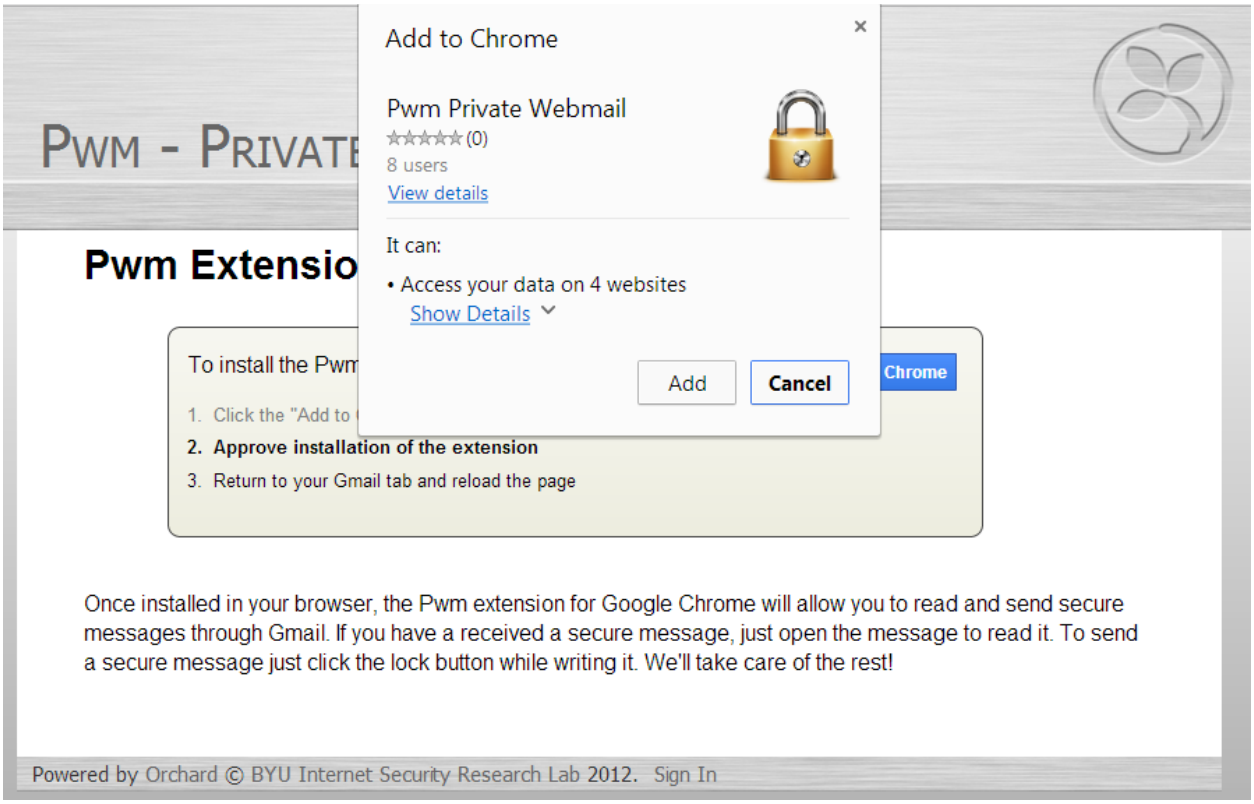

Figure 4.5: Extension installation approval dialog

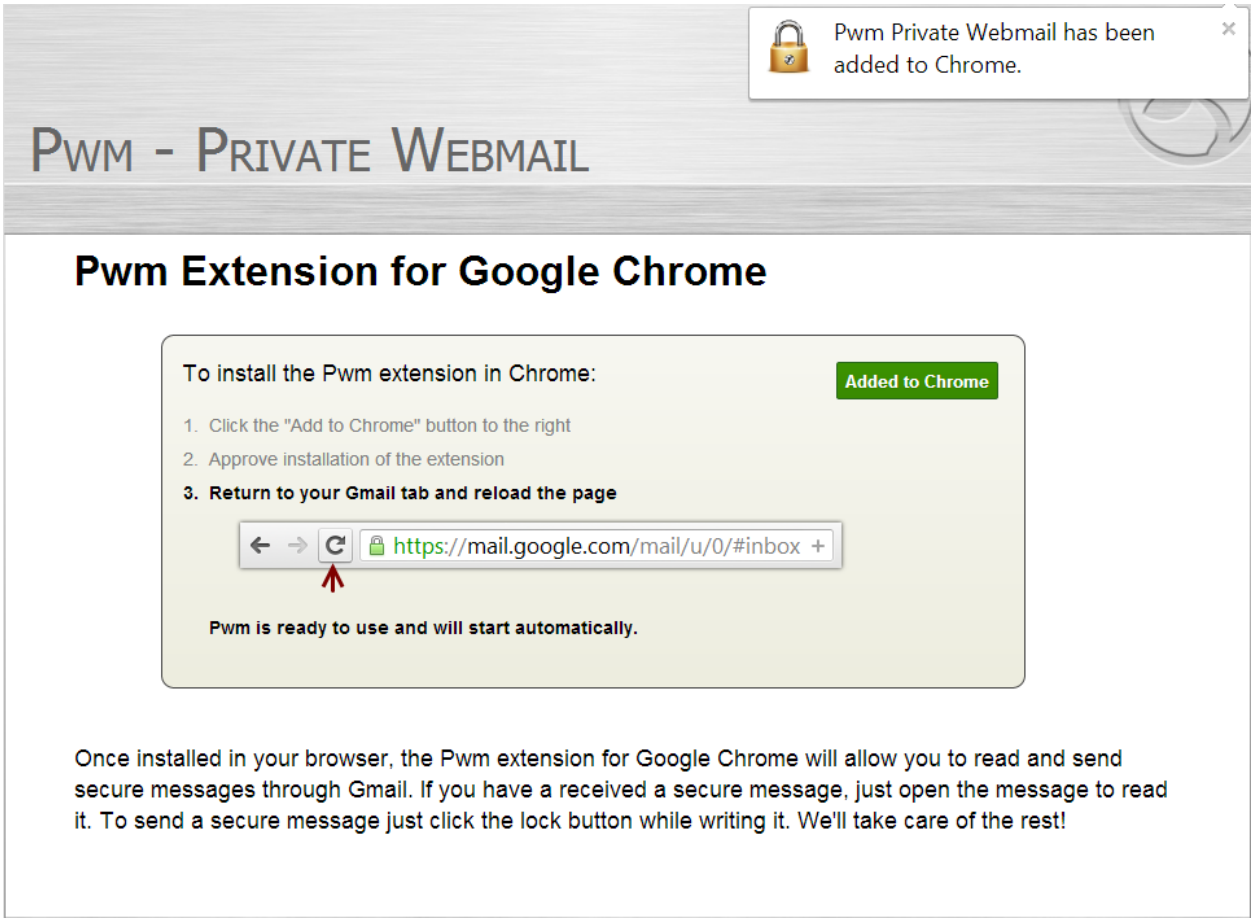

Figure 4.6: Extension installation completed with instructions to return to Gmail and refresh the page

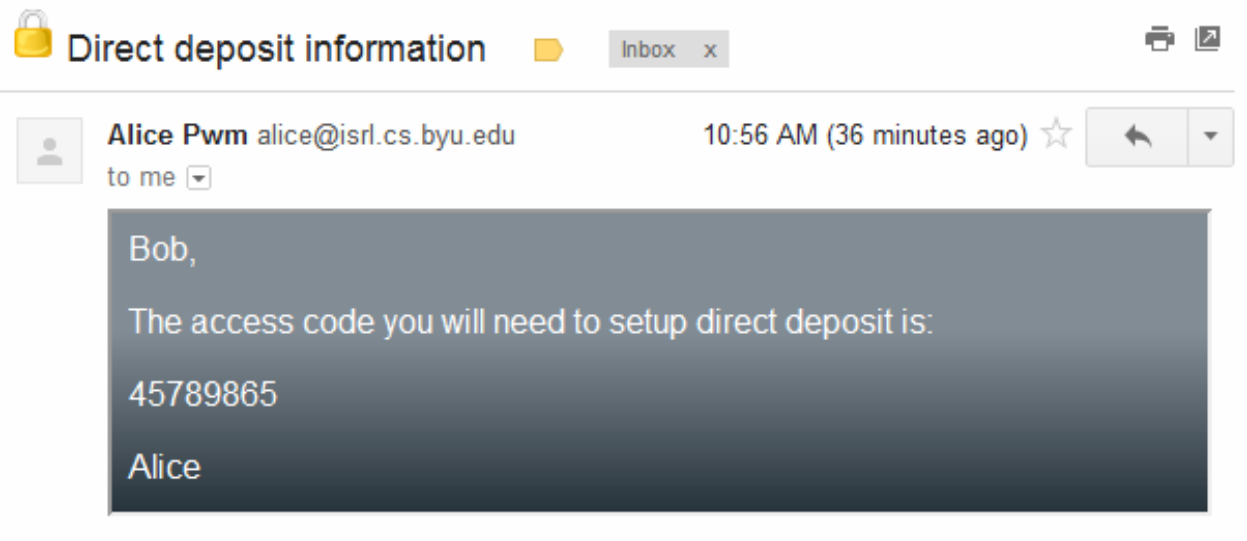

Figure 4.7: The plaintext contens of a Pwm protected message displayed after successful decryption

the user of successful installation. We simultaneously gray-out the second instruction and emphasize the third step in bold. This final step directs the user to return to Gmail and refresh the page. Refreshing is only necessary because the user's Gmail session was started before the extension was installed. In the future, the extension will automatically run whenever the user opens Gmail.

