Ideals and Reality: Adopting Secure Technologies and Developing Secure Habits to Prevent Message Disclosure

Shirley Gaw

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
By the Department of
Computer Science
Adviser: Edward W. Felten

June 2009

© Copyright by Shirley Gaw, 2009. All rights reserved.

Abstract

Development of security technologies tends to ignore difficulties with deployment in the real world. One research approach for improving adoption of secure practices and technologies is improving the usability of security technologies; however, this belies the underlying need to understand people's practices and the non-technical factors influencing adoption.

In this thesis, I examine the problems users face when adopting secure practices and technologies in the real world, with a focus on preventing message disclosure. I first examine individual's adoption of secure practices with respect to the management of passwords for online authentication with a survey of undergraduates. Next, I consider group adoption of a security technology, namely encrypted e-mail for group discussions. I consider the latter issue from two perspectives. The first perspective investigates user experiences with an existing technology via interviews with employees at an activist group who were highly motivated to protect the secret information of their employer. The second perspective investigates a redesign of secured communication of encrypted e-mail for group discussion with a web application.

Often the issues faced by users are not purely issues of increasing or decreasing the level of security theoretically attainable. Adoption is attenuated by convenience (in the case of password reuse) and stigmatization of secure practices (in the case of social meaning attached to usage of encrypted e-mail). People's models of security attacks could be more sophisticated than previously thought, for example, many survey participants understood that randomness in the construction of a password increased resistance to guessing attacks. At the activist group, people understood that encryption could protect messages against eavesdropping and seemed ready to use the technology for organizational secrets.

The challenge for researchers in the development of secure technologies is how to encourage security adoption by novel users while pragmatically increasing the level of security achieved in the real world. I present the EMBLEM (Encrypted Message Board with Lists for E-Mail) system as an example of how one could accommodate the needs of a specific group of users to encourage use in borderline cases, where the need for increased security is not obvious or the population of users is lightly connected together. I presented this system to two groups of people, one group with no experience with encrypted e-mail and one group with extensive security knowledge. While the technology itself seemed usable for novices, one concern was that using the technology was an unnecessary step. In contrast, those fastidiously practicing security seemed more dubious of adopting a system that increased usability or supported heterogeneous groups but provided less assurances of end-to-end protections. Finding a balance between these groups of users remains a challenging problem.

I frame the findings of this dissertation with an analogy to sociologist Howard Becker's work on deviant careers. Adopting secure practices can be a departure from accepted normalized practice. Understanding the factors influencing adoption of deviant practices, particularly the vital role of social networks in creating a desirability to adopt deviant practices, can illuminate the rational behind adoption and non-adoption of technically secure, but socially stigmatized practices. I further argue that more work encouraging desirability to adopt secure practices, and more generally work understanding real world deployment issues of security technologies, is a necessary future for progress in the field.

Acknowledgments

I would like to thank the following people and institutions for their support during these years in graduate school.

My committee and advisers: Edward W. Felten, Maria Klawe, Paul Dourish, Brian Kernighan, Perry Cook, and Olga Troyanskaya.

Faculty and senior researchers: Kenneth R. Wood, Patricia Fernandez-Kelly, Joanna McGrenere, Kori Inkpen, Lorrie Faith Cranor, Robert C. Miller, Batya Friedman, Jason Hong, Szymon Rusinkiewicz, Adam Finklestein, Thomas Funkhouser, David Notkin, Jennifer Rexford, Andrea LaPaugh, Fei-fei Li, Kai Li, Andrew Appel, Robert Schapire, Doug Clark, Marilyn Tremaine, Gaetano Borriello, Ben Schneiderman, Francois Guimbretiere, Richard Harper, Randolph Wang, and Edward Lazowska.

Study participants and paper reviewers.

Colleagues and friends: Kevin Wayne, Anthony Tang, Rowanne Fleck, Karyn Moffatt, David Kirk, Alice Goffman, Limin Jia, Lujo Bauer, Frances Perry, Bolei Guo, Ge Wang, Ananya Misra, Sonya Nikolova, Miroslav Dudik, Asa Rennermalm, Cristina Mora, Gregoire Mallard and Eleonore Lepinard, Sofya Aptekar, the security research group (Matthias Jacob, Brent Waters, J. Alex Halderman, Harlan Yu, Ari Feldman, Joseph Caldendrino, William Zeller, William Clarkson, and Timothy Lee), Amir Goldberg and Gil-li Vardi, Grayson Barber, Laura Felten, the Microsoft Research Cambridge Socio-Digital Systems group, the graphics research group, the Aphasia Project, Maia Ginsburg, Donna Gabai, Yael Berda, Matthew Hibbs, Melissa Carroll, Matthew Hoffman, Adriana Karagiozova, Becky Yang Hsu, Leo Coleman, Iannis Tourlakis, Nir Ailon, Leslie Hinkson, Matthew Lease, Gary Yngve, Janet Davis, Marianne Shaw, Alison Norman, Jamie Meyers, Tulika Kumar, Darrell Anderson, Daniel Dulitz and Alice Sheppard, John Prendergast, Adriana Abdenur, Sada Aksartova, Anirudh Badam, Heather Collister, Philip Shilane, Martin Schonger, Hana Shepard, Jessica Salvatore, Paul DiGioia, Jozef Muller, Mark Johnstone, Rebecca Fiebrink, Amar Sagoo, CS Staff, Ginny Hogan, Laura Cummings-Abdo, Melissa Lawson, and the Princeton rock climbers (particularly Mark Huang, Antti-Pekka Hynninen, Katherine Meierdicks, Erin Kimball, Jiayue He, Shannon Hughes, Mark McCann, and Benedict Brown).

Healthcare workers affiliated with the following institutions who have cared for me after a serious accident: St. Vincent's Catholic Medical Center Manhattan (particularly the emergency

room staff, nurses, and Drs Andrew Sands, Joseph Tansey, William Mandell, and Francis Pelham), St. Vincent's Midtown, Princeton Regional Physical Therapy, Olympic Physical Therapy (Seattle), Princeton Healthcare System Home Nurses, Princeton Dermatology Associates, and Princeton McCosh Health Center (particularly Michelle Gregory, Miriam Torres, and Drs Janet Neglia and Peter Johnson).

My family and my partner, Pierre-Antoine Kremp.

The research described in this thesis has been generously supported by a National Science Foundation Graduate Research Fellowship and funding from Microsoft, the Princeton Computer Science Department, the Princeton School of Engineering and Applied Science, and the Princeton Graduate School.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the digital world, protection against undesired message disclosure tends to focus on encryption, though authentication systems also play a significant role. Although strong encryption and authentication systems exist, they fail to provide a solution to the problem of undesired message disclosure. At its core, the nature of computer security is fraught with problems: new attacks foster new defenses; new defenses breed new, more secure systems; new, more secure systems ought to replace older, less secure systems. The problem with upgrading older, less secure solutions to newer, more secure solutions is that users are already “locked-in” to their current systems.

Economist Paul David describes the lock-in as a *quasi-irreversibility of investments* that hinders the adoption of new technologies [26]. People make choices based on information available to them at the present time. In the case of computer security, these choices can become less secure over time. Unfortunately, without information about possible future vulnerabilities, people become locked into less secure choices. David posits the development of this lock-in as a historical accident:

> Competition in the absence of perfect futures markets [can drive] the industry prematurely into standardization *on the wrong system*—where decentralized decision making [sic] subsequently has sufficed to hold it.

David’s lock-in applies to computer security: since these technologies are secure at time t, when adopted by a growing number of actors (both collective actors, such as organizations and companies, and individual actors, such as computer users at home or work), their use becomes normalized regardless of vulnerabilities appearing at time $t+1$, $t+2$, or later ("standardization-by-sheer-force-of-numbers" [59]).

My thesis considers how people adopt (or do not adopt) secure practices and how to encourage people to adapt to newer, more secure practices. I approach this problem by examining people's habits (in the case of password management practices in Chapter 2) and the habits they develop (in the case of adoption of encrypted e-mail in Chapter 3).[1] The presence of network externalities[2] within social networks encourages people to adopt secure habits such as practices and technologies that enforce message confidentiality.

Encrypted communications contain a quasi-irreversibility of investment in cleartext messaging technologies, such as e-mail, that inhibits a change to a more secure protocol. Katz and Shapiro suggest that sponsorship can foster technology adoption in the presence of network externalities (a single firm that absorbs entry barriers). In Chapter 4 I explore whether centralizing the cost of entry barriers can aid the adoption of technology for the purpose of protecting message confidentiality.

My studies of secure habits are conceptually similar to Howard Becker's studies of deviant careers [10]. Becker's use of career contingencies as a model for the development of deviant behavior applies to my studies of the development of secure habits. Careers typically refer to occupational trajectories, particularly ones with vertical mobility within a hierarchical structure; for example, an academic career might include a progression from graduate student to post-doctoral researcher to assistant professor to tenured professor. My use of the terms *career* and *career contigencies*, however, derive from the Chicago school of sociology, and refer to the terms coined by Becker. Stephan Barley argues that students of Everett C. Hughes (such as Becker) from the Chicago school of sociology broadened the term careers to encompass a more general pursuit of success. Careers came to include trajectories of individuals in the context of their social settings: [8]

> ...careers remained something that only individuals could experience but they were

[1] From a technical standpoint, these two mechanisms (password authentication and message encryption) are related as they generally focus on preventing unwanted information leaks. In spite of the relationship between the two concepts (for example, proper authentication is necessary for proper message encryption and decryption), the former mechanism establishes access privileges of a given identity and the latter enforces message confidentiality by obfuscating the message content.

[2] Network externalities refer to the benefits derived from consuming a good when many others make the same choice. Katz and Shapiro explain that there are products for which the utility that a user derives from consumption increases with the number of other agents consuming the good. [58]

> not solely of the individual's making. ...the options they foresaw and the choices they made were always limited by contextual possibilities. Careers, then, were pieced together from the string of alternatives and the set of interpretive resources offered individuals at any point in time by the collectives to which they belonged.

Becker applies careers to deviance by considering individuals in their social context in order to demonstrate the necessary conditions for departing from preexisting social practices. My research demonstrates how adopting secure practices (such as the use of encrypted e-mail, and multiple and different strong passwords) can be compared to the adoption of deviant practices, since secure practices tend to depart from standard, normalized practices (sending cleartext e-mail, reusing weak passwords across multiple accounts) and run the risk of being labeled as deviant (indicating paranoia about attacks, for example).[3] How individuals shift from their normal practices to deviant ones remains an interesting unanswered question. Using Becker's terminology, the adoption of secure practices depends on three conditions. The first condition is the transfer of knowledge, such as information about computer security threats; I will demonstrate how this alone is difficult for novices. A second condition to adopting secure practices is the availability of resources, such as availability and usability of secure alternatives. Resource availability remains the focus of both computer security research and research in the hybrid field of HCI-Sec (human-computer interaction and computer security); together, the two fields create new security systems and refine the usability of these systems. Finally, the third condition, and the theme of my work, assumes that people must learn and perceive benefits of altering their habits from standard practices to more secure ones. The perceived utility necessarily derives from a social perspective: automated attacks or amorphous opponents motivate people far less than seeing defenses as personal and organizational protection. In order to achieve full conversion of behavior, all three conditions need to be met. Rather than adopt the standard application of the term careers to describe a linear progression in an occupational hierarchy, I use careers to describe the factors that must be in place in order to achieve specific security outcomes.

In Chapter 2, I discuss college students' password management practices. My study shows that

[3] Like Becker, I use the term deviance to refer to the label applied *by those who have not adopted the practices*. Conversely, those who have adopted deviant practices would see their behavior as normal within their social network. In other words, my use of the terms "normal" and "deviant", like Becker's, is relative and contains no judgement as to the value of these practices.

students justify reusing their passwords, even when faced with evidence that diversified collections of passwords are more secure. Throughout the chapter, the study of password practices shows that users knew of more secure, available alternatives. In fact, many understood the purpose of creating random, harder-to-crack passwords and understood the benefits of avoiding password reuse. For many users, the less secure practices, however, were more convenient and the more secure practices required much more work. My research shows that convenience rather than ignorance outweighed the perceived benefits of more secure practices.

In contrast, in Chapter 3, I study how an activist organization adopted encrypted e-mail. In this case, the perceived benefit of using technology (encryption software) to protect message confidentiality was significant since it protected the organization's classified communications; however, people generally limited their use of the technology, often associating the technology with the protection of secret data. What emerges most clearly from the study is the stigmatization of excessive use of encryption software. If a user casually introduces confidentiality protection in casual conversation, one either considers the user paranoid about potential eavesdroppers or overly ambitious with an inflated sense of self-importance. The negative stigma attached to usage discouraged the development of regular and habitual practice in encrypting communications.

Introducing encryption into otherwise normal e-mail communications presented social challenges for the activist group employees. On the one hand, many of those interviewed knew that encryption software protected the confidentiality of communications, but, for the sake of normalized relations with colleagues, employees had to be circumspect in choosing the messages requiring such a degree of protection. In fact, I suspect that people tend to be conservative in their interactions, applying secure protocols only when people are part of the same organization and fully committed to the success of the operation. For example, interacting with volunteers required far more deference than with other employees and people generally avoided impositions, even if that meant increased risk of disclosure.

In Chapter 4, I explore whether system design changes could help alleviate some of the stigma (and work) associated with using encryption software while still providing more protection than cleartext messaging. Traditional software for encrypting communications provides end-to-end encryption, which translates into requiring each end user to properly configure and setup compatible systems. The process involved in becoming even a temporary user of encryption software is far more onerous than necessary, which could be a contributing factor to the negative stigma attached to adopting the technology in borderline cases, where caution shifts from prudence to paranoia.

Rather than labeling a single message or e-mail thread as confidential, I designed and implemented a system that supports the designation of entire projects as confidential, so that the project rather than any individual message imposes confidentiality requirements. Furthermore, the system is relatively agnostic to each group member's level of experience; it supports those who frequently use encryption software as well as those uninitiated to the technology without seriously interfering with their ingrained habits. However, the system relies on a trusted server, which can introduce a single source of vulnerability to attack. In addition, the system does not provide end-to-end confidentiality since it focuses on encrypted transport and message storage; groups must weigh this lower level of protection against the system if they can enforce end-to-end encryption or if such protection is necessary.

This thesis contributes to existing research in the area of HCI-Sec and aims to render secure practices more accessible. Technical experts often advise users to adopt more secure practices, but their suggestions are frequently disconnected from or lag behind actual user behavior. In certain circumstances, such suggestions are unreasonable, but in others, the precautions are fully justified. Clearly, users and researchers are faced with a dilemma: how to overcome strong barriers to more secure and better practices, whether they stem from problematic designs that burden users unnecessarily or from poor user choices.

Why are good security practices difficult for users? As mentioned above, users can take numerous precautions against attacks, but the amount of work behind these precautions seems excessive for a novice computer user. Table 1.1 provides examples of possible user precautions; each example is accompanied by an explanation of the precaution. The explanations illustrate that understanding the precautions demands a level of technical competence that many users lack. Furthermore, following advice to the letter without understanding the reasons behind it could discourage users from practicing a high level of vigilance. When users lack the technical knowledge of attack methods and defenses, they fail to see how their vigilance has improved security [84].