Once the user refreshes the Gmail page (which is still displaying the initiation message at this point) the extension will automatically decrypt the message and display the plaintext contents in Pwms distinct gray overlay (figure 4.7.)

4.3.2 Actions Required for Reading Subsequent Messages

Correct Action Sequence for the Task

- (Gmail) Open message from the inbox
- (Pwm) Read decrypted message

Walking Through the Actions

The Pwm extension runs when the user opens Gmail and automatically replaces the [Pwm] prefix in message subjects in the inbox view with lock icons to indicate to users which messages have been encrypted. When the user clicks on a Pwm message to open it (just like any other message) the Pwm extension automatically decrypts the message and displays the plaintext contents in the Pwm overlay. No additional action is required of the user.

4.3.3 Actions Required for Replying to a Pwm Message

Correct Action Sequence for the Task

- (Gmail) Open message from the inbox
- (Gmail) Click the Gmail reply box below the message body
- (Pwm) Enter reply contents into Pwm overlay
- (Pwm) Click the *Send* button

Walking Through the Actions

The user opens the Pwm message and reads the automatically decrypted content. The user can then click the existing Gmail reply box below the message. The Pwm extension detects that the user is replying to an encrypted message and automatically overlays the Gmail editor with a secure composition overlay. If the user has started entering text before the overlay is loaded it is automatically moved into the overlay. No additional user action is necessary.

4.3.4 Actions Required for Composing New Secure Messages

Correct Action Sequence for the Task

- (Gmail) Click on the *Compose* button
- (Gmail) Enter sender address and subject

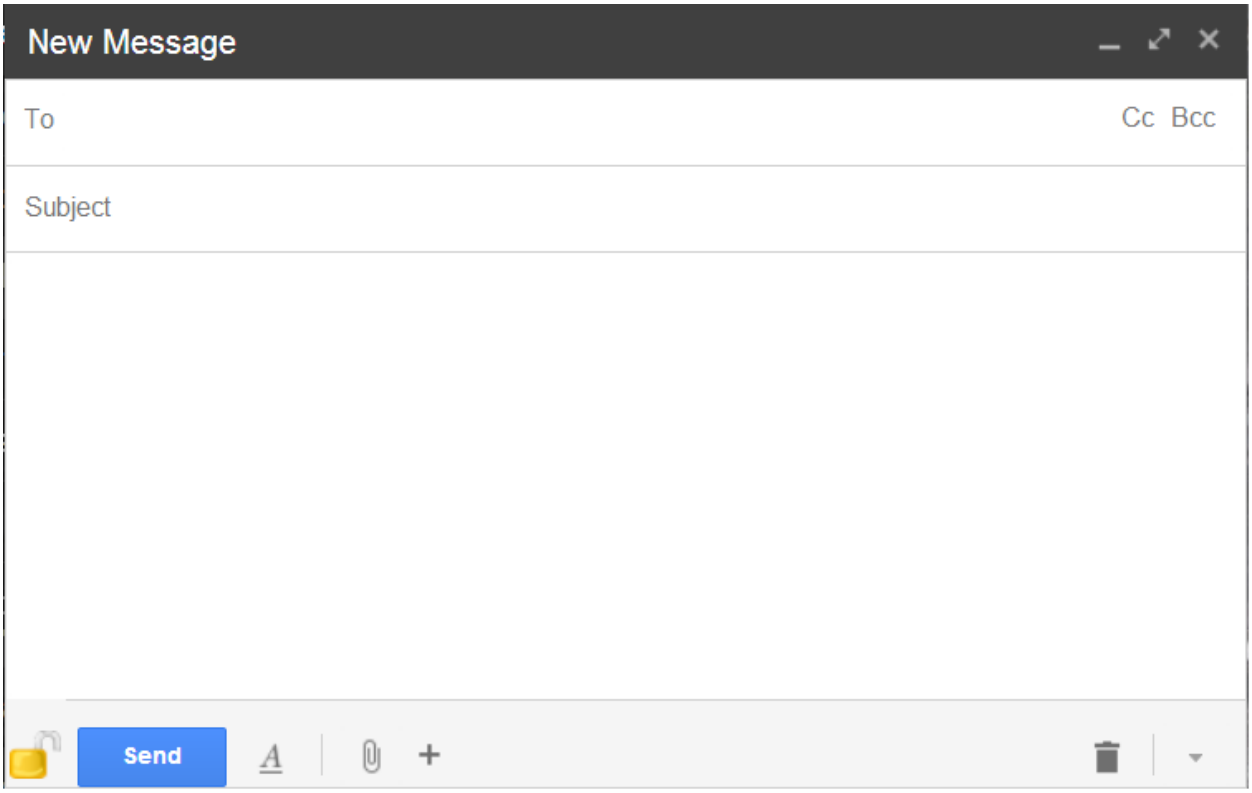

Figure 4.8: Compose view without an overlay. Note that the extension has automatically added a lock button next to the send button

- (Gmail) Click the yellow lock button
- (Pwm) Enter secure message contents into the Pwm overlay
- (Pwm) Click the *Send* button

Walking Through the Actions

Pwm automatically runs when Gmail is loaded so there is no need for the user to take any action to enable it like the bookmarklet. When the extension sees that a new compose view has been opened by the user, it places a bright yellow lock button next to the send button (figure 4.8). The position and color of this button make it very noticeable when the user looks at the section of the screen containing the send button. Clicking this button will insert the Pwm overlay and copy any text the user has already entered into the overlay (figure 4.9. The user may click this at any time before sending the message. While requiring the users

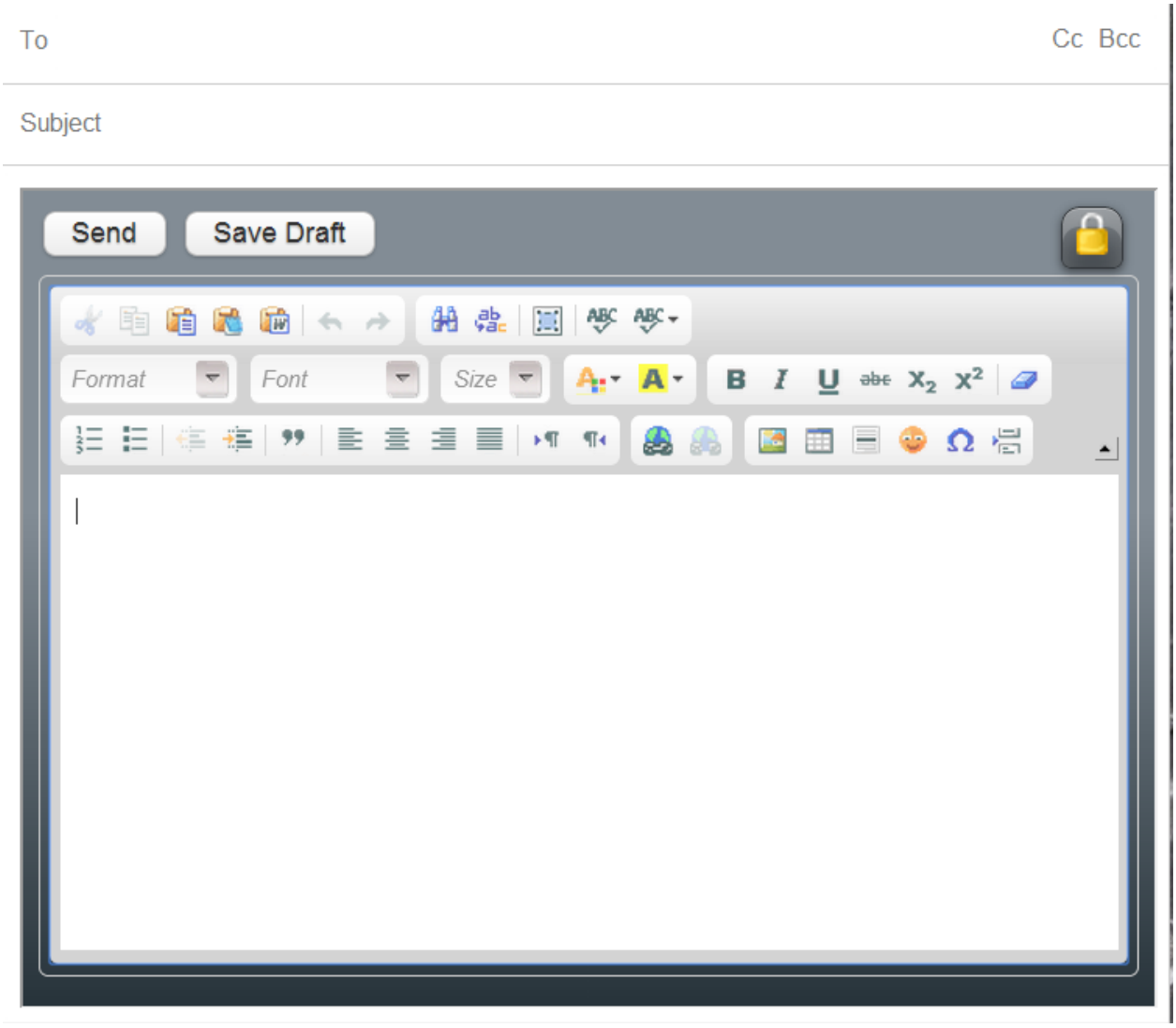

Figure 4.9: Compose view with overlay inserted by the Pwm extension

to click a button to enable encryption does create some room for user error, keeping the required action inside the existing compose form eliminates the need for the user to leave a familiar context. In the user studies, a few users did forget to click the lock button, but users almost unanimously stated that they would not want all new messages to be encrypted by default and prefer to have to click the lock button to enable encryption.