The examples listed implicitly describe care for a single machine, but the problem truly becomes unwieldy when faced with the numerous devices and data stores available. Users can own any number of devices on any number of platforms, including desktops, laptops, mobile phones,

User Actions	Defense Motivation
Avoid e-mail attachments	E-mail attachments can contain viruses aimed at infecting computers.
Update software and close vulnerabilities on every application that uses insecure default configurations	Software updates can patch known system weaknesses. Application-specific attacks, like the Melissa virus, also take advantage of vulnerabilities in the specific software users may run on their machines.
Run a firewall	A firewall will close holes vulnerable to network sniffing.
Avoid loading images from spammers	Loading remote images that are linked inside e-mail messages can verify e-mail addresses for spammers.
Log out of websites when finished with browsing	Cross-site request forgeries take advantage of other open webpages to circumvent authentication requirements for online transactions.
Change passwords regularly and make passwords random strings	Dictionary attacks and offline attacks use automated guess-and-check processes that find the easiest accounts to compromise.
Avoid using unsecured wireless networks and disable network interfaces (such as pulling out ethernet cables) when they are not used.	Packet sniffers eavesdrop on wireless communication in unsecured networks. Closed network interfaces provide a hardware solution similar to firewalls and prevent network-based attacks.
Don't use pirated software, music, or movies	Software, specifically Trojan horses, may include hidden malware features inside software or data. These Trojan horses otherwise appear identical to the desired item.
Run virus detection software	Malware can make infected machines slaves for launching more attacks, and virus detectors can find and remove ones that have infiltrated the system.

Table 1.1: A sample of user precautions for securing computers and explanations of the defenses. Adapted from "Actions home users can take to protect their computers" [93].

music players, and game consoles. Each of these devices may have silent vulnerabilities that give control to malware. Users have some responsibility for bolstering the defenses of such devices, but they cannot protect data stored on other machines. Many web services transfer user data from local machines to outside servers. These services, such as Google Documents, Flickr, YouTube, Facebook, and Gmail, now store documents, photographs, videos, social networks, and e-mails (and soon health records). Web services often store data without using encryption, a feature that users cannot readily change. Therefore, enforcing the confidentiality of data, preventing it from leaking unknowingly or unwillingly, seems beyond the ability of any individual user.

Public education on these issues has been ineffective so far, as I demonstrate in Chapter 2. Expecting more from educating users also seems unrealistic (and, perhaps, ill-suited) given the constantly changing security landscape, where new vulnerabilities and new attacks appear daily. Each user's circumstance also changes a system's configuration and, consequently, the vulnerabilities facing it. Automated solutions can help, but as Edwards et al. have pointed out, automated solutions may not solve these problems [33].

Why do users still work without much support? The desire to understand the human-side of computer security has garnered much interest recently, as the Workshop on Security and Human Behavior can attest [3]. It remains easier to ignore the issue than to improve it, but leaving the problem for later prolongs the pain and weakens possible solutions; a late fix will have less success than a premeditated one [102].

The field of HCI-Sec has developed as a result of recent interest in combining research in security systems with work to improve user experiences (the field of human-computer interaction, or HCI). Security problems from the human-side of computing have inspired studies of user practices and evaluations of user interfaces [25]. Experimental design and evaluation methods from HCI have contributed to user-education in the realm of privacy and security [44, 41, 101, 64] as well as measuring user susceptibility to attacks [29, 55, 45]. Likewise, design heuristics (such as Donald Norman's design criteria and Jakob Nielson's heuristics for evaluating interfaces) now form the basis for measuring and criticizing the usability of a system [69, 70, 98]. Projects focused on designing more usable systems have led to new interactions for securely completing tasks, including a system that incorporates user gestures to configure a wireless network and a system that uses eye-gaze to input a password [7, 63]. Other systems rely on new interface designs to help users avoid phishing scams [103, 28], to securely share devices [32], and to detect new identity attacks [38]. Overall, HCI-Sec has focused its energy on simple performance metrics, such as determining a user's accuracy in completing tasks, or creating designs that remove security decisions from a user's experience [22]. In practice, these approaches improve the achieved level of security, but they also have limitations. Improving the accuracy of completing a task securely requires researchers to have a firm understanding of current user practices and approaches to task completion. Removing security decisions from a system demands that designers have the ability to hide security processes, which is not always feasible.

Few qualitative studies of HCI-Sec exist, though anthropological and sociological approaches to HCI have yielded fascinating and novel results. Some of these results reveal potential for work and home technology use [61, 60, 92]; the uses range from following religious practices [99] to communicating with friends [47, 4]. Historically, HCI methodologists have advocated approaches that incorporate field work, participant observation, or interviews (for example in ethnography, participatory design, and contextual design [75]) into the development of new technologies and our understanding of technology adoption. In spite of these widely-advocated approaches, few papers demonstrate a specifically qualitative framework. Although qualitative work could analyze policy and systematic issues, thus far qualitative research has not reached beyond the process of gathering design requirements [30].

Qualitative research in HCI-Sec could offer insight into a user's practices and non-adoption of more secure practices [31, 1, 80, 95]. This thesis examines the adoption of secure practices with an emphasis on message confidentiality and OpenPGP (although, I also apply survey techniques to look at password management strategies).[4] While previous work discussed security in terms of individual actors, my analysis explores security as social interaction among colleagues or friends and further studies the implications of interactions in larger groups (see Chapter 3).

What are some approaches to solving the problem? My work contributes to the growing literature in HCI-Sec. Cranor and Garfinkel outlined two approaches to HCI-Sec research: 1) hiding security within applications which results in avoiding inconvenience to an uninformed user, or 2)

[4] OpenPGP is an e-mail encryption protocol that defines the data format for encrypted messages as well how these encrypted messages should be produced when given a cleartext message. Given the same input messages and cryptographic keys, different software implementations of the OpenPGP standard should produce the same output encrypted messages. Common software implementations of the OpenPGP standard include Pretty Good Privacy (PGP) and Gnu Privacy Guard (GPG).

highlighting security thereby informing an unaware user) [24].

Protocols such as SSH (Secure SHell) are designed to avoid user decisions: end users only agree to recognize a SHA (Secure Hash Algorithm) fingerprint but avoid negotiating the actual encryption decisions communicated between the two computers [89]. Several projects at PARC remove users from avoidable security decisions, calling for "implicit security" [89]. In Casca, the system takes advantage of physical proximity between users to provide an environment where users could easily share devices while, at the same time, abstracting the underlying security structure [32]. Similarly, the network-in-box also uses physical proximity (user gestures) to help users set up a secure wireless network [7]. Hiding the use of digital certificates, ESCAPE provides a system that invites users to view material published online with fewer and simpler steps for access control [6].

Researchers have evaluated designs that urge users to adopt more secure practices. Whitten and Tygar's seminal example highlights usability issues in PGP 5.0 with the Eudora e-mail client [98]. The first part of their study discusses the ambiguities in the PGP software interface, while the second part summarizes the observations of twelve users attempting to send encrypted messages. Following on the heels of Whitten and Tygar, Garfinkel and Miller demonstrate how colored backgrounds could help users determine whether they sent an encrypted message successfully without succumbing to new identity attacks [38]. Garfinkel et al. describe how merchants prefer suppliers who certify e-mail messages and also stress how changes to e-mail clients could simplify understanding this service [37].

Other projects outside the encrypted e-mail domain have also tried to improve user awareness, specifically, informing the user. Good et al. describe how users ignore End User License Agreements (EULAs), even when confronted with EULAs accompanied by simplified and standardized versions [43]. Dhamija and Tygar employ random graphics ("security skins") in website backgrounds to create a personalized environment for each user. Such personalized environments inform users of possible phishing attacks whenever users come into contact with a website dressed in an incorrect security skin [28]. Millet et al. proposed a just-in-time plug-in to facilitate users' understanding of cookies installed by websites [68] .

As Dourish et al. pointed out, "effective security will require that we examine the conceptual models on which our systems are built" [31]. Sasse and Flechais argue that we cannot expect tweaking the usability of existing systems to lead to usable, secure, and readily adopted systems:

more secure systems usually just add more restrictions and less control to existing systems [84]. Informative security approaches expect users to start paying attention to security if designers can simplify security. Though invisible security avoids hassling users, it makes them “pay a price for such convenience” [7]; furthermore, it may “not [be] possible to seamlessly integrate security and user goals in every situation” [89]. Both approaches have enjoyed some degree of success, but it remains difficult to apply the approaches to specific problems without a firm understanding of user security practices and work contexts.

The approaches discussed above usually to compare design A to design B. However, some researchers have adopted sociological methods to explain the success and failure of systems. Two papers have used Grounded Theory to frame observations and interviews; the theory enables Adams and Sasse to identify factors affecting secure password practices [1] and, similarly, it provides Dourish et al. with a methodology to investigate people’s strategies toward security management [31]. These efforts correspond to the work previously mentioned in qualitative HCI, as both HCI and HCI-Sec use similar methods. My work extends such qualitative research in order to understand how people interact with security technology.

1.1 Thesis Contributions

This thesis investigates the development of secure habits by considering this development as a change from normal behavior to deviant behavior, that reaches above and beyond accepted practice.[5] In order to change one’s habits, users must feel a significant incentive to make such a change, or at least feel that the change will not negatively affect them. My initial study of password management practices in individuals, where participants checked their password through actual logins to websites, was the first to quantify password practices using verification, and also the first study to weigh participants’ perceived threat of attackers by considering both an attacker’s motivation and ability to attack. Unlike previous research, my study showed that people were aware of threats to their accounts and, what is more, were knowledgeable about more secure password management practices. However, the lack of sufficient tangible benefits to alter their habits to more secure ones prevented them from undertaking change. In a second study I continued to investigate the devel-

[5] Note that Chapters 2 and 3 were originally published as research papers [39, 40].

opment of secure habits, but altered the context to a group environment. My study of activists' use of encrypted e-mail was the first published work to consider real-world use of encrypted e-mail for users who required confidential communications; the results of this study demonstrated that in order to avoid appearing paranoid about overprotecting their communications, people opted-out of using secure technology even when they knew how to use it. In response to these two studies, I designed and built EMBLEM, a mixed mode group discussion board for confidential communications. I hoped the system would relieve pressure from introducing a more secure protocol and would also remove the barrier to setting up these technologies.

Part I

User Experiences

Chapter 2

Surveying Password Management Practices for Website Logins

“Where did you meet your spouse?” The answer of “Wasilla High” led to a flurry of news articles and blog posts, for one person used this answer to break into the webmail account gov.palin@yahoo.com. The news stories recounted how someone took advantage of Yahoo’s password reset mechanism to access the entire inbox and contact book of Governor Sarah Palin, the 2008 Republican nominee for Vice President [15].

As a suddenly high-profile politician, Gov. Palin may have had little warning about infiltration of her mailbox from newly interested snoopers. Attacks like this case essentially allow someone to rifle through an e-mail inbox, but, while relatively easy to execute in Yahoo’s system, seem unlikely for most account holders. In fact, more generally, the risk of eavesdropping can seem remote for all but the most paranoid users.

To those aware of security technologies, however, Palin’s situation displayed a high profile break-in, but perhaps a preventable one as well. Both Palin and Yahoo placed only minimal barriers to prevent message disclosure. Yahoo’s authentication system (and, moreover, password authentication in general) is considered relatively weak, so a few well-aimed guesses could fool the website into believing an impersonator who claimed to be Gov. Palin. So, one method to thwart an attacker would have Yahoo applying a more rigorous authentication test. A second method, although more rare, could actively obfuscate the message itself rather than merely prevent access. Palin could have used encryption software, which I will discuss further in Chapter 3, to protect her messages stored online by making the message unreadable to all but a select few. Had either Palin or Yahoo implemented additional protections, such as better authentication systems or any encryption mechanism, an attacker might have had far more trouble accessing and disclosing Gov. Palin's communication archive.

Yahoo's system weakness illustrates one of the weaknesses of online authentication, where the security of the protection relies on the "something you know" form of authentication. Online logins require users to correctly recall a secret password for their online account, but the ability to generate the correct password represents the only barrier to otherwise unfettered access to private data online.[1] More secure authentication methods exist, but password authentication is the most popular method for implementing online authentication. Websites provide few other security checks.

In this chapter, I present my survey of how undergraduate students managed their passwords for online accounts. The Palin example helps demonstrate some of the difficulties in using automated means for identifying users. Prior research has developed alternative authentication technologies and also software to help people manage their passwords, but few projects have quantified password use based on participants' attempts to login to websites they normally visit. Few researchers have investigated users current practices to test whether or not they know about more secure practices, which presents a gap in understanding why they have not adopted these more secure habits.

In my survey, I found participants used a handful of websites and they reused passwords for these sites; I also found that participants reused their passwords more (had a higher website to password ratio) when they had more online accounts. This result is unsurprising given that participants tended to rely on their memory to store and recall their passwords. Participants cited reuse as a way to help them remember their passwords, but also recognized that avoiding reuse was beneficial for protecting private data. Participants' mental models of having their passwords compromised include attacks from anonymous hackers, but also attacks from people they know well. In fact, being motivated seemed to have a large influence in perceiving someone as likely to

[1] In the case of Yahoo, insecure password reset mechanisms made circumventing this flimsy barrier even easier by letting the requirement of providing a secret password regress to a requirement of answering their Challenge Question with a "private" answer that is, in fact, guessable.

attack (although having the ability to attack was also influential). This may also relate to how participants modeled the methods attackers used to compromise accounts. They modeled attacks as knowing personal information about the victim to guess the password rather than enumerating commonly used phrases and passwords.

Traditionally, authentication mechanisms in computers identified users in order to limit access to private data and services. This mechanism is known as *access control.* Websites also use password authentication for access control. First, access control helps users put data online without making the data universally public. Second, it helps websites implement subscription services, only allowing paid members to get content from the website. A third use of password authentication, however, implements identification for its own purpose rather than for access control: developing online communities and interaction.

Rather than isolating users, these sites can create asynchronous interaction (although synchronous interaction such as chatting is also a function offered by some websites). For example, on livejournal, each user has their own online diary. Other users who have their own journals can comment on each other's entries. In the comment section of the diary entry, the original poster can respond to a commenter and other commenters can respond to each other. Although anonymous comments are supported, users often choose to reveal their digital identities. Other sites, like Facebook, require users to always identify themselves. Facebook takes the additional step of trying to enforce a correspondence between a person's online profile and their real self, as in the case of British Member of Parliament Steve Webb. Facebook closed the account based on a report that concluded the account was for an impostor, having no other proof that the online Steve Webb corresponded to the real British politician [5].

Other websites, like the Yahoo e-mail situation, are more focused on access control and preventing outsiders from accessing private data. Isolating the actions and data of identified users (known as sandboxing) helps limit access. Websites like Flickr may also use password authentication to develop their online communities, but identifying users primarily helps websites give each user a separated virtual space for working online. On Flicker, one user's photo collection can be separated from another user's collection. In online banking sites, each customer sees only their own account information. Websites like Google Calendar also let users individually configure access control for specifying a subset of colleagues who can view an otherwise private calendar.