Chapter 5

Usability Studies

We conducted user studies to evaluate the usability of Pwm. These studies focused on assessing the ability of untrained users to learn and successfully use Pwm and were designed to accomplish several goals:

- Determine whether new, untrained users could set up and use Pwm relying only on the directions provided in the plaintext portion of the encrypted email and the Pwm website (Figures 5.1 and 5.2)
- Identify problems in Pwm that prevent users from correctly sending and receiving encrypted mail
- Observe any cases in which users fail to use Pwm correctly and identify how such failures can be avoided or reduced in the future.

Due to inherent limitations of studies in a lab environment, [**?**] these studies did not focus on evaluating the level of trust that users placed in Pwm, or how trust could be more effectively established. Such evaluations of trust would indeed be beneficial in verifying the usefulness of Pwm, and could potentially be accomplished via an IRB approved study. An outline for how such a study could be carried out is described in the future work section of this thesis.

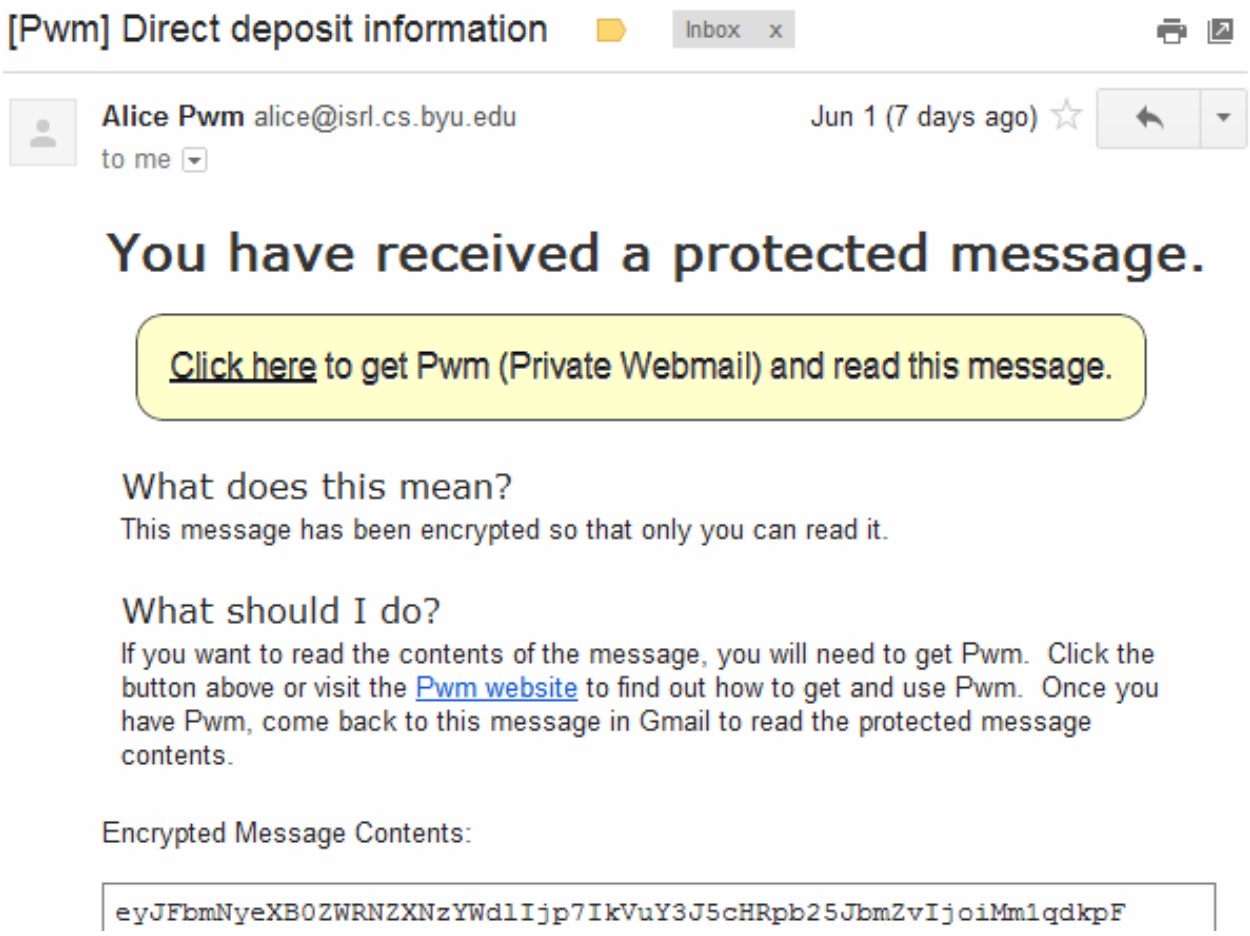

Figure 5.1: Plaintext portion of a Pwm encrypted message

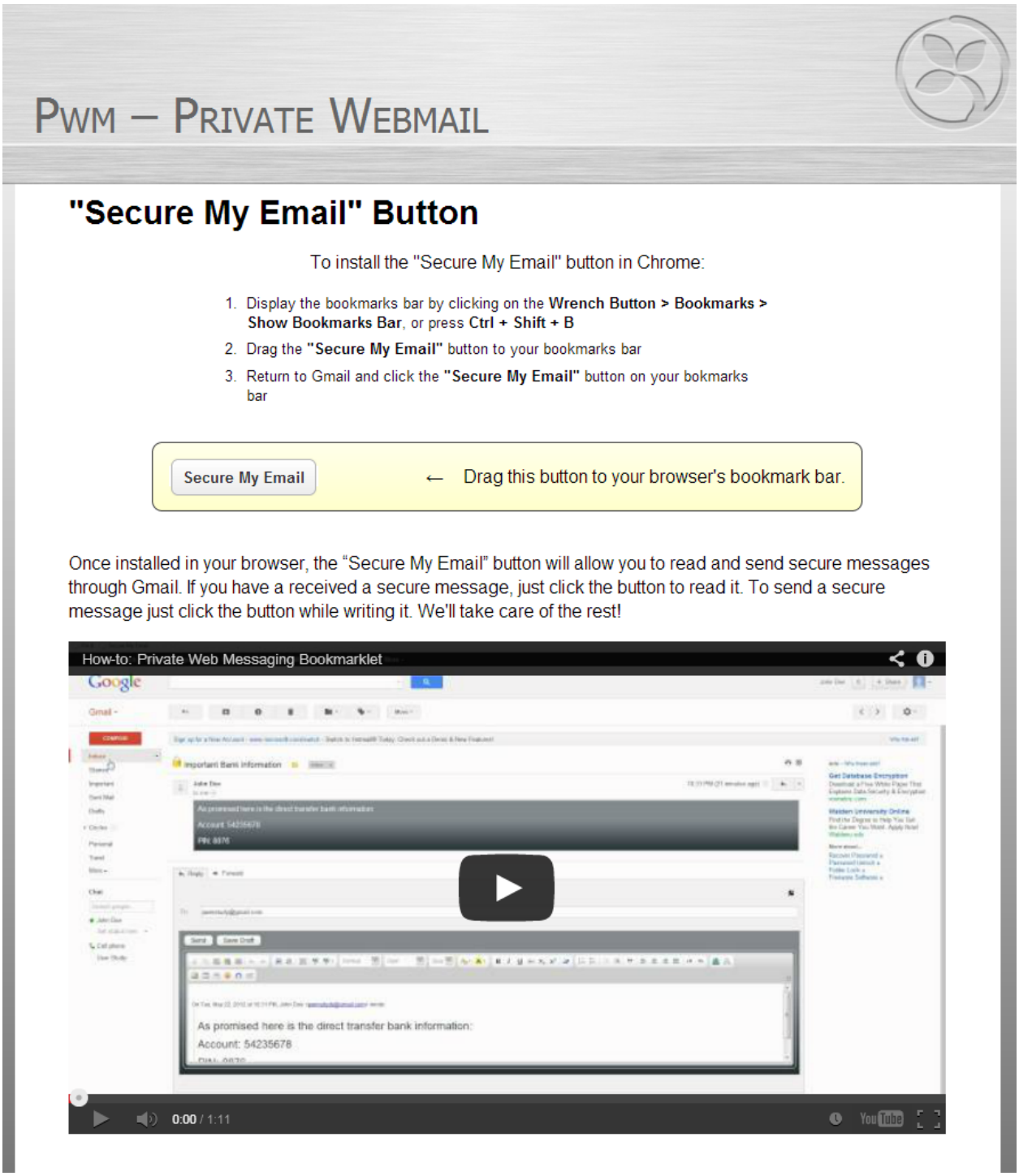

Figure 5.2: The bookmarklet installation page of the Pwm website

5.1 Study Design

The Pwm user studies were divided into two separate stages. This division was made so that changes based on the results of the first study could be implemented and evaluated in the second study.