Some websites, like the online version of the Wall Street Journal and the online version of Harper's magazine, implement subscription services. Although both websites allow users to read some of the content for free, paid subscribers can access additional "exclusive" content. Users pay a fee for access to this content and the site uses authentication to separate those who pay from those who do not. Pandora (an online music radio provider) provides content for free, but blocks access to the site when users fail to login. This approach facilitates future use where users may later pay for additional services. This approach additionally helps sites to track users as they browse through the content of the website. More generally, users of free web services may login before accessing content but may not see a individualized environment. In these cases, user incentives to avoid payment or to avoid spurious registrations can circumvent these authentication mechanisms. One technique, epitomized by the website BugMeNot.com, collects the login information for many accounts on many websites, so that the login information for one digital identity is shared between more than one physical person. Furthermore, users may willingly share authentication information for paid services to barter for similar access on other websites [62].

The prevalence of password authentication is an artifact of previous system designs and the kinds of online interaction these systems provide. Password authentication systems are relatively simple to implement and lend themselves to online authentication. Since password authentication is implemented in software, websites need only present their login pages before the content of the rest of the website. This approach avoids special software or hardware that biometric and token-based authentication systems require. A biometric authentication system needs biometric readers to collect fingerprints, eye scans, or other "something you are" properties. A token authentication system needs both tokens (such as rings or RFID cards) and token readers. Although software implementations exist, for example where each computer holds a token rather than having a separate physical item, this does not eliminate the special requirements of token systems.[2] Because of these software or hardware requirements, password authentication makes sites more portable—the login is the same nearly anywhere on any platform. Users can access their webmail from their own laptop, and, on the road, they can still access their webmail from common terminals in airports and Internet cafes. Borrowing a laptop from a friend does not change a user's ability to access their data through a website. This portability may be one reason some novel authentication

[2] Furthermore, the tokens still represent the user but are not equivalent to this person: someone else can use the same token to masquerade as the token's owner.

systems have failed to replace weaker password authentication systems when they provide greater security but less portability.

Another difficulty with alternatives to password authentication is lack of anonymity. Decentralized password authentication, where a user can have a different online identity for every website account they have, lets users surf the web pseudo-anonymously.[3] For e-mails, the recipients of messages often do not know the exact identity of senders, but the senders cannot rely on their e-mail providers to ensure their messages are untraceable, as shown by the incident where Yahoo admitted to turning over e-mails of Chinese dissidents [86].

Given the prevalence of password authentication online and given the likelihood no other authentication mechanism will replace it soon, I present in this chapter a survey of how users manage passwords for online accounts. Relative to ideal practices, users tend to have poor password management practices, as the related work in the next section will show. These practices exacerbate the weaknesses inherent in this type of authentication. In our study with 49 undergraduates, we measure the extent of password reuse and examine users' justifications of this practice. We ask about current management strategies and use data from failed login attempts to understand where users have problems with password authentication. We also investigate users' models of attacks and attackers, which provide context to their security precautions. The large scope of this work helps us understand real users' practices along with the environment and culture that leads to these practices.

2.1 Related Work

Many projects try to overcome poor password practices, from providing password managers to empirically studying users' password choices and management strategies. Some have looked at tools for helping users to manage their passwords, particularly password hashing systems. Yee discusses several password hashing systems which can use a master password on a second identifier (such as a website URL) to generate unique passwords across different websites [103]. Often

[3] On the other hand, web portals consolidate content from multiple web services, so that one account maps to more than one service. Some tools such as onion routing and services that offer anonymous online accounts can be more effective for anonymous web surfing; however, outside of onion routing, online anonymity is more easily implemented in creating online identities that are misleading representations of the user rather than in making online behavior untraceable.

these tools try to add convenience by hiding their functions from the user. Both LPWA and PwdHash automatically substitute or fill in passwords based on specific user input [35, 81], while Site Password [57] and a remote version of PwdHash displays the generated password for the user. Browser features such as Internet Explorer's AutoComplete and Firefox's Saved Passwords similarly automate filling in passwords without displaying cleartext to the user. These functions then relieve the user's burden to memorize several passwords. One question that remains, however, is whether users adopt these tools for managing password practices. Section 2.4.3 discusses the results of my survey that asked participants about the tools they used for aiding recall of passwords.

Researchers have also conducted empirical studies of password use and management. Petrie collected passwords from 1,200 employees in the United Kingdom. The author concluded that people tended to pick passwords that represent themselves, a person's "password has to sum up the very essence of their being in one word" [72]. Using interviews of users, Adams and Sasse conclude that users lack motivation and password policies encourage users to use poor practices as a coping mechanism [1]. Weirich and Sasse further study attitudes toward strengthening password management [95, 96]. Their studies indicated users, to some degree, deny their vulnerability. These prior results indicate that participants have naive models of attacks, presuming their practices are good enough. In my survey, I asked participants about different types of attackers, having them weight the motivations of attackers and the ability of attackers. Additionally, I asked participants about the construction of passwords and what constituted a strong password. Both of these areas of the survey shed more light about the sophistication of participants' models of security attacks.

There have been few papers that empirically quantify how many passwords people have. Dhamija and Perrig used interview data from 30 people to estimate participants had one to seven unique passwords for ten to fifty websites [27]. Sasse et al. investigated several aspects of password use. They reported that the 144 employees surveyed had an average of 16 passwords, but this was not limited to online activities [83]. Two other studies have based estimations of people's passwords on surveys. Brown et al. surveyed college students and asked how many passwords they had. Students had an average of 8.18 password uses with 4.45 unique passwords ($N = 218$) [16]. Riley also used a survey to focus on online accounts. Her results similarly indicated college and graduate students had an average of 8.5 accounts with an average of 3.1 passwords ($SD = 2.028$, $N = 328$) [79]. The laboratory portion of my work is similar to this vein of research, however, my work emphasizes use of real online accounts, and I created a task where participants verified their accounts and specifically reported password reuse.

2.2 Overview of Study

I broadly studied password practices, focusing on real users' password reuse and the technology designs that encouraged (or discouraged) these practices. My study was part laboratory exercise and part survey. Unlike previous studies, I quantified password reuse by explicitly asking users to login to websites they used, counting the number of accounts and the number of unique passwords. I also used a survey to understand people's attitudes toward password management, such as how they justified their practices. I also investigated their perceived vulnerability by having users rank groups of people by their ability and motivation to compromise passwords.

Participants who completed the two sessions (the lab exercise and the survey) of the study were compensated with $10 USD. Almost all participants were Princeton University undergraduates, with the exception of one graduate student and two people unaffiliated with the university. Sections 2.4 and 2.5.1 present results from the first session, where students completed an online questionnaire (58 participants: 18 males, 40 females). Sections 2.3 and 2.5.2 present results from the second session, where students came into the laboratory. Only 49 of the original participants completed the second session (33 females, 16 males).

2.3 Quantifying Password Reuse

How many online accounts do people have? One technique to improve password security is for users to make unique passwords for each website account they own. This separates accounts so that an attacker who compromises one website account cannot reuse this information to compromise another account. Users may not adhere to this ideal practice, however. In my survey, I asked participants to quantify how many website accounts they had and how many passwords they used across these accounts.

I wanted participants to recall the websites where they had accounts and their login information.[4] Unfortunately, people are unlikely to recall more than a handful of websites they use. They

[4] In the results, unless otherwise specified, I did not distinguish between website accounts for confidential infor-

also need to check their login information online to be sure they are correct. If I provided lists of websites, participants could select the websites where they had accounts and then login to those accounts. If I provided lists of websites, however, I would miss any website the participants used but we were unaware of. Instead, I took a two-pass approach, and I developed a login task where participants make *one pass* at recording their online account information with pre-made lists and then a *second pass* with open-ended queries.

2.3.1 Method

I asked participants to bring "anything you use to help you remember your passwords (password lists, daily planners or notebooks, digital assistants, copies of bank or travel statements, copies of items in your Internet browser cache, etc.)" Of the 49 participants, six brought aids (e.g., a travel statement, a daily planner, and paper password lists). Twenty-six participants used their own laptops in the study while the remaining 23 were provided with a Firefox web browser on a Dell PC.

Participants were told that the study would ask them to indicate which websites they used, login to these websites, and write down their passwords. Using provided writing materials and a manila folder, they were instructed to track their passwords and to hide their passwords from the experimenters. They were also told that they could access e-mail accounts to help them in the experiment.

Participants estimated their use of websites and passwords in two passes. In the *first pass*, participants were directed to a CGI script that presented the names of 139 websites grouped into 12 categories (see Appendix A). Each of the websites used login authentication, although some were login services. This created overlap: at the time of the study, Expedia.com had its own authentication system but also supported Microsoft Passport. In this case, Microsoft Passport and Expedia were considered two different websites.

In each category of websites, participants indicated if they "have an account on the following websites." In cases where a participant was unsure if they had an account, the experimenters instructed them to overestimate which websites they used. Participants also included accounts

mation, such as e-mail account passwords, and website accounts for customization and user tracking, such as free online newspapers.

that were shared with family members. For each site where a participant indicated they had an account, they were presented with a webpage that instructed them to login to the website using a provided link. Clicking on the link popped open a new browser window. They were told "you have 90 seconds to try to login to the website. When you have finished, close the [website] window to return to this page." If participants spent longer than 90 seconds without responding to the CGI script, the webpage refreshed and recorded an unsuccessful login. For each site, participants were asked if they were "able to login to the website on your first attempt" although the experimenters observed participants attempting to login more than once. For successful logins, participants wrote down their passwords on a paper list. For unsuccessful logins, participants explained why they were unsuccessful at logging into the site.

After finishing all logins, participants self-reported summary statistics on the number of passwords they used in the experiment. Participants reported counts for five measures: the number of passwords collected in the experiment, the number of unique passwords, the size of classes of similar passwords, the number of password repetitions, and the number of passwords with related meanings.

In the *second pass*, participants listed sites that they used but were overlooked by the provided lists in the first pass. Participants were told to "write down all of your other passwords that you can recall" and re-report their summary statistics. They were instructed to use any tools "that will help you recall your passwords."

After completing the second pass, participants were instructed to destroy their lists in a provided strip-cut paper shredder.

2.3.2 Results and Discussion

Table 2.1 reports summary statistics for both the first and second passes of the study.[5] The number of accounts in the first pass is the number of successful login attempts, a conservative measure of the number of online accounts. The reported statistics from the second pass incorporate the information from the first pass; it was not an independent measure. The number of participants

[5] One influence on the descriptive measure of password reuse could might be passwords that were essentially the same, if participants reused passwords with some transformation, such as appending punctuation or numeric characters. I defined several possible transformations and had users group these classes of similar passwords into "families." Thus, the reported statistic on the number of families is equivalent to reporting the number of unique passwords.

	First Pass						Second Pass					
Variable	*N*	*M*	*SD*	Median	Min	Max	*N*	*M*	*SD*	Median	Min	Max
Number of Accounts	49	4.67	2.49	4	1	11	49	7.86	4.96	6	1	24
List Length	48	4.06	1.99	4	0	9	49	5.98	3.27	5	1	18
Number of Families	48	2.25	0.98	2	0	4	49	3.31	1.76	3	1	10
Size of Largest Family	46	2.87	2.01	2	0	8	49	3.35	2.35	3	1	10
Size of Smallest Family	47	1.43	0.93	1	0	4	49	1.33	0.94	1	0	5
Number of Repeated Passwords	48	3.06	2.19	3	0	11	49	3.76	3.96	3	0	25
Number of Related Passwords	48	0.77	1.34	0	0	7	49	1.18	1.62	0	0	5

Table 2.1: Descriptive Statistics for Activity Coverage by Login Task

Reason	Frequency
Didn't know the account password	46
Didn't know the account username	42
Discovered didn't have an account	15
Needed multiple attempts	6
Didn't know the account number	6
Needed the registered e-mail address	4
Entered with typographical error	3
Couldn't access browser stored password	2
Other	6

Table 2.2: Reasons Cited for Failing to Login. Multiple responses allowed.

is fewer in the first pass than in the second pass due to noise introduced by requesting self-reported statistics. One participant was confused between the goals of the first and second passes; his observations were inconsistent and, therefore, dropped. One participant entered nonsense values for the first pass and these observations were dropped. We also altered two observations of password list length in the first pass, as these observations were clear typos (e.g., a list length of "41" was reduced to "4" after discovering the number of successful logins was "4").

Out of the 139 sites presented to participants, they used a small portion of the sites (N = 49, M = 6.67, SD = 3.34. Median = 6). In the subset of sites where participants had accounts, they were largely successful at logging into these sites (N = 49, M = 4.67, SD = 2.49, Median = 4) and their password list length reflects this. Respondents indicated the first pass of the login task generally captured most sites participants used, where 24 of 49 participants said the first pass captured 75-100% of their passwords and 11 said it captured 50-74% of their websites.

Participants reported having only a few unique passwords, where half of the sample had three or fewer families in their list. Password lists could include reused passwords, or multiple entries of the same password, but participants also tended to reuse a password without transformation rather than permuting a base phrase (e.g., appending a number of the end of a password). Using passwords in a theme was relatively unpopular, as the median use of related passwords was zero.

Participants averaged 2.43 failed logins (N = 49, SD = 1.86). This included timeouts (M = .69, SD = .82) where participants were unable to login to a website within 90 seconds. Table 2.2 lists the reasons why participants said they were unable to login to websites. Even though participants were asked to bring anything that would help them remember their passwords, they still had trouble recalling their usernames (46 times) and passwords (42 times). While participants had trouble recalling both usernames and passwords; the majority of failures were from forgetting either the username or the password rather than from forgetting both (17 times). Unsuccessful logins were often for online shopping websites (JCrew, Old Navy, etc.); students thought they had accounts but realized later they used the site for purchases without logging in.

After using the initial suggestion of sites, participants reported other sites they used.[6] Although they were instructed to recall as many passwords as possible, participants still had just a few unique passwords ($N = 49$, $M = 3.31$, $SD = 1.76$, Median = 3).

If we quantify reuse as the number of online accounts per unique password, the median reuse ratio differed slightly between the first ($N = 45$, $M = 2.18$, $SD = 1.12$, Median = 2.33) and second passes ($N = 49$, $M = 3.18$, $SD = 2.71$, Median = 2), although the dispersion (variance) between the two passes more than doubled. This was due to the increased range of reuse ratios. In the first pass, the reuse ratios ranged from 0 to 5 while, in the second pass, reuse ratios ranged from .25 to 14. I was unable to detect a difference between the reuse ratios of those who used aids (laptops or paper notes) and those who relied on only memory in the first pass, $F(1,\ 44) = 0.71$, $p > .05$ as the effect size was small, $\eta^2 = .01$; the small effect and the small number of observations led to a low power, power = .12. Similarly, no difference was detected in the second pass, $F(1,\ 48) = .04$, $p > .05$, $\eta^2 = .00$, power = .05. Participants received no significant benefit from using their own machines or their own browsers and even paper aids did not help significantly.

Although participants had relatively few accounts, they still reused their passwords. In fact, we expect that password reuse will become a bigger problem over time. Figure 2.1 shows that the number of accounts increased by year in school. This difference is significant[7] at an alpha of .05, $F(3,\ 42) = 3.81$, $p = .02$, $\eta^2 = .04$; people accumulate more online accounts as they get older. Yet, the number of unique passwords did not change by year of study, $F(3,\ 42) < 1$. People have more accounts over time, but they do not have significantly more passwords.

Another way of looking at this is to see that when people have more accounts, they tend to reuse passwords more. Reuse ratios were positively correlated with the number of accounts in

[6] Some participants reported categories of websites (e.g., "blogs") rather than actual site names. In this case, we underestimate the number of sites by counting each category as one site. Some participants also report internal university sites that use the same authentication system ($N = 49$, $M = .55$, $SD = 1.00$, Median = 0). Thus, all of these internal sites used the same account information and were, therefore, not unique. These internal site entries are ignored as they were captured in the original first pass list and would over-inflate the reuse rate estimates.

[7] One caveat is that the students were unevenly distributed by year with 17 freshman, 12 sophomores, 7 juniors, and 18 seniors. Additionally, the small age range captured by year of study also reduces the generalizability of the results.

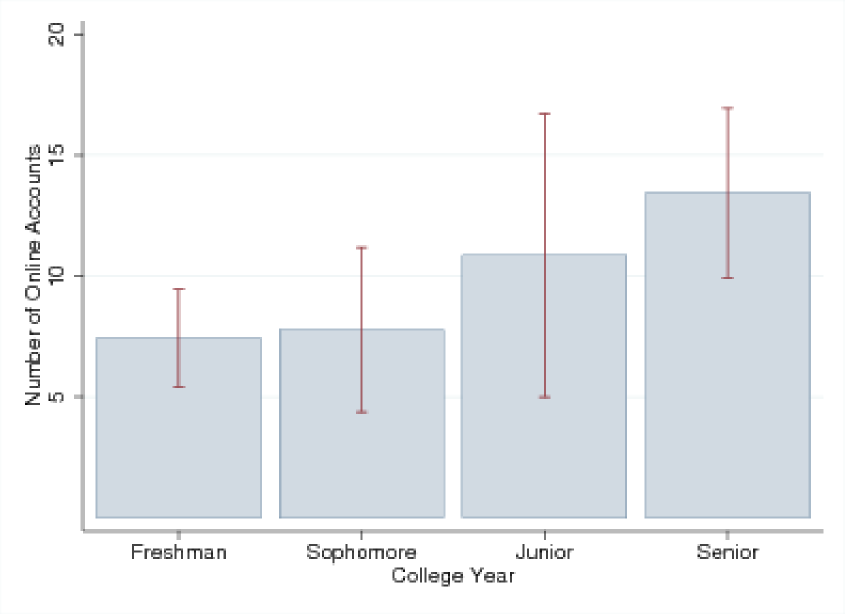


Figure 2.1: Mean number of website accounts by year of school with standard error bars.

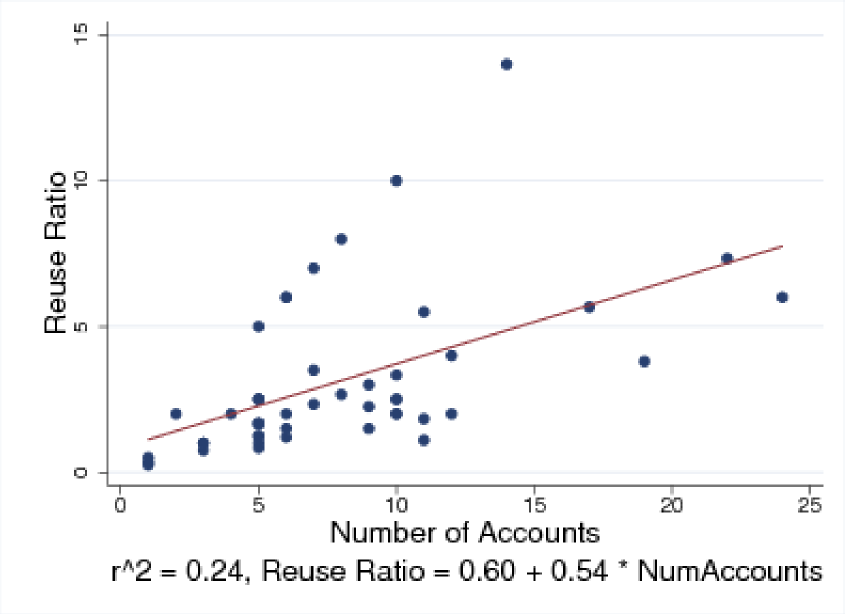


Figure 2.2: Plot of reuse ratio and the number of online accounts with login authentication in the second pass

both the first pass ($r = .68$, $N = 45$) and the second pass ($r = 0.53$, $N = 49$). A scatter plot of the reuse rate and the number of websites for the second pass is shown in Figure 2.2. This plot demonstrates that people will reuse passwords more often when they have more accounts.

The results from this section show users on average have relatively few accounts, but they already reuse their passwords within these small collections of online accounts. Furthermore, this problem of reuse is likely to exacerbate over time, although the results may not be generalizable. The results indicate an increasing problem with password reuse: people will accumulate more online accounts as time passes, people will not generate significantly more passwords over time, and people will tend to reuse passwords more as they have more accounts.

An increasing size of website collections is unlikely to be linearly correlated to age of the user in a general population. The small age range captured by the school year of the participants may simply indicate that increased socialization increases the number of accounts. Students may have more friends that tell them about websites as they develop their social circles at school over time. Furthermore, few of the websites captured by the study relate to utilitarian use of online accounts. For example, nearly all Princeton undergraduates live on campus (the university then bundles water, heath, electricity, and rent into one cost of living charge) and some have parents who pay their bills. Thus, non-students and older people may have more accounts related to online bill paying. Although I did not analyze the type of website, I speculate that many of the sites used by undergraduates are for entertainment, and therefore, their reuse relates more to privacy vulnerabilities than confidentiality of personal data, such as communications and health records. Since these accounts also involve financial information, users of these accounts may be more cautious about password reuse. Also, few (if any) undergraduates at Princeton have children, and parental use of websites may differ from other user groups, since parents may use websites related to their children's education, health, or care in addition to websites for personal use.

2.4 User Priorities

The previous section predicts password reuse will be a bigger problem as people accumulate more online accounts over time. The numbers do not show, though, why people are reusing their passwords. Prior work has indicated that security is not a priority for users and that password authentication is seen as a nuisance rather than a protection [1]. In this next section, I present the survey results of how users describe their practices and how they justify these practices.

Websites accounts are unusual in that many websites use password authentication in a fundamentally different way. The premise of password authentication is identifying the user to protect access to a resource. The motivation for the protection can have obvious benefit to the user, such as a PIN that prevents others from stealing funds. The motivation for protection may also be indirect, such as door codes that prevent outsiders from stealing from an organization. This protection is intuitively beneficial and this may obviously transfer to the online realm for websites that store financial data. For example, online banking systems such as eZCardInfo.com stores VISA credit account information but also supports money transfers from other banking accounts. Shopping sites also store financial information. Amazon.com stores credit card information and bank account information for its 1-Click shopping feature.

On the other hand, many websites are simply identifying users. Online newspapers, such as the online version of Washington Post, use logins to track users rather than protect users' accounts. Another example is Wikipedia, which uses password authentication to identify users in histories of article changes. The identified users receive little benefit, as the mechanism is primarily in place to discourage inconsiderate modification of articles. Outside of paid subscription services such as the online version of the Wall Street Journal, users receive no benefit from being identified. These systems burden the users with an additional password to manage. In addition, the accounts may remain available even when the user stops logging in. The user may forget these accounts exist but the password state (the username and password) remain. Coupled with the likelihood that people reuse usernames and passwords, users are vulnerable to attacks that collect login information on one site and use it to compromise an account on another site.

In fact, Schneier suggests that creating a Trojan site will likely help an attacker compromise multiple accounts. First, the attacker may collect login information and guess other sites that have accounts with similar information. Second, the Trojan website could encourage more collection by rejecting all login attempts. Since users can mix-up their passwords, they are likely to enumerate through their limited set of passwords [85]. Thus, the attacker could compromise multiple accounts through a *single account's* login information but also could compromise multiple accounts through a *single user's* login information for multiple accounts.

As alarming as these attacks may be, it is unclear whether attackers are employing these techniques. Identification of users who have no incentive to protect this registration information is a problem as well though. Users are habituated to poor password practices with online accounts that merely identify users rather than authenticate them for their own protection. These sites encourage users to subvert the system. They may share registration information [18]. They may also follow poor password practices through either weak password selection or through password reuse. Essentially, these websites take a protection mechanism and turn it into an inconvenience that accustoms users to bad password habits.

Given this context, it would be unsurprising to find that users justify password reuse. Looking at justifications for poor practices as well as explanations for better practices would illustrate the forces that enable poor security practices but also what motivates users to do better. This section describes our results in studying the behavior of users and the role technology has played in increasing password security.

2.4.1 Method

To reiterate Section 2.2, this part of the study is based on the questionnaire administered to students before performing the login task. There were 58 participants (18 male, 40 female) although one participant did not complete the questionnaire.

Participants took a 115-question survey. Questions included collection of demographic information, explanations of password reuse and avoidance, explanations of password creation and storage, and descriptions of password management methods. Participants were presented with both open-ended questions and also statements using 5-point Likert scale (1 = Strongly Disagree, 2 = Slightly Disagree, 3 = Neither Agree Nor Disagree, 4 = Slightly Agree, 5 = Strongly Agree) for responses.

2.4.2 Justifications of Password Practices

We asked participants if there were “two websites where you use the same password” and, if so, “why do these websites have the same password.” As Table 2.3 lists, the most common reason for reuse was that it makes a password easier to remember. One participant wrote “I usually use the same password or a variation of it, because that way I know I will always remember it.” In fact,

Reason	Frequency
Easier to remember	35
Have Too Many Accounts	8
Same Category/Class of Websites	7
Unimportant website	4
Too Difficult Otherwise	3
Only Use One Password	2
Other	3

Table 2.3: Reasons Cited for Using the Same Password. Multiple responses allowed.

when responding to our questionnaire, participants strongly agreed with the statement "if I reuse a password, it is easier for me to remember it" (M = 4.70, SD = .79, Median = 5).

Similarly participants indicated any other approach would be more difficult, requiring more management for many accounts. Only four participants justified reusing a password because they didn't care about the account, as one participant wrote "[the sites are] message board sites where it is not a disaster if someone were to crack it. I doubt anyone would take too much time trying to figure out my message board password. The only benefit would be to pretend to be me and post." As the quote suggests, people may categorize sites, where sites in the same category would use the same password. This would allow users to weight the relative benefits of reuse against the increased security of avoiding reuse. Although anecdotal evidence suggested people followed this idea by maintaining levels of security, it was not cited as a common reason for reusing a password in the free-form question. Responses only weakly agreed with the statement "I have different passwords for different security levels of websites. For example, I have a generic password for online newspapers but I have a special password for my online bank account" (M = 3.52, SD = 1.55, Median = 4).

The result was somewhat ambiguous. Some people may categorize website as "unimportant" and reuse a password while others might turn to a service like BugMeNot.com; participants did not justify reuse in unimportant websites: "I reuse a password if it is unimportant to me" (M = 3.21, SD = 1.56, Median = 3). Rather than justifying reusing a password on unimportant websites, people may prioritize creating unique passwords for important sites. What kinds of information would increase the priority of avoiding password reuse?

Students placed a higher priority on avoiding password reuse when the website contained financial data or personal communication in comparison to health information. Students agreed

Reason	Frequency
Security	12
Site Has Certain Information	11
Site Restricts Password Format	10
Important Website	7
Particular Category of Site	4
Other	12

Table 2.4: Reasons Cited for Choosing a Different Password. Multiple responses allowed.

that "I reuse a password when there isn't much financial information (bank account, credit card number, etc) about me on a website" (Median = 4), that "I reuse a password when there isn't much personal information (sexual orientation, health status, etc) about me on a website" (Median = 4), and that "I reuse a password when I use a website for routine communication (e-mail, chat, etc)" (Median = 4). Wilcoxon's matched-pairs signed rank test[8] was significant when comparing responses to protecting financial information over health information ($T(57) = 3.25$) and when comparing protecting personal communication over health information ($T(57) = 2.30$). Responses to protecting financial information versus personal communication was not significantly different ($T(57) = .01$). This sample of online users was particularly concerned with protecting their correspondence and their financial information but was less concerned about their health information. I suspect this because students are young and their health status generally does not affect their careers or insurance.

Protecting private information might motivate people to create unique passwords. I asked participants if there were "two websites where you use different passwords" and, if so, "why do these websites have different passwords". Table 2.4 shows that, for responses to this open-ended question, one of most cited reason was security; many were particularly concerned that having the password to one account would help an attacker compromise another account: "I don't like to think that if someone has one of my passwords, she or he could access all of my information for all pages I log into." Another common reason was protecting information, such as financial data: "I don't use my 'less secure' password for accounts that contain credit card information, etc." Similarly, people explained that some websites were more important than others, "for less important accounts, I use an easy password for simplicity." These free-form answers contradict

[8] When comparing responses to two questions, I tested differences in medians using Wilcoxon's Matched Pairs Signed Rank Test (T). While the t-test would be appropriate for interval measures, Likert scale responses were not always normally distributed, so I chose the nonparametric version of the t-test.

the previous result that people did not have levels of security for their passwords. Perhaps these respondents represented a subgroup of users.

An important factor in creating a unique password was a restriction on the format of a password. These websites effectively prevented reusing an old one. As one participant wrote, “different websites have different system requirements for passwords, such as some require a certain amount of capital letters, or numbers, or password length, etc.” Participants generally agreed that they have had experience with this problem: “I wasn’t allowed to use one of my passwords because it wasn’t in the correct format (too long, too short, did or didn’t have numbers, did or didn’t have punctuation, did or didn’t use capitals, etc),” M = 3.83, SD = 1.37, Median = 4.

The reasons for avoiding password reuse in Table 2.4 highlight the situations where users might be amenable to using technology to help them manage passwords. They avoided reusing a password when they believed they needed increased security. They also agreed that “it is harder to guess my password if I use different passwords on different websites” (M = 3.70, SD = 1.19, Median = 4) and that “it is harder to gather information about me if I use different passwords on different websites” (M = 3.75, SD = 1.08, Median = 4).

This leads to a pragmatic problem, however. While increased security is warranted for important websites, participants also agreed that “it is unrealistic for me to periodically create new passwords” (M = 3.78, SD = 1.15, Median = 4). They would like to avoid password reuse in some circumstances, but it is difficult to do so. One solution is using software to automatically generate a password for new accounts; however, participants strongly disagreed that “if a password is generated for me by a website than I use this password instead of one of my normal passwords” (M = 1.69, SD = 1.23, Median = 1).