5.1.1 System Usability Scale

To support the goal of objectively evaluating Pwm's usability, we needed a standard method of rating Pwm's usability in a useful way. To this end, we used the System Usability Scale (SUS) [**?**], a usability evaluation metric developed at Digital Equipment Corp. to rate the usability of software. SUS works by asking participants to use a system and then respond to ten statements on a Likert scale. We included these statements as part of the survey we administered to participants. The questions in the SUS alternate between positive and negative perspectives on various aspects of usability in order to avoid bias. Each statement is rated on a 5 point scale from "Strongly Disagree" to "Strongly Agree". After responses from all participants have been recorded, the mean SUS score is calculated based on all user responses. This score is a scalar value between 0 and 100, with 100 being a perfect, ideal score.

Naturally, a scalar value is not a meaningful indicator of usability. A notable amount of work has been done in defining a meaningful interpretation of SUS scores. One particularly valuable work in this area [**?**] surveyed 2,324 participant responses in 206 studies of different systems. The results of SUS evaluations in these systems were compared against other empirical measurements of the various systems' success to derive an adjective-based ratings for SUS scores. (Figure 5.3)

Applying SUS to Pwm It is worth noting that while SUS has been demonstrated to be useful in evaluating a wide variety of systems of various sizes and diverse interface types (including GUI, text-based terminal, web, cell phone and interactive voice response) [**?**]

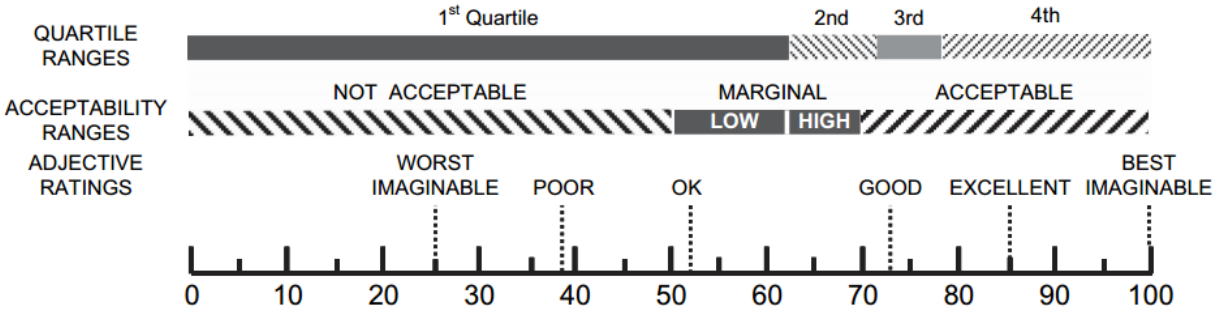

Figure 5.3: An adjective-based interpretation of SUS scores [**?**]

Pwm differs from many of those systems in that encrypted email was an entirely new concept to many of our study participants. This differs from many studies, most notably the DEC studies for which SUS was developed, that focused on evaluating software for performing duties that were part of the participants' daily jobs or were otherwise familiar. Unfamiliarity with the concept of secure mail on the part of participants in our study means that when they are asked to respond to statements such as "I think that I would like to use this system frequently" they may be likely to conflate their desire to perform email encryption with their opinion of Pwm as a means of encrypting email.

5.2 Bookmarklet Study

The first stage was conducted using the bookmarklet implementation of Pwm.

5.2.1 Setup

Our study was comprised of 25 students from Brigham Young University with low to medium technical experience. We told participants that they would be participating in a Gmail usability study, but did not alert them that it was related to security. To minimize unfamiliarity with Gmail impacting our results, we stipulated that volunteers for the study should be active Gmail users. Of the 25 participants 19 (76%) had been Gmail users for over

a year and only 3 (12%) had been Gmail users for less than 6 months. Twenty-three (92%) of the participants reported using Gmail on a daily basis.

All participants used the same computer[1] and were required to complete the study using the Chrome web browser[2]. To match a fresh install of Chrome we ensure that the bookmarks bar was not initially displayed. The participants used a Gmail account we had created to complete the study. Participants began the study by providing a small amount of demographic data. They were then presented with simple tasks to complete using Pwm. After completing the tasks, participants were presented with a short survey about their experience using Pwm. Finally, we conducted a brief interview with each participant to try to gather more detailed information about their experience.

5.2.2 Tasks

We gave each participant two tasks to complete using Pwm. In the first task, participants were told to check their inbox for an email containing instructions on how to proceed with the study. Unknown to them, this email had been encrypted using Pwm. Participants were given no additional explanation or help and were required to rely only on the directions provided by Pwm. Once decrypted, the email instructed participants to send an encrypted reply and return to the study instructions. The primary goal of this task was to observe how successfully untrained users could set up and use Pwm without outside assistance. Because we were in a lab environment where participants knew they would not be exposed to any real risks, we do not attempt to draw any conclusions about participant's trust towards Pwm or bookmarklets from this task.

For the second task, participants were asked to open a new Gmail session, send an encrypted email to the study conductor, and then wait for a reply with further instructions. The requirement to use a new Gmail session ensured that the Pwm bookmarklet was not running and would need to be started by clicking on the "Secure My Email" bookmarklet

[1] 3.0 GHz Intel Core 2 Quad CPU with 8 GB of RAM running Windows 7
[2] Version 19.0.1084.52 m

obtained in Task 1. If participants did not encrypt their email they received an unencrypted reply informing them that their email had not been encrypted and instructed them to try again. Once the participant successfully sent an encrypted email they received an encrypted reply instructing them to close Gmail and return to the study instructions. The primary goal of this task was to determine whether participants would restart Pwm to compose an encrypted email.

5.2.3 Results

Overall, participants were highly successful in using Pwm. All but one of the 25 participants (96%) successfully set up Pwm and decrypted the email received in task 1. The only participant who failed to decrypt the email had correctly set up Pwm, but then moved on to the second task without trying to read the decrypted email. When asked why she did this she said that it was because she misunderstood the task and thought she had completed it when she had added the Pwm bookmarklet.

After decrypting the email in the first task, 23 out of 24 (96%) successfully sent an encrypted reply. The only participant who failed to send the encrypted reply had correctly used Pwm but then clicked Gmail's "Compose" button rather than the "Send" button. He did not repeat this error on the second task. When asked about this, he said that he was accustomed to using Gmail on his iPod Touch where the send button is in the upper-left corner of the screen where the "Compose" button was in our test.

On the second task 22 participants (88%) successfully sent an encrypted email on their first try. Of the three who failed one immediately recognized his mistake and correctly sent an encrypted email before receiving a reply. When asked about this, he reported that he realized it had been sent without encryption when he didn't see the security overlay's black background. The remaining two participants successfully sent an encrypted email after receiving the reply asking them to try again. One of the two stated that they had misread the instructions and didn't realize they were supposed to encrypt the email. The other reported

that he didn't realize he needed to click the bookmarklet again, and said that he wouldn't make that mistake again.

While minor, the most common area of difficulty for users was in setting up the bookmarklet. Nine of the participants (36%) read the instructions and were able to set up the bookmarklet in approximately thirty seconds. The remaining 16 (64%) didn't initially read the instructions and attempted to drag the bookmarklet without first enabling Chrome's bookmarks bar. Two of these participants discovered that dragging the bookmarklet onto the tab with their Gmail session ran Pwm, but did not create a bookmark that could be used later. The remaining 14 participants eventually read the instructions and finished setup without further problem. One participant later commented that prior to reading the instructions he had assumed Google Chrome lacked a bookmarks bar and lack of that feature was the reason he had not been using it as his primary browser. Another participant said that while she initially had trouble displaying the bookmarks bar, once it was visible the bookmarklet was intuitive and familiar because she had used one before on the site Pinterest. In the end, all participants (100%) succeeded in setting up and using the bookmarklet, typically taking about one minute, with no participants taking more than two minutes.

SUS Score

Based on the participant's responses, we calculated a SUS score of 75.70 out of 100 for Pwm with a standard deviation of 13.6 and a 95% confidence interval of ±5.33. When compared against Bangor's findings on 273 SUS studies, our score of 75.70 falls in the third quartile (70.5 - 77.8) and above the mean score of 69.69. Under Bangor's adjective rating scale for SUS results, Pwm's score falls within the "acceptable" range and qualifies for an adjective rating of "Excellent" in Bangor's adjective scale for interpreting SUS scores.

5.2.4 Lessons Learned

The most important lesson learned from this study is that bookmarklets are highly usable so long as users are familiar with the process of adding bookmarks in their preferred browser. Only five (20%) of the participants reported using a bookmarklet before participating, nevertheless all participants were able to set up and use Pwm. When asked in the survey what they liked about Pwm, 23 out of 25 (92%) stated that it was simple and easy to use. No participant indicated that they felt Pwm was difficult to use. Five of the participants (20%) even asked if Pwm was currently available for download as they would like to begin using it to encrypt their email.

We also learned that while the instructions were sufficient to help participants set up Pwm, they needed to be made even more prominent on the Pwm website. No participant demonstrated preexisting knowledge of how to enable the bookmarks bar and the instructions were crucial in helping them set up Pwm. The participants who read the instructions before attempting to add the bookmarklet set up Pwm far faster (average of 30 seconds) than those that tried to add the bookmarklet without first reading the instructions (average of 1.5 minutes).' Because of these results we moved the setup instructions to appear above the bookmarklet button in order to encourage users to read the instructions before attempting to add the bookmarklet.