There could be several reasons why participants do not use passwords that are generated for them. First, these passwords are often temporary and users are given specific instructions to immediately change the password to something different than what was generated. Secondly, people may want to reuse their password – they could just change the generated password into something they commonly use for an unimportant website.

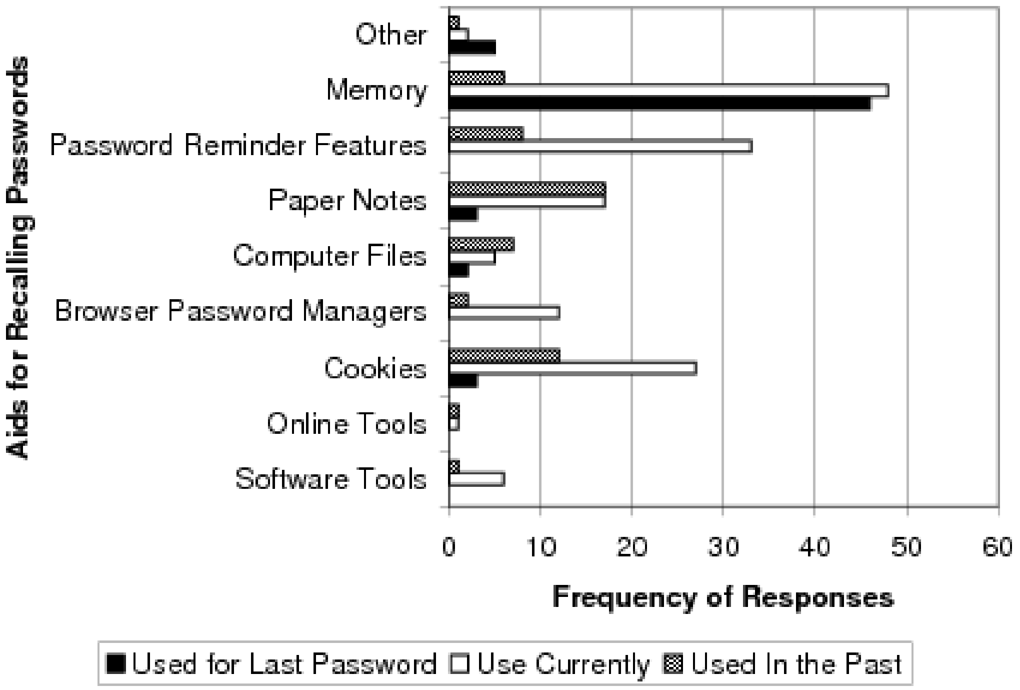


Figure 2.3: Aids Participants Cited Using to Help Recall Passwords. Multiple responses allowed.

2.4.3 Methods of Storing Passwords

Although participants avoided generated passwords, participants might have other tools they were comfortable using. I surveyed the participants to determine what they used to help manage their passwords. Participants were instructed to think about the last time they created a password and were asked where they stored their last password. As shown in Figure 2.3, participants relied on their memory. I also asked participants to select which aids they used currently to manage their passwords and which aids they had "used in the past but no longer use." Again, memory was more commonly used than any computing technology, and the technology helped for a limited number of participants. Participants also relied on password reminders such as a website feature that asked "Forgot your password." This mechanism often relies on e-mail authentication to, in turn, authenticate the user. An e-mail message is sent to the user's registered address, which either sends the password or provides the means for resetting the password. In the case of Sarah Palin's Yahoo e-mail account, mentioned in the introduction, this mechanism reset her password but relied on publicly available information to do so. This approach reduces the security of the authentication. On the one hand, this method is convenient when users need to access websites but, on the other hand, this method relies on recalling the registration information and having access to a predefined e-mail account (in cases where the password is resent instead of just reset). This method is also inconsistent across websites and users may experience a delay between when they want to access an account and when they can finally do it.

In the study, participants indicated they relied on website cookies, given the example of "Website checkbox: 'Remember my password on this computer.' " Like using password reminders, when people use cookies, they are not recalling their password at all. There are also disadvantages to using cookies. Users must avoid logging in from another machine (which requires recalling the original password); they must avoid clearing their cookie cache; and they must expect no other user will login to the same website on the same browser on the same machine.[9]

While paper notes such as Post-it notes, a notebook, or a day planner were the most commonly abandoned tools, the digital equivalents were relatively unpopular. Few users claimed to use an online password manager like Gator eWallet or PasswordSafe.com and few claimed to use software password managers like Password Agent, Password Tracker, or Any Password. Instead, they used managers embedded in their browsers such as Internet Explorer's AutoComplete, Netscape's Password Manager, and Firefox's Saved Passwords. Even this was relatively unpopular. These features were less popular than relying on memory. Yet, when users rely on their memory, they have to store, recall, and type their password rather than having it automatically collected and filled in. Some participants might have been concerned that these features are vulnerable to attack. Like cookies, browser password managers could allow unintended users access accounts they should not.

Password managers in the browser also tie users to a particular browser on a particular machine. The issue might become obsolete with devices that store account information on behalf of the user, such as cell phone-based authentication [9] or password managers stored on devices or jump drives. One example of the latter case is the application Portable Firefox, which is a zipped version of the Firefox browser that can be stored on portable devices such as USB jump drives, became more popular. Portable software would provide the benefits of browser password managers without the

[9] Using cookies may also make users more vulnerable. While the New York Times's online newspaper users authentication to track users, traditionally password authentication protects access. If users are accustomed to using cookies, they may inadvertently expose financial data. For example, when someone uses cookies in Amazon.com, they implicitly agree to purchases whenever someone accesses the same browser.

problems mentioned earlier.

In this section, I have discussed some of the tools users could use to help manage their passwords. These tools were, for the most part, unadopted, and users tended to rely on their memory. This lack of adoption seems inconsistent given that participants indicated some private information warranted increased password security, but other results from the same survey indicated this system (using their own memory to recall passwords and reusing passwords to make recall easier) was easier than the alternatives.

2.5 User Models of Attack

In this section, I discuss user's perceived threat models for what they believe makes a strong password and for who they believe is likely to attack their online accounts. This work is pertinent to understanding the adoption of aids for managing passwords as the results offer insight into whom participants believe their passwords need protection from and how they understand password strength.

2.5.1 Perceived Threat by Others

I first considered whom participants saw as likely attackers to online accounts. I wanted people to consider situations where someone would feel motivated to attack and also to gauge their ability to attack online accounts. I was interested in seeing if motivation or ability had a greater weight in perceiving someone as likely to attack an online account.

Method

Participants completed the attacker ranking section of the study in conjunction with the password management questionnaire in the first session (57 observations).

Participants were first provided with examples where a password could be compromised, for example: "A stalker could guess a password after learning some personal facts, like a pet's name or a social security number." The online questionnaire then asked participants to rank types of people by their likelihood to compromise passwords. I would have preferred to present categories with all combinations of three independent characteristics (personal relationship, computer expertise, affiliation), but using all of the combinations would have been too many choices for meaningful rankings. Instead, the population was partitioned unevenly: someone you know well (abbreviated as “friend” in this chapter), someone without computer expertise that you have met (“acquaintance-nontech”), someone with computer expertise that you have met (“acquaintance-expert”), someone from your organization that you do not know (“insider”), someone from a competing organization that you do not know (“competitor”), and someone that is unaffiliated that you do not know (“hackers”).

The instructions first told participants “ignoring their motivation, for each person listed below, write at least one scenario where this person could obtain one of your passwords.” Afterwards, participants were asked to rank the six categories of attackers three times. In the first ranking, participants were asked to rank attackers by their ability to “access information without permission from one of your web accounts.”[10] They were instructed to ignore motivation and consider ability only. In the second ranking, participants were instructed to rank people by their motivation to compromise passwords, “ignoring ability and considering motivation only.” Finally, participants were asked to rank attackers by their likelihood to attack an online account, “considering motivation and ability.”

Results and Discussion

While the rankings themselves might prove interesting, I focus on the extreme ends of the rankings, those considered most and least likely, able, or motivated to attack respondents accounts.

Considering the ability ranking, users’ perceived threat models might emphasize attacks from hackers, but my findings suggest that users realize the threat posed by those closest to them. Friends were considered most able attackers (51.79% of 56 responses). Surprisingly, only 10.71% of respondents thought a hacker who had no personal connection would be the most able attacker. In fact, 35.19% (19 of 54 respondents) said an “unaffiliated stranger” would be *least* able to compromise their passwords.

In the motivation ranking, most respondents (62.50%) indicated that a competitor or a hacker was the most motivated to attack and that a friend or an acquaintance without technical experience

[10] Although I describe the categories of people as “attackers” in this chapter, I avoided using “attack” to describe compromising passwords as I predicted people might only consider those with malicious intent. By saying the passwords were compromised “without consent”, I ignored the situations where someone would willingly disclose their password, a situation studied by Singh [88].

would be the least motivated. This is the typical response I would expect, as one respondent explains:

> Anyone who wants to [compromise a password] can, you don't need to know them, and the shield of anonymity may make it less morally reprehensible to do so.

This is even considered a normal belief of who would attack:

> This is my standard perception of identity theft, where retaliation from young people or hackers tops the list.

Yet, a minority of participants had an opposing ranking. Overall, a quarter of responses (15 of 56) noted that a friend would be *more* motivated than a hacker. A quarter of responses said a hacker would be *least* motivated. In fact, 9 respondents (16.07%) said that a friend would be most motivated and a hacker would be least motivated. I was surprised by this result, as I believed people assumed those who had a personal relationship would be most trustworthy. A few respondents mentioned those closest to them had more opportunities to build ill will:

> My ex-boyfriend falls into the top category and I'm concerned....

Two female participants, including the above participant, revealed that former boyfriends had or possibly had attacked their accounts. This is consistent with work by Dourish et al. [31] which observed female office employees were concerned about stalking. Other participants mentioned that a personal tie made a familiar attacker motivated by curiosity:

> The closer they are, the more curious they may be.

When considering overall likelihood of compromise, participants seemed to weigh both motivation and ability. I used a fixed effects logistic regression of all six types of people to predict the odds of considering an attacker most likely to compromise a password based on 1) their ability and 2) on their motivation. This techniqued helped separate the effects (and thus assess the relative importance) of perceived ability and perceived motivation on ranking someone as likely to compromise a password. When an attacker is perceived as very motivated, the odds of considering him/her a likely attacker are multiplied by 6.28 (or $e^{1.83}$), regardless of the type of person (friend, hacker, etc.) or their ability. When people perceive someone as having great ability to compromise their password, the odds of considering him a likely attacker are multiplied by 3.82 (or $e^{1.34}$), regardless of the type of person or their motivation. Both effects are significant at the .05 level. The effects of ability and motivation on perceived likelihood to attack are strong, and the effect of motivation is stronger than the effect of ability on perceived likelihood of attack.

The responses to the motivation and likelihood rankings indicates that respondents subdivided into two camps, the first believing those closest to them would be interested and would actually attack their online accounts while the remaining thought that hackers were the most motivated and likely attackers. Motivation has a strong effect on perceived likelihood of attack. One possible explanation is that gender affects the rankings, but the distribution of the most motivated attacker rankings were not related to gender.

Participants believed those closest to them had the greatest ability to compromise their passwords. This is consistent with Petries's hypothesis that passwords are one word personal identifiers: those closest to the responses are most able to guess the content [72]. Notice that hackers are considered *least* able to compromise passwords – it is as if users believe using personal identifiers is safer because guessing the passwords requires personal knowledge. This belies the actual technique used in dictionary attacks, where attackers collect common identifiers that are often used in passwords.

2.5.2 Perceived Strength of Passwords

If users expect that having a personal connection to the attacker presents an advantage, I also expected that this influenced what users perceive as strong passwords. I was interested in seeing what users thought were strong passwords. For example, Princeton provides a webpage that explain how to create strong passwords [76]. I wondered if students understood these tips.

Method

Participants completed this part of the survey in conjunction with the login task in the second session (49 participants).

Participants were presented with the following scenario:

> Many websites have tips and rules for creating strong passwords. Pretend your friend Eve Jones (evjones@princeton.edu) is also a student at Princeton and she is having trouble understanding these rules. For each rule or tip, she's provided three example passwords with an explanation of how she created her password. Help her learn what makes a strong password by ranking her examples from strongest to weakest and explaining your ranking.

This was followed by a series of eleven statements, which were chosen by finding webpages that suggested methods for creating stronger passwords:

1. Use uppercase and lowercase letters in the password.
2. Use a password of at least six characters.
3. Avoid common literary names.
4. Mix up two or more separate words.
5. Create an acronym from an uncommon phrase.
6. Avoid passwords that contain your login ID.
7. Use numbers in the password.
8. Avoid abbreviations of common phrases or acronyms.
9. Drop letters from a familiar phrase.
10. Use homonyms or deliberate misspellings.
11. Use punctuation in the password.

For each participant, statements were presented in a random order. Additionally, the three example passwords were presented in a random order. I tried to construct the passwords so they were approximately equal in length except for the case where increasing length was a suggestion. As I needed to create short explanations, all of the passwords were relatively weak. Each included some randomness, whether it was a randomly capitalized letter, a randomly selected set of characters, or a randomly added or subtracted character. In the explanations of password selection of Eve

Jones, I avoided the use of the word “random”, which would likely influence the rankings. One example password was “01/12/85” and its explanation was “this is my birthday.” An example of a random password was “snyfe” which had the explanation “I took the first or last letter of words at the end of paragraphs in an excerpt from the Undergraduate Announcement: (s = interests, n = information, y = year, f = field, e = education).”

Results and Discussion

I collected responses to the rankings and a cursory analysis indicated that participants understood that randomness was beneficial to password strength, but they linked this to human guessability. If this were the case, the explanations of password rankings would frequently describe human attackers and include some notion of randomness. With a total of 17,035 words collected from participants’ password rankings, I constructed a word frequency list and checked if these terms occurred regularly. Removing parts of speech such as articles and conjunctions from the list, the following words are top ten unique words and their frequency of occurrence: it (408), one (252), password (221), secure (217), random (215), most (180), guess (176), letters (172), not (169), first (157).

Password strength was indicated by the words “secure” (217), “random” (215 and “randomly” - 11 and “arbitrary” - 11), “common” (89 and “commonly” - 14 and “obvious” - 39), “crack” (13), and dictionary (10). Having common words or phrases was seen as a negative quality (for example, “the third is least secure because it’s a common word backwards”) while having randomness was seen a good quality (for example, “the password incorporating personal information is the least secure, whereas the first two are more random”). Although participants did not use security terminology, the prevalence of randomness and commonness in their explanations implies users understood they could avoid terms that are frequently used in passwords. Yet, respondents rarely mentioned cracking using dictionaries. In fact, “hack” only occurs 7 times, with “hacker” (1) and “hack” (1) also appearing infrequently.