We also determined that participants were able to clearly tell the difference between Pwm's secure interface and the underlying interface. Some liked the distinct dark background of the security overlay, while others wished it looked more like Gmail's native interface. When asked, all participants indicated that it was easy to determine whether an email was encrypted using Pwm. This suggests that users would be unlikely to accidentally send unencrypted email. The three participants (12%) that initially failed task two indicated that in the future they would exercise extra care to ensure that Pwm was running before composing their email and indicated that they doubted they would make the same mistake again.

It was also interesting to find out that approximately a third of participants were interested in how their email was being encrypted. While these participants lacked the technical background to understand the cryptography being used, they stated that they would still like to see these details published on the Pwm website. They indicated that this would make them feel more confident using Pwm.

We also found in this study that some participants would prefer to have a browser extension instead of the bookmarklet. Although only two specifically mentioned wanting an extension, this desire was indicated by several of the participants stating that they would prefer that Pwm was always running when they used Gmail. Others indicated that while the bookmarklet worked well, they would like to see the "Secure My Email" button placed directly in Gmail's interface. When told that this would require an extension, they indicated that they would be willing to install the extension to have the greater integration. None of these participants indicated that the bookmarklet should be removed, but rather that both a bookmarklet and an extension should be available for users.

5.3 Comparison Study

The second user study was designed to serve two primary purposes:

- Test whether improvements suggested by the first study result in an improved user experience
- Compare the Pwm user experience to an existing standard solution

5.3.1 Evaluating Improvements

The first study revealed that users had two primary difficulties in using the Pwm bookmarklet: correctly using the browser's interface to add the bookmarklet and remembering to click the bookmarklet to load Pwm in a new Gmail session. To address these difficulties, we created a new version of the Pwm prototype that runs as an extension to the Chrome web browser.

The extension version of Pwm addressed these two user challenges while providing the same functionality once Pwm was loaded in a Gmail session.

Running Pwm as a browser extension rather than a bookmarklet allowed users to bypass the challenges of enabling the Chrome bookmarks bar and dragging the bookmarklet link to the corresponding target area in the browser chrome. Instead, users were able to install the extension simply by clicking on a button on the Pwm web page followed by clicking "Add" on a confirmation dialog. Alternatively, curious or cautious users could follow a link to view and install the extension in the Chrome Web Store. The default, two-click installation process was simpler than the bookmarklet installation process which required users to use hot keys to make the bookmarks bar visible followed by clicking and dragging.

The ability of browser extensions to run whenever a page on a specified domain is loaded allowed us to also solve the challenge users faced of remembering to load the Pwm bookmarklet in new Gmail sessions. Whenever a page on one of the Gmail domains was loaded, the extension checked if it was a read, compose, or inbox view and injected the appropriate Pwm components just as the bookmarklet did.

5.3.2 Comparison

In order to establish that Pwm provided a usability improvement over other existing solutions potentially available to webmail users, we used this study as an opportunity to compare our Pwm extension against an existing solution. Surveying available options for such a comparison yielded two viable choices for comparison: Hushmail and Voltage SecureMail Cloud.

Like Pwm, Voltage SecureMail Cloud was designed to allow messages to be encrypted and sent to recipients who had not taken any preparatory action. While Hushmail does support sending encrypted mail to non-Hushmail accounts, the requirement of establishing a question to which only the sender and recipient would know the answer presented a usability issue that could not be meaningfully tested in the environment available for the study. Because

of this limitation, Voltage SecureMail Cloud presented closer feature parity for the tasks we needed to compare.

In addition to comparing usability of similar features, comparing Pwm against Voltage SecureMail Cloud also allowed us to compare users' reactions to systems requiring trivial software installation (the Pwm extension) against systems requiring account creation and verification (Voltage SecureMail Cloud).

5.3.3 Participant Demographics

Participants for this second study were recruited by posting advertising fliers nearly identical to those used to advertise the first study and by announcing the study at a campus social activity where a calendar was passed around allowing people present to sign up for a time slot. The only additional information disclosed to participants recruited in person was that the study was part of a graduate thesis project whereas the flier only stated that the study was being conducted by the BYU Computer Science Internet Security Research Laboratory.

In total, 37 participants began the study, but five were prevented from completing all tasks. In three of these cases, the blocking problem was due to an unexpected change in the configuration of the university SMTP relay server that the Pwm service used to send authentication emails as part of the SAW protocol. The remaining two system failures were due to significant delivery delays on the part of the university SMTP relay server that prevented Pwm's SAW authentication messages from being delivered in the time available for each study (in one case, delivery was delayed by approximately 42 minutes.) Survey and observational results from these five participants were considered invalid and are not discussed in this thesis.

Of the 32 participants who completed the study, 31 (97%) were current BYU students, the remaining participant was the spouse of a student participant and worked as a sales representative. Fourteen participants (44%) were male and eighteen (56%) were female.

As with the first study, we stipulated that all participants should be active Gmail users. Twenty-eight participants had been Gmail users for over one year. Only two (6%) had been Gmail users for less than six months. Twenty-seven (84%) reported that they use Gmail on a daily basis. Four (12%) reported that they use Gmail once or more per week, and only one used Gmail less than once per week, but more than once per month.

In this study, we also asked participants about how they typically access Gmail. All participants indicated that they regularly access Gmail through the webmail interface and 22 (69%) indicated that they also regularly access Gmail using mobile devices. When asked about browser use and whether they were familiar with Google Chrome, 25 (78%) indicated that they used Google Chrome on a regular basis. The remaining seven (22%) reported that they had used Google Chrome in the past, but did not use it regularly.

Three participants (9.4%) were studying in technical fields. Of these three, two indicated in discussion after completing the study that they were at the beginning of their programs and were unfamiliar with encryption and computer security. One participant, an Information Technology major, stated that he had used PGP once in the past as part of an assignment for a class but that he had not used it since and was unfamiliar with any other approaches to secure messaging. One other participant, a Spanish Education major, reported that he had worked as a programmer in the past and that he had tried PGP once out of curiosity. He

5.3.4 Study Design and Tasks

While the tasks users were asked to complete in this study tested their behavior in taking the same actions as in the first study, there were some minor differences.

First, in the initial study, all users used the same Gmail account, and the contents of the account's inbox were reset between participants. In this study, each participant was given a unique account which they used in completing tasks with both Pwm and Voltage

SecureMail Cloud. Each account was prepared with two messages in the inbox with different subjects:

- A message using Pwm
- A message using Voltage

Participants were told to open the corresponding message, identified to them by subject, to begin the tasks associated with each system.

Second, To counter bias caused by participants being familiar with one system before evaluating the other, we randomized the order in which participants evaluated each system. To ensure that the randomization was as even as possible, participants were assigned contiguous unique ID numbers. Participants with an even ID number evaluated Voltage SecureMail Cloud first, and participants with odd ID numbers evaluated Pwm first. Participants completed system-specific survey sections after completing the tasks for each system but before continuing on to the next system.

Third, in some tasks in the first study, participants were asked to send a message and wait for a reply with further instructions. In this study, the tasks were restructured so that participants could continue to the next task without needing to wait for a reply.

After completing all tasks and the corresponding survey questions, we briefly interviewed the participants about their experience. We asked all participants two core questions:

- "Was there anything in either system you used that you thought was unclear or confusing?"
- "Of the two systems you used, is there one that you would prefer over the other?"

Based on the participants' responses we sometimes asked followup questions to ensure that we correctly understood their opinions.

Most participants used the same lab computer as in the earlier study[3]. The only exceptions to this were four cases in which two participants were present simultaneously due to unscheduled or late arrival. In these cases, the second participant used a guest account on a similar computer running the same version of Chrome. This secondary computer was selected because it was positioned such that we could observe the second participant's screen while simultaneously viewing the main study computer's screen All session data on each machines was purged after each participant finished using it.

Pwm Tasks

The study tasks for testing Pwm consisted of the same actions as the tasks in the prior bookmarklet study. Each participant was asked to read the provided message with the subject containing the word "Pwm". This message was encrypted and the plaintext explanation of the message was essentially the same as in the first study, but the "Get Pwm" and other URLs had been changed to link to the extension installation page. After following the link, installing Pwm, and decrypting the message, the decrypted ciphertext contained a message telling the participant to send a protected reply to the message. When sending a reply to the protected message, the reply is automatically encrypted and required no additional user action.

At this point, the participants were asked to return to the survey. After answering some questions, the users were asked to return to Gmail and send a new encrypted message to on of our lab Gmail accounts. Sending this message in a new session was to test whether the users would recognize the lock button that Pwm had inserted in the Gmail compose view and use it to enable encryption for the message. Testing this was an important step in verifying that we addressed the problem of users accidentally sending sensitive information in the clear when they wanted it to be encrypted.

[3]The Chrome web browser installed on the study computer had automatically updated from version 19 to 20 in the time between the studies. This update in the version of the Chrome web browser did not change any functionality or interface elements that are being compared between the two studies.

After sending that message, regardless of their success in encrypting the message, the participants were asked to return to the survey to complete the Pwm related sections.

Pwm Results

Based on observations of participants' interacting with Pwm, two differences from the first study were immediately apparent:

- Participants completed installation with significantly fewer mistakes or delays
- A delay between the time Gmail appeared to finish loading and the time that Pwm was visibly running caused confusion

As with the first study, all users were successfully able to install Pwm. However, all of the users in this study completed installation on their first attempt. There are several factors that are likely to have influenced this increase in successful first attempts.