I believe that participants conceptualized attacks from a human using a guess-and-check technique. They frequently referred to “guess” (176) and related words: guessed (26), guessable (11), guessing (4) deduce(d) (4). Participants would write things such as “[the] 1st because its fairly simple to learn but not a word someone would guess.” Guessing was often equated with a human attacker guessing the password. Participants frequently referred to people: “someone” (69), “people” (56), “anyone” (27). For example, participants would write “the last one is very obvious to anyone who knows the user” or “The first one though is just nonsense and random so I doubt anyone could guess it.” Participants also discussed accumulation of knowledge: “know” (39), “knows” (36), “known” (17), and “knew” (10). In particular, people indicated that knowing the victim would help crack a password: “PrincetonNJ is too easy for someone to guess if they know where you live” or “one would have to know her decently well to know her favorite novel.”

These explanations overlook the common techniques for cracking passwords. Humans may guess how a password is constructed, but attackers can use automated tools for enumerating all of the possible choices. They can do this without directly knowing personal information. Dictionaries can store combinations of all cities and states, all valid telephone numbers and social security numbers, and many phrases from literature. Yet, participants associated this content with personal knowledge. The results from this activity indicate participants were unaware of the size and breadth of available dictionaries for attack and were unaware of how these dictionaries could be constructed. Preferably, users would understand that people have common techniques for creating passwords and these passwords could be cracked given enough attack attempts and provided with large enough potential password lists.

2.6 Survey Implications

How can we practically encourage users to avoid reusing passwords? They see creating new passwords as difficult and they see avoiding reuse as helping increase security, yet they see using more than a few passwords as onerous strain on their memory. Technological solutions could help in each of these cases. There are several tools for generating passwords and tools for generating passwords in specified formats. People can avoid reuse when a computer stores and retrieves a password (or in the case of stateless password managers, regenerates a password). This lightens the memory burden.

Despite the evidence that users rely on their memory, few technological solutions support that habit.[11] Instead of helping users recall their passwords, many tools hide passwords from users. The

[11] Alternatives to text passwords mentioned in the related work section, however, try to take advantage of the human ability to recall other kinds of information such as faces or pictures.

incentive to use a browser password manager is that it keeps users from retyping their password when logging into a website. At the same time, this also prevents the user from learning their password and increases their dependence on a particular browser. Assuming people are not using portable browsers or portable devices to manage their passwords, this convenience becomes an annoyance when they need to login from another location. Instead of just storing the passwords and filling in forms for the users, the browser could help users learn their passwords. For example, rather than filling in the password, the browser could display it with a low-contrast background. This could help remind users what their passwords are on a commonly used machine - it matches a website to specific login information. Once the association is learned, the user could stop using the browser feature and rely on their memory. Thus, users could be helped to create strong, unique passwords through generators and remember the passwords with browser hints.

Websites could also change the way they authenticate users. Any site that sends password reminders over e-mail essentially uses e-mail to authenticate the user. It inherently expects users receive e-mail messages quickly. Some sites provide another mechanism for authenticating users, however. Instead of querying usernames when users forget their passwords, websites ask users to provide an e-mail address, check their registration data for a match, and send an e-mail message to that address. From the message, users can be directed to a page that not only logs them in, but also installs a cookie that identifies the user. Each time the user logs in from a different machine, they can repeat the process. The benefit of this system is that it relies on a single password that the user places on a single server rather than allowing the user to distribute this information across multiple websites or to use a single service (such as Microsoft Passport) that distributed this information across multiple websites. This approach would not weaken the security of systems that already use e-mail for password recovery or reset. Anyone who had access to the victim's e-mail account could already compromise these existing systems. This removes password state and also eliminates a situation that promotes password reuse, but business incentive structures may not support alternative authentication systems.

We could also promote better password security practices. When registering at a new website, users select a unique password. Once chosen, there is little incentive to change a password – in fact, participants agreed that "I don't have a reason to change the password on the websites I use" (M = 3.58, SD = 1.26, Median = 4). Unfortunately, when a user creates an account they have little motivation to generate a unique password. They have not yet started storing private or financial information on the website. Reuse is encouraged because it makes a password easier to remember. Furthermore, with a new account, the site is not important yet and users cannot predict how often or frequently they would use the account in the future. Even with online stores, users may not store financial information until they become accustomed to shopping at the store. In sum, when users register with a website, they have not yet disclosed anything worthy of protecting with a password.

As time passes and after building a relationship with a website, users are locked into their reused password. Once they have started reusing a password across multiple sites, not only would it be a burden to remember a unique password, they would have to remember which site had the new password. Participants agreed that "I try to use the same password for multiple websites, so it would be inconvenient to change the password" (M = 3.84, SD = 1.25, Median = 4). Compounding the problem, websites rarely ask participants to change their passwords. Websites need to attract and retain users; enforcing security policies might drive users away.

If sites avoided authentication before users invested private information, the websites could transition users to password authentication. The sites could choose a time when users are motivated to protect an account and when users understand the benefits of avoiding password reuse. Again, the vendors may have little incentive to do this.

We can also push the solution to browsers. While browser password managers ask users for permission to store newly entered login information, they do not monitor online use of website accounts. Browsers could notice frequent use of an account and suggest refreshing that account's password. In fact, our participants strongly agreed that "I will change a password if have been asked to change it" (M = 4.25, SD = 1.00, Median = 5). Currently, browser password managers and stateless password managers interrupt the user's behavior when they have the least motivation. Instead of presenting stronger password construction or suggesting easier password management at an appropriate time, the browsers present their benefits when the user is unaware of the problem. Instead, the browser could use browser history as a development of trust, as suggested by Yee [103]. Once the user has returned to a site multiple times, the browser could suggest that changing the password to something stronger as it appears the site has grown in importance.

2.7 Conclusions

While multiple papers have studied password security, the work I presented in this chapter has developed a broad description of password management strategies for online accounts. Like some prior work in the area, I quantified password reuse, but my unique method helped me measure actual login attempts rather than relying on participants to recall website use. The method also elicited explanations for why participants had trouble with logging into websites and demonstrated that using memory aids or a personal laptop had a negligible benefit for password management. The questionnaire on password management strategies also demonstrated that people relied on their memory. Even though people have access to a computer and the Internet when logging into online accounts, I was able to show the technology they used did not help them with recalling their passwords.

While participants understood the benefit of having randomly generated passwords, they still picture human attackers. Current tips for strengthening passwords fail to explain the nature of dictionary attacks. Participants then saw strengthening passwords as making it difficult for a human to guess it. This was demonstrated when participants ranked those closest to them as having the greatest ability to compromise their accounts. Participants suggested that simply knowing personal information would be beneficial to compromising a password. This implies users understood personal information might be used in the construction of a password and that some words or phrases may be commonly incorporated; however, the model fails to account for the construction of dictionaries, which could enumerate these possible passwords without having a personal connection to the victim.

My findings also indicated that the nature of online accounts and tools for managing passwords in online accounts enable poor password practices rather than discourage them. There is a gap between how technology could help and what it currently provides. My study was specifically interested in how technology could be used to ameliorate the problem of password reuse and the participants came from a technologically savvy demographic: they are both well-educated and well-connected. According to Princeton University's Student Computing Initiative, 90% of last year's incoming freshman owned their own computers and used the campus network service [91]. Over half of my participants used their own laptops when they came into the lab. I could argue that the students represent the forefront of what to expect with online activity: these users easily adopt new technology and have a culture of computing. Yet, the study findings indicated that despite their technical abilities and education, they still had trouble understanding the nature of some attacks. Rather than hinting at impending proliferation and adoption of new password management tools, these students demonstrated that the available technology has not aided password management. Furthermore, they demonstrate that password reuse is likely to become more problematic over time as people accumulate more accounts and having more accounts implies more password reuse.

Chapter 3

Barriers to Adopting Encryption for Group Communication

In the previous chapter, I presented a survey of password management practices. Password authentication is a mechanism for access control. It is a common mechanism for preventing message disclosure in that password authentication prevents arbitrary access to message contents; people must log into their e-mail accounts. Furthermore, password authentication is usually used in one technology for enforcing message confidentiality (encrypted e-mail). Password authentication is the primary mechanism for protecting encryption keys used encrypted e-mail systems. Thus, while this chapter focuses on message confidentiality through use of encryption, the two mechanisms (encryption and password authentication) are intimately tied. Any gains achieved in increasing the adoption of encryption will be undermined by poor password practices.

The rarity of encrypted e-mail is the simplest (and perhaps the best) example of a technical promise unfulfilled in actual practice. The study in this chapter presents interviews with activists who have dealt with balancing a preference for secured communication against a predisposition for using unprotected e-mail. I discuss the circumstances where general e-mail users have had their communications monitored and speculate on why one protection mechanism, encrypted e-mail, has remained relatively obscure. I then explain some technical details of how e-mails can be monitored as well as how public-key encryption can protect these e-mails. These background sections provide a context to the study of the activists. Their use of encrypted e-mail makes them an unusual subset of e-mail users. I will describe how these employees self-identify their work as dangerous, and later, how some employees displayed concern about surveillance. I also compare the situations where some employees vigilantly protected their e-mails while other employees pragmatically sent messages in the clear. Finally, I discuss some limitations of the study and limitations on the generalizability of the results.

3.1 Background: Surveillance Examples

Encrypting the body of an e-mail message prevents eavesdroppers from learning the contents.[1] Yet, encrypting e-mail remains an abnormal practice, despite more than ten years of availability as open software[2] and despite integration with most e-mail clients.[3] Despite attempts to make encrypted mail easier to use, universal and routine use of encrypted e-mail has never materialized in the general populace.

While e-mail users may lack concern about those eavesdropping on their communications, the reverse has a clear disparity: governments *are* interested in surveillance. Recent legislation in the United States (US) has broadened the government's ability to conduct domestic surveillance [2]. Mark Klein, a former employee of AT&T, testified in Hepting v AT&T that he observed the construction of a secret room in AT&T's facilities for the National Security Agency (NSA). This room received one copy of all voice and data communications that passed through this portion of the Internet backbone using split fiber optic circuits; this technique would allow the NSA to monitor all unencrypted voice and Internet traffic that passed through the facility [52]. The Chinese government provides another example of government interest in surveillance. The government has routinely censored data, especially dissenting opinions, flowing into and within the country [23]. In addition to government enforcement of restricted communication, Chinese government surveillance

[1] Some caveats exist, such as attacks that log all keystrokes on the sender's machine. In the case of United States v. Scarfo (2001), Criminal No. 00-404 (D.N.J.), representatives of Scarfo moved to suppress evidenced gained by the Federal Bureau of Investigation (FBI) after it installed key-logger software of Scarfo's computers. The key-logger software recorded the keystrokes of Scarfo's password to his encryption software and the FBI used this information to retrieve the contested evidence [94]. In this case, the FBI could not recover the encrypted data originally but could after having the password.

[2] Pretty Good Privacy and its derivative software packages have been available since 1995 when Phil Zimmerman published the source code to PGP 3 [105].

[3] Microsoft Outlook, Mozilla Thunderbird, Eudora Mail, and Apple Mail all provide built-in support the S/MIME (Secure / Multipurpose Internet Mail Extensions) protocol for encrypting e-mail messages

may include the cooperation of companies such as Yahoo China. The corporation may have turned over copies of e-mail messages as well as login and registration information relating to a political dissident movement [86]. While the American public may be blithe about surveillance of their communications, at home and abroad, governments are attentive to these conversations.

People may believe that they have nothing to hide in their messages and that no one would be interested in reading them [105], but, as shown above, government surveillance is one example of interest eavesdropping. Legal cases have shown corporate interest in eavesdropping as well. For example, in the case of Michael A. Smyth versus The Pillsbury Company (1996), Smyth alleged the company wrongfully fired him. Pillbury's management fired Smyth after he sent "inappropriate and unprofessional comments" over the company's e-mail services. Although Pillsbury had previously assured employees of the confidentiality of their e-mails and that the contents of their e-mails could not be used against them, Judge Charles R. Weirner concluded that "the company's interest in preventing inappropriate and unprofessional comments or even illegal activity over its e-mail system outweighs any privacy interest the employee may have in those comments" [90]. Employers have an incentive to police internal communications that could misrepresent their company and an incentive to police abuse of their service when that services is used to threaten others [97]. In this respect, most people should expect that their employer monitors e-mail passing through their employer's network.

Perhaps, most people should even expect that *any* e-mail provider has an interest in peeking at the mail messages passing through their systems. Google explains that their targeted adverstising system works by automatically searching through GMail messages to link message contents to their advertisements. While they assure that they aggregate their data so that no identifiable information leaks [46], mail providers can find it advantageous to arbitrarily read messages, such as in the case of the United States v Bradford C. Councilman (2005). Councilman, as Vice President of Interloc, Inc., had the Interloc mail servers routinely copy messages received by interloc.com users. By siphoning off messages sent by Amazon.com (a competing online bookseller), Councilman hoped to gain information that could help Interloc's business. Similarly, some businesses have some incentive to provide users with e-mail account services. Wary users of these services may therefore suspect that the businesses pursue these incentives, including surreptitious eavesdropping.

Cases of eavesdropping highlight instances where encrypted e-mail could thwart wiretappers (and could make monitoring more difficult for law enforcement). For the vast majority of users, the contents of their e-mails are probably innocuous, and, furthermore, these message likely generate little interest even among those who eavesdrop.[4] One argument for using encryption more generally, however, is that encryption helps obfuscate traffic (see Section 3.8.1 for more discussion). This has led to encouraging users to universally and routinely encrypt messages. In this chapter, however, I discuss whether this advocacy is sensibly placed from both the usability of the software (as shown in previous work) and the utility of encryption.

3.2 Background: Influences Discouraging Adoption

The purpose of encryption is relatively intuitive: conceal messages in a process that garbles an input message so that the content of the output, known as *ciphertext*, looks disconnected from the original content. Modern encryption techniques are computationally intensive, using, but furthermore requiring, both computer hardware and software to implement the encryption algorithms and also to generate and to store the secret keys.

The complexity of the algorithms used make them harder for people, even reasonably educated people, to understand. This includes both the algorithms themselves but also the design choices in the algorithms. In the past, anyone with enough manpower and time could check an encryption algorithm's output, figuring by hand how an original message became concealed using the process outlined by the algorithm [74]. New encryption algorithms are too complicated for this approach. The process (and furthermore, all the details of real implementations) are too obscure and computationally intensive for the average educated human to follow by hand, unless that person was educated as a cryptographer or a security researcher. Now, most people never know if the hardware or software implements the algorithm correctly and few can diagnose its vulnerabilities.

Most computer users suffer few consequences from lacking an understanding of the workings of the computer programs they use. Encryption programs, like other security processes, seem to ask for more attention though. First, users may be lulled into thinking they have a higher level of security when what they really have are false visual indicators of increased security [29]. If the pro-

[4] This discounts the interests of marketers who may be interested developing profiles of individuals from web traffic. This is outside the scope of this chapter, however, as I am mainly concerned about e-mail confidentiality rather than web privacy.

cess was more easily interpretable, perhaps users could diagnose problems and vulnerabilities more easily as well. Second, sometimes these systems require user decisions; for example, OpenPGP and S/MIME require users to trust an assignment between a person's public key and their e-mail address. Little support is available in the case of new identity attacks, where an attacker poses as an acquaintance and convinces victims that a new e-mail address and key should replace the ones they already know [38].