In the first study 5 of the 25 participants (20%) reported that they had used bookmarklets before beginning the study. In this second study, 28 of the 32 participants (87.5%) reported that they had installed browser extensions in the past. This familiarity with the extension installation process is one possible explanation for at least some of the increase in first-attempt installation success.

The aspect of installing the bookmarklet that caused the most participants difficulty in the first study was making the bookmarks bar visible so that the bookmarklet link could be dragged and dropped onto it. The fact that extension installation required no actions outside of the web page itself (with the exception of clicking "Add" in a confirmation dialog) eliminated the need for any browser specific knowledge and allowed the user to remain focused on the context of the web page rather than the browser chrome. The only aspect of the installation process that was not completely smooth for all users was reloading the Gmail web page after installing the extension. The few participants who did not refresh the page immediately did so after waiting a few moments and referring back to the instructions on the

installation page. Two of these participants commented that they did not initially understand that they had to reload the Gmail page in order for the Pwm extension to begin running. Manually refreshing the tab with the Gmail session was necessary for the extension to start since extensions that automatically run on a given domain are started when a page on that domain finishes loading.

When reloading the Gmail page (or loading it again in a new browser session as in the new message composition task), there was a delay of a few seconds between the time that the Gmail page appeared to finish loading and the time that Pwm started running and became visible in the page. This delay caused visible confusion in 7 of the 32 participants (22%) four of whom commented on the survey that this delay was something that they did not like about the system.

The delay is due to the fact that Gmail loads most of the visible page content quite quickly and then continues to load other resources such as some JavaScript libraries asynchronously in the background. One of the last resources to be loaded in most sessions was the JavaScript implementing the Gmail Greasemonkey API[4]. Although the extension starts running before the JavaScript providing the Greasemonkey API is loaded into the Gmail page, it has to wait for the API to become available before it can begin interacting with some necessary elements of the Gmail page.

Encrypting New Messages In the first study, the only observed participant failures that resulted in the possibility of sensitive data being exposed were instances in which participants forgot to click the bookmarklet to start Pwm when composing a message in a new Gmail session. While using a browser extension eliminated the need to manually start Pwm, users still needed to click the lock icon that Pwm inserted in the composition page, just above the primary text entry region next to existing text editor buttons. The instructions on the

[4]The Greasemonkey API is a small JavaScript library that allows ContentScript developers to access certain page regions directly. Pwm uses this API to enable the Page Scanner to quickly find specific regions of the Gmail page without relying on manually constructed CSS selectors that tend to be very fragile in relation to frequent changes in Gmail's dynamic CSS and HTML.

Pwm extension installation page stated that this button needed to be clicked in order to activate Pwm for each message that was to be protected. Most participants had no difficulty activating Pwm on new messages. Even in cases where users commented that they had not read the instructions and were unsure what they were supposed to do, almost all stated that they immediately saw the button with the lock icon and recognized that it should be clicked to secure the message. Two of these users did state that in addition to the graphical icon on the button, a text label or tool tip would have made them more confident that it was, in fact, the correct button.

In the first study, three of the 25 participants (12%) failed to encrypt a new message before sending. Only one one of these three recognized the failure without being told. In this second study, three of the 32 participants (9%) mistakenly sent a message in a new session without encrypting it, but two immediately realized their mistake without prompting and tried again. One of the two who tried again stated in his survey comments that he noticed the button with the lock icon just after he had clicked the send button and realized that he had made a mistake. The other participant who tried again said that he realized after sending the message that the compose interface hadn't had the dark grey look that Pwm had put in the compose interface when replying to a message. After encrypting and sending a new message, participants were asked to rate how confident they were that they had correctly encrypted the new message. The three who had failed to encrypt the message responded that they were either "not very" or "not at all confident". All remaining participants responded that they were either somewhat (39%) or very confident (59%) that they had used Pwm correctly to secure the message.

The next question on the survey asked whether participants would prefer encryption for new messages to be enabled or disabled by default. Overall, 23 of the 32 participants (72%) responded that they would prefer that new messages not be encrypted by default except when the user specifically enabled encryption for individual messages. The remaining 9 (28%) preferred that all new messages be encrypted by default except when encryption was

disabled for individual messages. Of the three participants who did not correctly encrypt the new message on their first attempt, one stated that she would prefer for new messages to be encrypted by default. The other two indicated that they would still prefer message encryption to be disabled by default.

SUS Score The extension version of Pwm earned a mean SUS score of 70.7 with a standard deviation of 12.28 and a 95% confidence interval of ±4.26 points. In comparison to the studies analyzed by Bangor et al. [**?**], this ranked at the low end of the third quartile, just above the median score of 70.5, an adjective rating of "Good" and lying in the "Acceptable" range on the acceptability scale.

The Pwm extension earned its lowest score on the question "I think that I would like to use Pwm frequently" with a median response of 2 on the 0-4 SUS scoring scale equating to "Neither Agree nor Disagree". A low score on this statement is not surprising since the language in the statement conflates the respondent's opinion of Pwm with their desire to send or receive encrypted email on a frequent basis. Considering that 25 out of the 32 participants (78%) responded in the survey that they agree or strongly agree with the statement "I trust Gmail with my sensitive email messages," it is reasonable to assume that many of the participants may not have had a strong desire to frequently send or receive encrypted email. Based on the language of the statement, such lack of interest in frequent secure email use would indicate a lower rating regardless of participants' opinions of the system being tested.

Voltage SecureMail Cloud Tasks

Due to the cost of licensing Voltage SecureMail Cloud services, we conducted this study using a free, 14-day trial account. The terms of the trial strictly limited the number of email addresses that could be allowed to send new encrypted messages using the trial account. There was, however, no limitation on the number of messages that could be sent using the trial account or on the number of recipients that could be addressed in each message. All

recipients who received a message encrypted using the trial account could send an encrypted reply to the original sender even though they could not encrypt new messages to other recipients. Due to this limitation, we were able to have participants test all functionality except sending new messages in a new mail session.

To begin this series of tasks, participants were asked to open a new Gmail session and open the message containing the word "Voltage" in the subject for instruction on how to proceed. This message contained a plaintext body and an attached HTML file, both of which had been generated by the Voltage service and are not specific to the individual message or recipient.. The plaintext message instructed the users to open the attached HTML file which instructs them to follow a link in the page to go to the Voltage website where they can retrieve the message contents. Once arriving at the Voltage website, users were prompted to create an account by providing their full name and setting a password. Prior to beginning the study, we instructed study participants not to reuse passwords they used elsewhere and to use a non-personally identifying name such as John Doe to preserve their anonymity when asked to provide a full name during the study. Once a name and password had been provided, the participants were directed to return to their email account to access an email account confirmation message. This email account confirmation message contained a link that, when clicked, took the user back to the voltage website where the contents of the original encrypted messages were displayed. This same page allowed users to securely reply to or forward the message by clicking on distinct buttons at the bottom of the message. After sending a reply to the provided message, participants were directed to return to survey.

Voltage SecureMail Cloud Results

As with Pwm, all participants successfully completed the task of decrypting the voltage message and sending a replay.

The most common complaint, expressed by 14 of the 32 participants (44%), about voltage was the number of steps and browser windows or tabs required to create and activate

an account to read a message. Two participants specifically commented that they did not like leaving Gmail and logging into a separate website to access their messages. Conversely, one participant specifically stated that she liked the idea of keeping the protected message separate from Gmail as it felt more secure to have all protected messages on a separate site.

In post-study interviews, six of the 32 participants (19%) stated that they preferred using Voltage over Pwm. Three of the six indicated that it was because Voltage seemed to have a more professional look and feel, one stated that it was because she did not like adding extensions which she felt bloated her browser, the fifth stated that she liked the idea of separating secure messages into a different site, and the last stated that he preferred voltage because they had a larger website with more information that gave him the impression that Voltage was well established and more likely to to be trustworthy and reliable. Overall, six participants (19%) commented specifically that they liked the simple, professional design of Voltage's web interface although one of them expressed frustration that the plaintext introductory message included images which Gmail automatically hid as a security precaution. Other aspects of Voltage that participants stated they liked included the simple, no-frills text editor (3 participants), large easily located reply and forward buttons (2 participants) and the fact that the interface "had green in it" (one participant).

SUS Score Based on participant survey responses, Voltage earned a mean SUS score of 62.65 with a standard deviation of 17.53 points and a 95% confidence interval of ±6.07 points. This score ranks Voltage SecureMail Cloud at the bottom of the second quartile amon the studies analyzed by Bangor et al. [**?**], a "Good" adjective rating, and falling in the low-marginal range on the acceptability scale. Participants' SUS scores of Voltage spanned a wide range with two participants giving a low score of 25 while two others gave its highest score of 90 points out of 100. As with Pwm, Voltage earned scored lowest on the first statement "I think that I would like to use Voltage frequently" for which the median response fell between "Disagree" and "Neither Agree nor Disagree."