This availability of software but the lack of adoption seems to beg the question: why are people uninterested? Having software features available and having features implemented in software do not translate into actual use of these features [70]. As mentioned earlier, many mail clients provide, as a default, support for the S/MIME protocol, yet few people have the keys necessary to use S/MIME encryption of messages. Similarly, many add-ons and plug-ins implement the OpenPGP protocol, yet few people install the software (and, similar to S/MIME, few people generate the keys necessary for encrypting messages). The ability to encrypt messages is widely-supported in e-mail clients. The will to use these features, though, seems missing.

3.3 Background: Risk of Eavesdropping

E-mail messages between sender and recipient have far more vulnerability than the average user often realizes. The points of vulnerability include all archives of messages: local archives of messages, temporary archives while sending and receiving, and archive sites throughout the Internet. Each of these expose potential breaches of confidentiality for however long they retain the archive unprotected by encryption.

Whenever someone loses a machine or has one stolen, they potentially lose local copies of their data to adversaries. Local archives include those of e-mail clients (like Microsoft Outlook) that copy messages for quicker searching and offline viewing. The e-mail clients generally leave messages unprotected except for the basic login into the operating system. If a user loses their laptop, their only data protection may be the operating system's login as few people take extra precautions such as encrypting their hard drive. (Even if users did encrypt their hard drives, disk encryption might not be enough. [49]) Additionally, unpatched operating systems or software can inadvertently expose laptops to network attacks that can leak any data stored locally, including

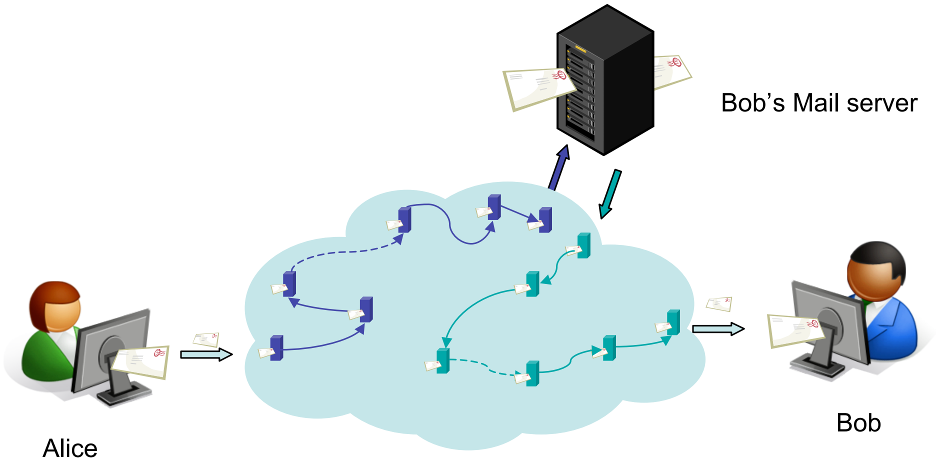


Figure 3.1: Representation of Alice sending e-mail to Bob (via Bob's mail server). From Alice's machine, she connects to the Internet and her outgoing mail server (not shown). The mail is forward through the Internet, one hop at time between routers, until it reaches Bob's mail server. Bob's mail server retains a copy until Bob connects to the server and downloads the message. The message is sent from his mail server, one hop at a time through Internet routers, until it reaches Bob's machine.

e-mail archives.

In addition to local storage vulnerability, e-mail messages are generally vulnerable while in transit between sender and receiver. As the testimony of Mark Klein illustrates, Internet Service Providers can wiretap Internet data lines and provide government authorities copies of all unencrypted traffic that passes through networks they control. As the case of Councilman shows, owners of mail servers and also attackers of mail servers may siphon copies of messages while the mail servers hold copies of messages.

Mail servers are drop boxes for delivering and receiving e-mail messages (see Figure 3.1), and they store messages in their original text or bytes, without protection beyond basic password authentication for access control. Each arrow shown in Figure 3.1 represents one hop between computers, whether they are user machines, mail servers, or Internet routers. In each hop, the owner of the connection can eavesdrop on unencrypted communications, as in the AT&T case. Each computer can also retain a copy of the message, like the mail servers in the Councilman case, but so can the user machines and also the Internet routers.

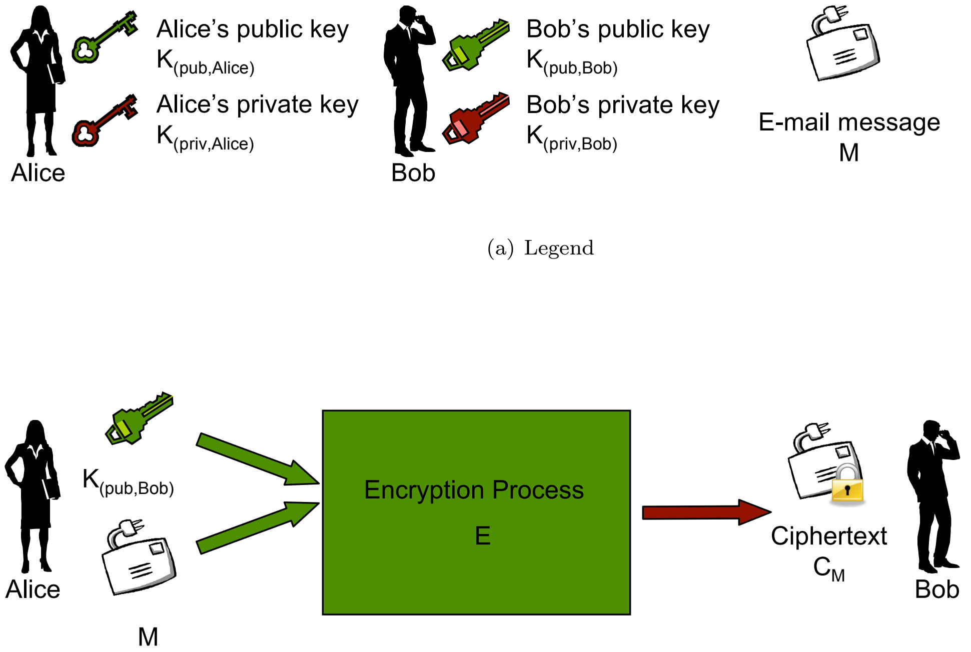
As part of the normal protocol, without malicious purposes, each of these machines may keep a copy of the message. Mail clients, like Microsoft Outlook, fetch messages from mail servers and retain a local copy. Mail servers hold received messages until mail clients pick up or delete the received messages, depending on the protocol. Mail servers also retain a copy of sent messages, sometimes for days, while waiting for confirmation the message was received by the recipient's mail server. As the Internet is a network of servers and routers, all machines that carry one "hop" of the transport generally keep a copy of the data, at least in memory if not on hard disk, for a short term.

Although the above example describes mail servers that temporarily store messages for e-mail clients, the same principles apply to webmail services like Google's GMail, but the location of the message archive changes. Users must always connect to the web service through their web browser to access their messages. Instead of mail servers delivering messages to local machines via mail client connections, the web-mail provider presents a website interface for the entire process. The provider maintains an archive of e-mail messages on their own data storage servers, and the mail servers deliver to the provider rather than the user. In these webmail services, the provider (and not the user) is the sole keeper of the user's message archive.

Those concerned about the technical possibility of exploits see e-mail encryption as a means of broadly protecting communication traffic. Someone versed in computer security can see that each of the above types of message archives (copies from e-mail clients, mail servers, or webmail services) provide points vulnerable to confiscation, intrusion, and eavesdropping. E-mail encryption is then a means for forcing message confidentiality compliance. When the vulnerabilities exist, focusing on thwarting eavesdroppers seems like the proactive solution.

3.4 Background: E-mail Encryption

While users may be unaware of these vulnerabilities, they should, in theory, easily protect their messages with e-mail encryption. The two well-known protocols, S/MIME and OpenPGP, specify public-key encryption algorithms. Although the details differ between the two protocols, the basics of public-key encryption are similar enough. This section will explain the basics of public-key encryption in terms of the inputs and outputs of each process (see Ferguson and Schneier's book for cryptographic details [34]).



(a) Legend

Figure 3.2: Representation of Alice encrypting an e-mail message for Bob. Process is from left to right. Alice uses Bob's public key to encrypt a message for Bob and sends him the ciphertext output.

3.4.1 Public-Key Encryption

Suppose Alice wants to send a secret e-mail to Bob, but Zeke wants to eavesdrop on the communication. Prior to encrypting messages, Alice and Bob agree on a specific encryption protocol and select an implementation of this algorithm (a software program) for protecting their messages. Alice and Bob may select different encryption programs as long as they use the same encryption protocol (for example, OpenPGP). A public-key algorithm for encryption is shown in Figure 3.2.

The encryption process (E) takes two inputs, an encryption key (K_{pub}) and a message (M) and generates a garbled message, known as ciphertext (C_M). (We use Bob as a subscript to distinguish Alice's keys from Bob's keys.) M, in this case, represents the unprotected body of an e-mail message that Alice wants to send to Bob. C_M represents the encrypted message that Alice will send to Bob. K_{pub} represents Bob's encryption key, known as Bob's *public key.*

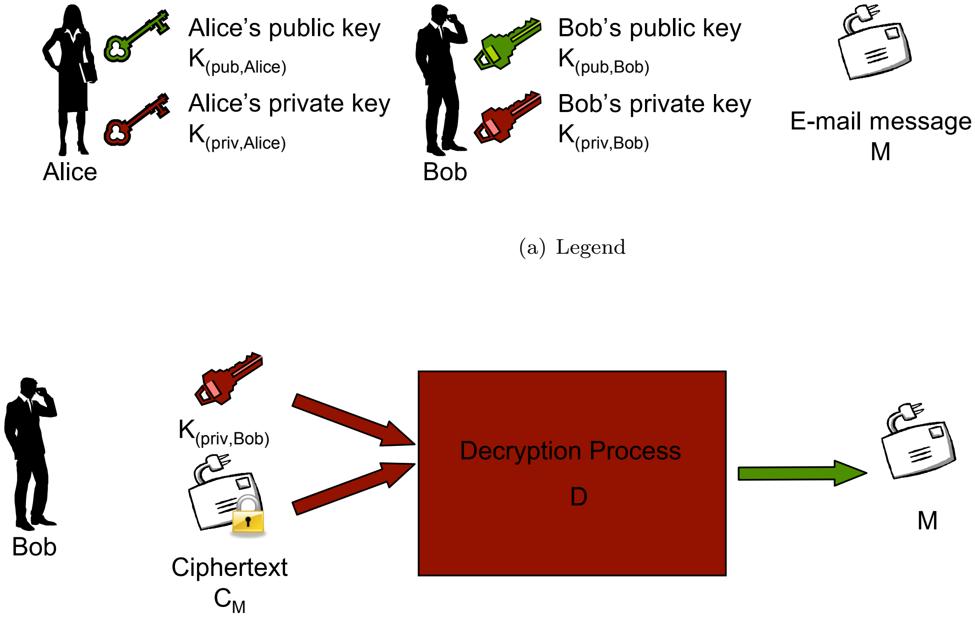
How does Bob get a public key? When Alice and Bob agree to an encryption algorithm, Bob uses his encryption program (or requests a certificate authority, such as the company Verisign) to generate a key pair, with one public (K_{pub}) and one private key (K_{priv}). Once the private key is generated, it must always stay secret, but its counterpart, the public key (K_{pub}), can be disclosed.

When Alice wants to send Bob an encrypted message, she takes a copy of Bob's public key and uses the encryption algorithm to generate a ciphertext version of the message. She can then send Bob a copy of the ciphertext. After Bob receives the ciphertext message, he reverses the encryption process. Software programs that implement the encryption process also implement the decryption process, so Bob uses his encryption program to retrieve Alice's original message. The process for decryption is shown in Figure 3.3.

The decryption process (D) takes two inputs, the receiver's decryption key (K_{priv}) and the ciphertext version of the message (C_M). C_M is a copy of the body of the ciphertext message Bob receives from Alice. The decryption process uses the private key (in this case, Bob's private key, $K_{(priv,Bob)}$) to take the ciphertext C_M and recover the original message M. If Bob keeps his private key secret, he alone has the ability to recover the original message, M, from the ciphertext C_M. An eavesdropper like Zeke may know any or all of the following three items:

1. the encryption process, E, and the decryption process, D

2. a copy of Bob's public key, $K_{(pub,Bob)}$



(a) Legend

Figure 3.3: Representation of Bob decrypting an e-mail message from Alice. Bob takes the received ciphertext and uses his public key to recover Alice's original message.

3. a copy of the ciphertext, C_M

Without a copy of Bob's private key ($K_{(priv,Bob)}$), though, Zeke cannot recover the original message M. (Some caveats exist, but this is currently believed to be the case with sufficiently large keys.)

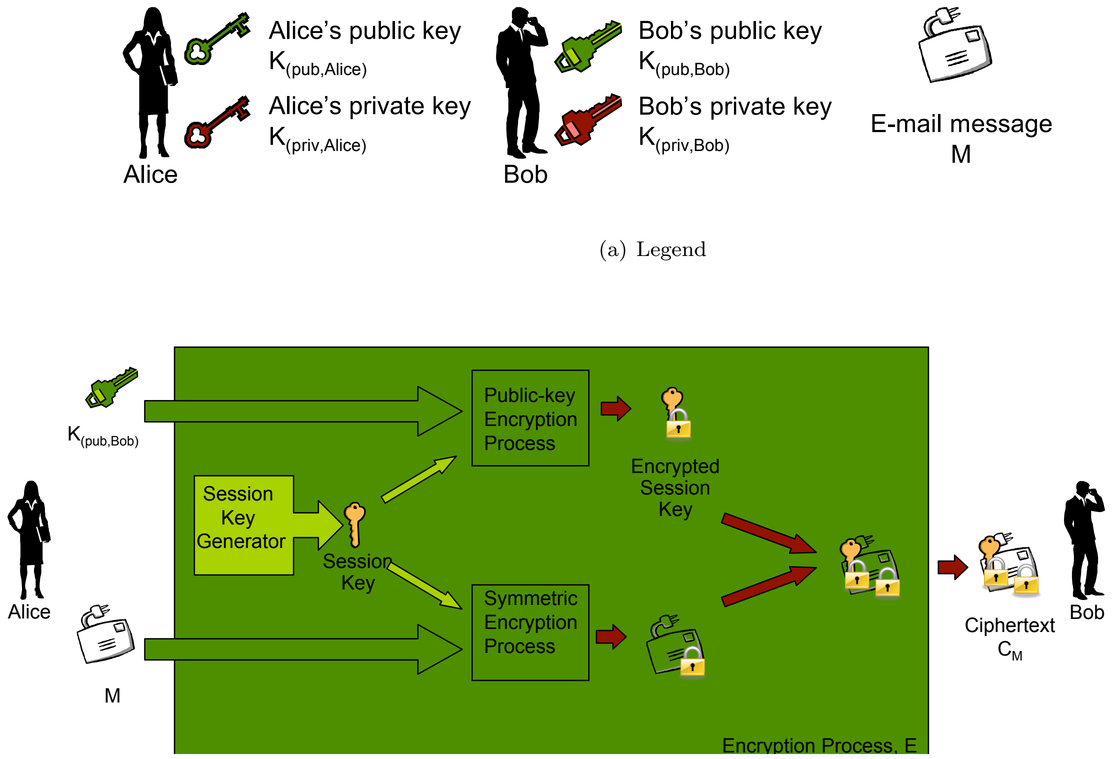
Whenever Bob replies or sends messages to Alice, he does the opposite process. Alice must generate her own key pair ($K_{(pub,Alice)}, K_{(priv,Alice)}$), so that Bob can use Alice's public key and his message as input into an encryption program. After receiving a copy of Alice's encryption key, Bob can generate and send a ciphertext version of his reply to Alice. Alice then uses her private key to recover Bob's reply.