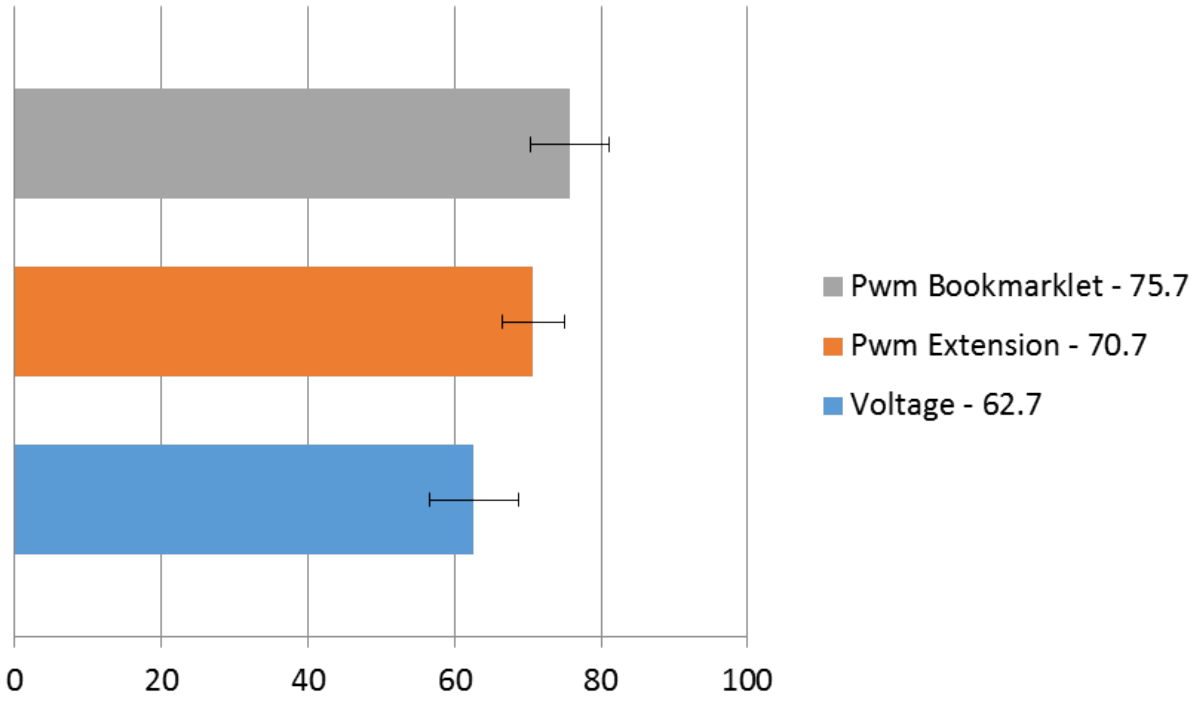

Figure 5.4: A comparison of the mean SUS scores earned by Voltage and the Pwm bookmarklet and browser extension

5.3.5 Lessons Learned

Participant responses, as well as observation of their use of Pwm revealed several trends in preferences and behavior. The most noticeable issue in observed behavior is that the loading delay in the when the Pwm extension is loaded in a new Gmail session proved to be problematic for several participants, some of whom stated that they were unsure whether the extension they installed was running due to the delay. Confusion or uncertainty in how to act due to the delay was observed in at least eight (25%) of the participants with six (19%) of them commenting that they found the delay to be confusing or otherwise disorienting. One possible solution to addressing this problem would be to add a splash screen or other indicator that the Pwm extension can insert into the page while it waits for all necessary background components to be loaded into Gmail.

A similar issue faced by some users, and noted above, was in recognizing that they had to reload their Gmail session after installing the Pwm extension. Instructions to do

this were placed clearly in the last step of installation instructions, but were not followed by all participants. While it would be possible for the extension to automatically refresh any open tabs displaying pages on the Gmail domain when it is installed, doing so would require the extension to run with security permissions that are far beyond those necessary for it's normal operation. Under the principal of least-privilege execution, running with such elevated rights when such rights are not absolutely necessary would not be advisable. An alternative solution could be to place a visible icon for the extension in the browser chrome that could refresh tabs where Gmail is open but the extension is not running. This solution would be undesirable because it would not reduce the actions required of the user (clicking a button to refresh a page) and would add extra elements to the interface, increasing the overall complexity presented to the user. Given the alternatives, the best solutions to this problem would be to make the instructions to refresh any Gmail sessions more prominent or provide a link to open a new Gmail session which would automatically start the Pwm extension.

Another, less prominent issue related to the installation instructions was that some users were confused by instructions for replying to secure messages and encrypting new messages. Three user expressed confusion because replies to encrypted messages were automatically encrypted and did not have a lock button like the instructions described for new messages. This problem can be easily addressed by clarifying in the instructions that replies to encrypted messages are automatically encrypted as well. One participant also commented that she was confused because the instructions mentioned clicking reply while Gmail's reply feature is most easily activated by clicking in a text-area which automatically begins a reply. This issue can be best addressed by clarifying the language in the instructions.

The most serious issue encountered in either study was in failures to secure new messages. While only three participants failed to properly protect new messages with the Pwm extension in this study and two of those three immediately realized their mistakes, the extension version of Pwm did not see the level of reductions in this failure that we

anticipated would be achieved by eliminating the need for participants to remember to click a bookmarklet in each new Gmail session. While the lock button (with the icon of an unlocked padlock) and the absence of the visually distinctive Pwm overlay are intended to indicate the lack of Pwm encryption to users, more prominent and active indications of the absence of Pwm encryption for unprotected messages appear to be necessary. The most promising option to correct this problem would be to exploit the always-on nature of the extension to insert visual cues in non-encrypted messages to indicate that Pwm is not active. Such cues could take such forms as a watermark over the standard Gmail text editor or a colored, labeled border indicating that the message had not been encrypted. A stronger, but more intrusive alternative would be to prompt users before sending any unencrypted messages. Such prompts would prevent messages from accidentally being sent without encryption but would add an extra step in sending all unencrypted messages that could be irritating to users. Both approaches, prompting or adding visual cues, run the risk of becoming overly familiar to users and hence being ignored. Whether such familiarity would become a problem would best be observed in a long-term study beyond what could be tested in the lab environment.

Participants in both this and the previous studies indicated in interviews when presented with alternatives to features they did not like, that they would want to have the option to choose between alternatives rather than having such decisions made for them. For example, two users in the second study expressed concern for how they would use Pwm on computers that were not theirs or where they could not install the extension. When the concept of the bookmarklet version of Pwm was described to them, they both said they would like having the option to choose which they wanted when installing Pwm. While we could not let users choose one option or the other while testing each individually in our studies, making both options available along with an explanation of the advantages and limitations of each would appear to be the best option in an actual deployment scenario.

Chapter 6

Conclusion

Pwm is a solution for providing usable secure email by augmenting existing webmail systems to provide end-to-end encryption while preserving functionality and simple usability to which webmail users have become accustomed. By tightly integrating with existing webmail systems, Pwm not only allows users to work with a familiar interface, but also eliminates the need to use an additional dedicated secure webmail system for composing secure messages or a depot system for accessing and reading secure messages that have been received. Pwm also uses key escrow and email-based authentication to simplify key management in order to avoid complex tasks that have been shown to be significant hurdles to users of earlier secure email solutions.

6.1 Contributions

This thesis presented a prototype implementation of Pwm and evaluated its success in providing a usable end-to-end secure webmail solution. The results of a cognitive walkthrough and two user studies demonstrate that Pwm is easy for new users, who are often completely unfamiliar with encryption and security, to begin sending and receiving secure messages using their existing webmail provider within minutes. This demonstrates an improvement over systems that have been evaluated in the past.

By applying established usability evaluation techniques (the cognitive walkthrough and system usability scale) this thesis demonstrates that the Pwm prototype is easy for new users to learn to use without training or outside assistance.

This thesis is the first paper to perform an analysis of the usability of bookmarklets as a means of deploying a web application. Analysis of the bookmarklet implementation of the Pwm prototype through both a cognitive walkthrough and a user study found that although the majority of users are unfamiliar with bookmarklets, bookmarklets provided a means of deploying and accessing the prototype that was both usable and easily learnable.

Users exposed to the bookmarklet for the first time demonstrated a 100% success rate at installing and using the bookmarklet once they were familiar with how bookmarks could be added in the browser being used. Once the bookmarklet was installed, the only difficulty users demonstrated was in remembering to click the bookmarklet to launch the application. This difficulty however, does not appear to be inherent to bookmarklets. User study participants who tested the browser extension implementation of the Pwm prototype forgot to click the button to enable encryption at nearly the same rate that bookmarklet users forgot to click the bookmarklet.

Users who were exposed to both Pwm's integrated approach to securing webmail and Voltage's depot style system generally showed a preference for Pwm's integrated approach. Two users who used and commented on Voltage before being exposed to Pwm stated that they would have liked Voltage better if it were integrated with Gmail.

6.2 Future Work

While the user studies presented in this thesis demonstrate that Pwm achieves the goal of usability by untrained users, there were some aspects of Pwm that could not be reliably evaluated in the lab environment. The most significant of these aspects is that of user trust. The lab setting provided an environment in which participants knew that they were safe and assumed that the study must be designed not to harm them, their property, or their privacy. Furthermore, participants were aware that even if the study and the software being tested were not entirely safe, they still were not using their own computer and email account and therefore didn't risk any loss of privacy.

6.2.1 Trust Evaluation Study

An open question is whether users will trust Pwm enough to use it when it is deployed outside the lab environment. Given the vast number of online threats Internet users are exposed to and the common advice to not click on links in email messages, will users trust Pwm messages from familiar contacts even if they have not heard of Pwm prior to receiving that message?

Further study will be needed to answer this question. Such a study will need to be performed "in-the-wild" so that users are operating in their normal environments; e.g., using their own computers, webmail accounts, contacts and message contents.

The Pwm key server provides a convenient point for instrumentation of an in-the-wild study since sender and recipient key requests go through the server. Hash digests of user email addresses could be stored in a sender/recipient table along with a field indicating whether the request had been made by the sender or recipient. This data could easily be logged to the server by modyifying the client to make a web service call to a logging service on the server. This would preserver the anonymity of the users while still allowing us to measure the number of unique users and track how well grass roots adoption is working.

It is reasonable to expect that some uninitiated users might click-through the link in a Pwm message invitation but then decide not to continue or fail to use Pwm. An experimental system could be instrumented to identify such cases by customizing the link in invitation messages to include a hash digest of the recipient's email address as part of the query string and then record it on the web server. User digests that appear in this list on the web server but are not recorded as requesting a viewer key for a message where they are a recipient would indicate a user who clicked-through to the website but never decrypted the message. Similarly user digests that appear as recipients in entries on the key server but are never logged by the web server will indicate that a message was sent to an uninitiated user but the user never followed the link in the Pwm invitation message.

While such a test does not strictly isolate trust as a single test variable, it does collectively test environmental, social and other trust related variables native to normal use cases that could not be tested in the lab.

Appendix A

First User Study Survey

The following is the survey distributed to participants in the first user study to assess their attitudes about secure email and their experience using Pwm.

A.1 Introduction

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. None of the video or audio content captured during the study will be released publicly or given to a third party. Before beginning the study, we will also ask you to provide some demographic information. None of the results published as part of this research will personally identify you as a participant.

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.

You will receive $10.00 as compensation for your participation in this study. The expected time commitment is 20-30 minutes. If you feel uncomfortable with any aspect of this study you may quit at any time.

Please advance to the next screen when ready.

A.2 Demographics

Please enter the ID number you were given

Please provide some basic information about yourself. This information will not be used to personally identify you.

Are you a student?

- Yes
- No

What is your major?

What is your occupation?

What is your Gender?

- Male
- Female

What is your approximate age?

- 18-25
- 26-35
- 36-45
- 46-55
- 56-65
- Over 65

How long have you been a Gmail user?

- Less than 6 months
- 6 months - 1 year

- 1 - 5 years
- More than 5 years

Approximately how often do you use Gmail?

- Daily
- 2-3 Times a Week
- Once a Week
- 2-3 Times a Month
- Once a Month
- Less than Once a Month

A.3 Tasks

A.3.1 Task 1

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.

Close Gmail when you are finished and advance to the next page of instructions.

Click here to open Gmail

Username: pwmstudy@gmail.com
Password: pwmusability

A.3.2 Task 2

Please log back into our test Gmail account with the user name and password shown below.

Send a secure message to gmailstudy@isrl.cs.byu.edu using Pwm. Include the ID number you were given in your message.

Wait for a reply with further instructions.

Click here to open Gmail

Username: pwmstudy@gmail.com
Password: pwmusability

A.4 User Reaction Survey

You have finished all tasks for this study. Please answer a few questions about your experience.

Please give your response to the following statements about using Pwm for securing Gmail messages. Ignore normal issues with Gmail itself.

Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you dont have a response to a particular statement.

Choices presented:
Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, and Strongly Agree

- I think that I would like to use this system frequently
- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use this system
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I felt very confident using the system
- I needed to learn a lot of things before I could get going with this system

Please give your response to the following general statements.
Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you dont have a response to a particular statement.

Choices presented:
Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, and Strongly Agree

- I trust Gmail with my sensitive email messages

- I am concerned about Gmail scanning my messages
- I feel safe sending important information through email
- I worry that some messages aren't really from who they say they are from
- I found the bookmarklet easy to use (The button you dragged to your toolbar is called a bookmarklet)

Have you used a bookmarklet before this study?

- Yes
- No

What did you like about Pwm?

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

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? (Check all that apply)

- Password(s)
- Social Security Number
- Medical Information
- Credit Card or Banking Information
- Other (Please specify)

Did you send the requested information?

- Yes
- No
- Not Applicable

Have you ever received information you were not comfortable receiving through email?

- Yes
- No

What type of sensitive information did you receive? (Check all that apply)

- Password(s)
- Social Security Number
- Medical Information
- Credit Card or Banking Information
- Other (Please specify)

If you started using Pwm on your own, would you prefer protection for new messages to be

- always on
- only on for the message that was open when you clicked "Secure my Email"
- off, unless I click a separate button on the Gmail page

Comments

Appendix B

Second User Study Survey

The following is the survey distributed to participants in the first user study to assess their attitudes about secure email and their experience using Pwm and Voltage.

B.1 Introduction

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. This will help us learn whether or not our software is easy to use. None of the video content captured during the study will be released publicly or given to a third party. Before beginning the study, we will also ask you to provide some demographic information. None of the results published as part of this research will personally identify you as a participant.
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.

You will receive $10.00 as compensation for your participation in this study. The expected time commitment is approximately 30 minutes. If you feel uncomfortable with any aspect of this study you may quit at any time.

Read all task instructions before beginning each task.

Please advance to the next screen when ready.

B.2 Demographics

Please enter the ID number you were given

Please provide some basic information about yourself. This information will not be used to personally identify you.

Are you a student?

- Yes
- No

What is your major?

What is your occupation?

What is your Gender?

- Male
- Female

What is your approximate age?

- 18-25
- 26-35
- 36-45
- 46-55
- 56-65
- Over 65

Is English your first language?

- Yes
- No

How long have you been a Gmail user?

- Less than 6 months
- 6 months - 1 year
- 1 - 5 years
- More than 5 years

Approximately how often do you use Gmail?

- Daily
- 2-3 Times a Week
- Once a Week
- 2-3 Times a Month
- Once a Month
- Less than Once a Month

Which web browsers do you regularly use? (Mark all that apply)

- Apple Safari
- Opera
- Mozilla Firefox
- Google Chrome
- Internet Explorer
- Not Sure
- Other (please specify)

Have you ever used Google Chrome in the past?

- Yes
- No

What technologies do you regularly use to access your Gmail messages? (Mark all that apply)

- Mobile Device (e.g. smartphone or tablet)
- Desktop Client (e.g. Outlook, Thunderbird, etc.)
- Web Browser
- Other (please specify)

B.3 Tasks

B.3.1 Voltage

Please log into our test Gmail account with the login name and password provided to you. Read the message with the word "Voltage" in the subject. Follow the instructions given in the message.

Close Gmail when you are finished and advance to the next page.

Click here to open Gmail

Please answer a few questions about your experience using Voltage to read an encrypted message.

Please give your response to the following statements about using Voltage for securing Gmail messages. Ignore normal issues with Gmail itself.
Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you dont have a response to a particular statement.

Choices presented:
Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, and Strongly Agree

- I think that I would like to use this system frequently
- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use this system
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I felt very confident using the system
- I needed to learn a lot of things before I could get going with this system

What did you like about Voltage?

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

Other comments on Voltage:

B.3.2 Pwm

Please log into our test Gmail account with the login name and password provided to you. Read the message with the word "Pwm" in the subject. Follow the instructions given in the message.
newline Close Gmail when you are finished and advance to the next page.

Click here to open Gmail.

Please give your response to the following statements about using Voltage for securing Gmail messages. Ignore normal issues with Gmail itself.
Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you dont have a response to a particular statement.

Choices presented:
Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, and Strongly Agree

- I think that I would like to use this system frequently
- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use this system
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I felt very confident using the system
- I needed to learn a lot of things before I could get going with this system

What did you like about Pwm?

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

Other comments on Pwm:

B.3.3 Compose New Message

Now that you have installed Pwm in an earlier task, log into the test Gmail account with the login name and password provided to you.

Compose a new message and protect it using Pwm. Send the message to pwmstudy@gmail.com. Include your ID number in the message.

Close Gmail when you are finished and advance to the next page.

Click here to open Gmail

What did you like about using Pwm to protect a new message?

What did you dislike about using Pwm to protect a new message?

How confident are you that you used Pwm correctly to protect the message?

- Not at all confident
- Not very confident
- Somewhat confident
- Very confident

If you started using Pwm to compose new messages on your own, would you prefer protection for new messages to be

- Normally on unless you turn it on
- Normally off unless you turn it on

B.3.4 General Security Questions

Please give your response to the following general statements.
Try to give your immediate reaction to each statement without pausing to think for a long time. Mark the middle column if you dont have a response to a particular statement.

Choices presented:
Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, and Strongly Agree

- I trust Gmail with my sensitive email messages
- I am concerned about Gmail scanning my messages
- I worry that some messages aren't really from who they say they are from
- I feel safe sending important information through email
- I feel safe creating accounts with usernames and passwords on new sites
- I feel safe installing browser extensions or plugins
- Creating accounts for new websites is easy
- Installing browser extensions is easy
- I feel safe clicking on links in email messages
- I feel safe clicking on links in email messages from people I know
- I never click on links in email messages

Have you installed browser extensions, add-ons or plugins before today?

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? (Check all that apply)

- Password(s)
- Social Security Number
- Medical Information
- Credit Card or Banking Information
- Other (Please specify)

Did you send the requested information?

- Yes
- No
- Not Applicable

Have you ever received information you were not comfortable receiving through email?

- Yes
- No

What type of sensitive information did you receive? (Check all that apply)

- Password(s)
- Social Security Number
- Medical Information
- Credit Card or Banking Information
- Other (Please specify)