3.4.2 Reducing the Cost of Public-Key Encryption

ElGamal and RSA are examples of public-key encryption algorithms. S/MIME and OpenPGP actually refer to data encryption that uses both public-key encryption and symmetric encryption. Implementations of symmetric key algorithms generally run much faster than public-key encryption algorithms. Unlike public-key encryption which uses two keys, symmetric encryption only uses one key for both encryption and decryption. In symmetric encryption, the key must always stay secret but both Alice and Bob need to know the key. The secrecy of the shared key is problematic for using symmetric key alone because both the sender and receiver must know the shared key (and somehow inform each other of the shared key), but both parties must keep everyone else from discovering the shared key.

So, programs that implement S/MIME or OpenPGP combine symmetric and public-key encryption; see Figure 3.4. These programs first generate a one-time symmetric encryption key (known as a *session key*). The encryption programs use symmetric encryption to encrypt the message with the session key. To protect the session key, the programs finally use public-key encryption to encrypt the session key. Because the session key has much less data than messages do and because symmetric encryption is generally much faster than public-key encryption, the overall process described by either the S/MIME or OpenPGP protocols is much faster than using public-key encryption alone.

When Alice uses programs that implement either the S/MIME or OpenPGP protocol, the



(a) Legend

Figure 3.4: Representation of Alice encrypting a message for Bob with detail on the encryption process, showing use of both public-key and symmetric encryption. Alice's encryption program generates a one-time session key which is used to quickly encrypt her message with symmetric encryption. The slower public-key encryption then uses a smaller input, the key used in symmetric encryption instead of the entire message. The ciphertext that Bob receives includes both the encrypted session key and the encrypted message. He must decrypt the session key before he can recover the original message.

ciphertext sent by Alice actually includes an encrypted version of the session key and an encrypted version of her original message. Bob's program must first decrypt the session key using public-key decryption. After recovering the session key, Bob's program can then use symmetric decryption to quickly recover the original message Alice wanted to send.

Using public-key encryption, Alice and Bob can enforce message confidentiality: only those who have copies of private keys can recover encrypted messages. Another cryptographic process that uses the same keys can enforce message integrity. *Digital signatures* can indicate if an attacker like Zeke has tampered with a message.

3.4.3 Digital Signatures

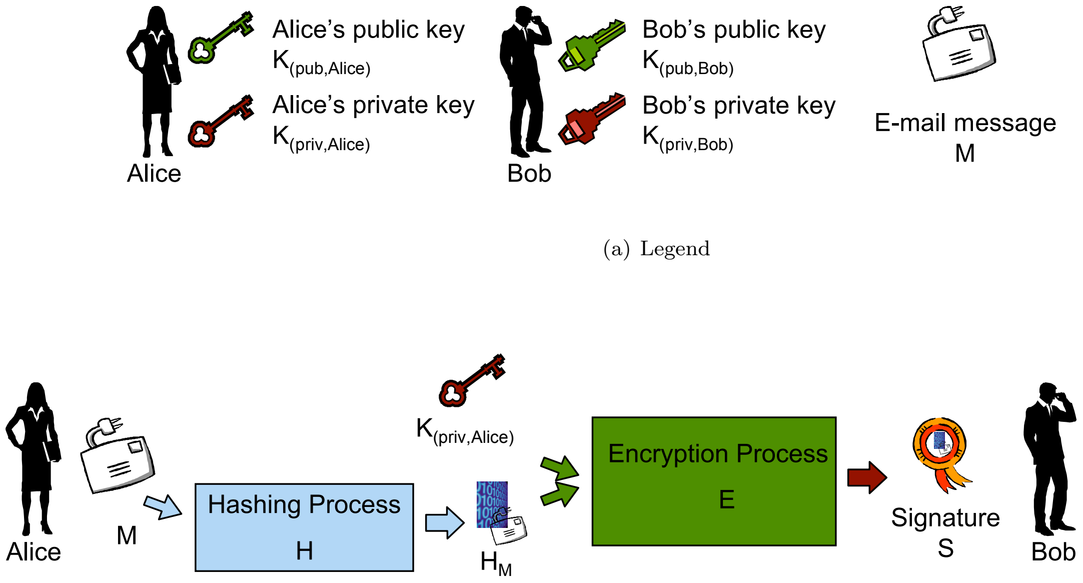
Digitally signing a message is nearly equivalent to encrypting a message with a private key, K_{priv}, see Figure 3.5. If Alice wishes to digitally sign her messages to Bob, she uses her private key to encrypt her message. (Usually, Alice actually encrypts a cryptographic hash of the message H_M rather than the message itself. This step reduces the size of the digital signature.)

Note that the encryption process (Figure 3.2) looks identical to the digital signature process (Figure 3.5); however, the keys are significantly different. When Alice digitally signs a message for Bob, she uses her own private key as input. When Alice encrypts a message for Bob, she uses Bob's public key.

Alice can send her message, M, with a copy of the digital signature, S, to Bob. Bob can verify that her message, M, has not been tampered with by retrieving H_M from S (Figure 3.6).

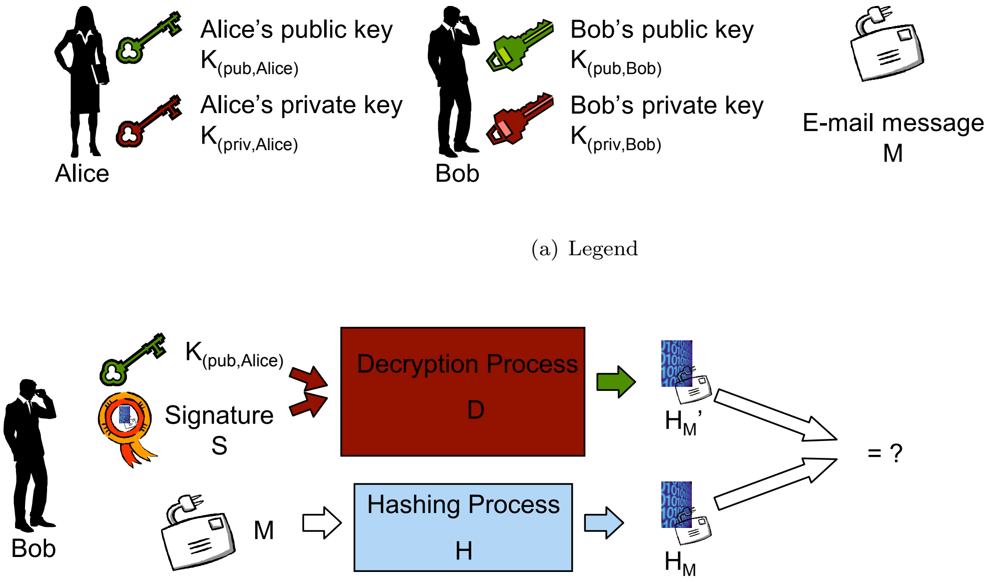
If Bob decrypts the digital signature (S) with Alice's public key ($K_{(pub,Alice)}$), he can determine whether or not the output $H_{M'}$ matches the hashed message H_M. If $H_{M'}$ and H_M match, then M has not been tampered with. Presuming Bob regularly receives both a message M and a digital signature S from Alice, he can always verify the message's integrity.

There is an essential difference between digitally signing messages and encrypting messages. When Alice wants to enforce message confidentiality, she can encrypt messages to Bob, but Bob has to do work (generate a keypair and decrypt received messages). When Alice wants to enforce message integrity, Alice does all of the work (she encrypts her messages with her private key). Bob only needs to do work if he is also concerned about message integrity (decrypting received messages with Alice's public key). Garfinkel has argued that this difference of burden makes digitally signing



(a) Legend

Figure 3.5: Representation of Alice digitally signing a message for Bob. Alice encrypts a hash of her message; instead of using Bob’s public key, she uses her own private key as input. Alice can now send the signature along with her message to prove message integrity.



(a) Legend

Figure 3.6: Representation of Bob verifying the digital signature S sent by Alice using her public key $K_{(pub,Alice)}$ and a copy of the message M. Note that Bob has to re-hash the message M to generate the cryptographic hash H_M.

messages a more adoptable practice [38]. Unfortunately, the two processes enforce very different aspects of security.

Authentication Issues

In truth, key management along with encryption and decryption have practical problems from the scenarios addressed above. One problem is enforcing the secrecy of the private key. Once a copy of the private key is disclosed, an attacker can recover any copies of messages encrypted with the public key. Furthermore, the attacker can also forge messages with a proper digital signature. Currently, most systems use basic password authentication to protect access to the private key, but as we saw in Chapter 2, password security is, at best, a weak method for access control.

Another problem with key management is trying to associate public keys with individuals. Suppose someone claims to be Bob and presents "Bob's" public key. How can you verify this is actually Bob's key? The S/MIME and OpenPGP protocols differ on their approach to this problem. S/MIME relies on third parties, known as CAs (certificate authorities), to link between keys and owners. Usually, customers pay the CA for the service of generating a keypair and vouching for ownership. The CA collects information about the customer to do this. In contrast, OpenPGP relies on a web of trust. Rather than a single authority verifying the association between a public key and an owner, many people can vouch for this association. OpenPGP's trust model incorporates measures of trust. Although there could be many intermediaries, individual users configure their measure of trust that someone's key is correctly associated with that person's e-mail address. Users can learn this directly from the key owner or they can choose to rely on a trusted intermediary who can vouches for that associate. In both the protocols (S/MIME and OpenPGP), breaking this association may not be immediate, which is problematic for confiscated laptops or otherwise compromised keys.

3.5 Research Contributions

Researchers have tried to make e-mail encryption universal and routine, but prior work has tended to look at barriers to use, what would make the encryption process or its startup cost easier for users. Whitten and Tygar ran an experiment and found that only two of eight participants could successfully generate keys, distribute the keys, and use the correct keys for encrypting and decrypting e-mail messages to and from the correct people. This often-cited experiment has influenced subsequent experiments in the general area of HCI-Sec in that they have focused on proving that certain designs were more or less usable by demonstrating rates of success in completing tasks. These papers provide a measurement of success on novel designs since experimental approaches highlight roadblocks to individual users. Generally in an experimental approach, researchers impose structure on participants. They provide motivation for the task, provide the environment for completing the task, and (sometimes) provide training for practicing the task.

This approach can find gains in usability, but it is unclear whether these gains in usability lead to gains in adoptability. In this chapter, I describe a qualitative study of barriers to adopting encrypted e-mail based on interviews with nine employees at an activist group. This work is complementary to experimental approaches. Prior work focuses on the usability of this technology for a broad population, while my work looks at the perceived utility of the technology for a highly-motivated subgroup. I look at their motivations to use encrypted e-mail as well as their feedback on actual use of the technology. This changes the scope of the investigation that prior work has left unexamined. Instead of focusing on average users who may see little risk of eavesdropping, I look at an organization that displays concern about protecting both their image and their data. This leads to looking at, then, why motivated users (and even people with experience using encryption) see universal and routine use of encryption as excessive.

3.6 Data Collection and Methods

The previous examples detailing how to use cryptography to protect communication between Alice and Bob present a theoretical use case. The examples do not reflect any real actors; they explain the process of public-key cryptography but do not explain why it never happens this way.

Few workplaces actually use encryption and few think it is needed; but, asking real users to discuss technology they use and like is far easier than asking them to talk about technology they might not even know exists. I was lucky to have access to a group of employees who had a strong incentive to encrypt. The employees were willing to discuss their use of encrypted e-mail, their technology selection, and their feedback on these technologies.

Secret plans were at the forefront of their work at ActivistCorp,[5] a non-violent, direct action (NVDA) organization, and this was cited as a factor for using encrypted e-mail. In semi-structured interviews, these employees explained their motivation for using (and not using) this technology. Employees at ActivistCorp had opponents working against them, they had secrets to protect, and colleagues' freedom was at stake when security failed. As a result of these interviews, our research contribution is primarily empirical insight into how using encrypted e-mail depends on more than individual perceptions of usability; *individual users also consider their interaction with others in a social context.*

3.6.1 Site Background

As mentioned earlier, I interviewed employees at an activist group, specifically a non-violent, direct action (NVDA) organization, because I believed they had more incentive to protect secrets than typical e-mail users. NVDA organizations have conflict as primary means of affecting change; this is the "direct action" component of the NVDA title. NVDA groups garner publicity for their causes by staging protests in an attempt to shame adversaries into the desired action or to increase awareness for public or government intervention. These groups also undermine their opponents by squatting on or damaging property in order to thwart the progress of their opponents' causes. While the groups are non-violent in terms of physically attacking their opponents, non-violent direction action can include violence; activists knowingly risk incarceration (whether for questioning, for dispersion, or for conviction of illegal actions) and possibly violent receptions during actions. The group I interviewed portrayed this danger and their identity as activists in the following way in an internally circulated document:

> As an organization, we have certain values and tactics that differentiate us from other groups. We are not afraid to stand up to corporations or governments, put our lives on the line or block roads.
>
> Even though all of us here are on the payroll ... try to avoid referring to us as "staff" or "employees" in materials. If someone works for [ActivistCorp], he or she is an activist. If someone is participating in an action, he or she is a supporter or [an] activist (or [a]

[5] The name "ActivistCorp" and the names of all individuals associated with this work are pseudonyms to protect anonymity.

professional for the dangerous ones).

At ActivistCorp, the pervasive image shown to employees portrayed them as activists. Photographs of ActivstCorp's previous direct actions prominently showed employees and volunteers unfurling banners and chaining themselves to sites. These photographs decorated the hallway of the ActivistCorp headquarters. Annual reports, coffee table books, and postcards replicated these images.

Earlier I highlighted some examples of wiretapping and interest in eavesdropping on communications in the US. Most users do not take proactive measures to combat eavesdropping, but some of the activists in this chapter were more wary than average users. Similar groups have had incidents of infiltration and compromised communications [78, 66]. Multiple sources for activist groups have suggested using encryption software and furthermore suggest using the open source version of the encryption software, though this may be a result of cost-savings issues rather than security concerns [67, 77].

As we will show later in the chapter, this risky work motivated some employees to take steps to protect the confidentiality of communications. Their precautions were based on previous compromises to the organization, including infiltration and laptop confiscation. So, while ActivistCorp employees poorly represent average e-mail users, they nonetheless represent users with a need for technology that protects the secrecy of communications.

Within ActivistCorp at the site we visited, Microsoft Windows remained a popular operating system, but many employees used free software to meet their needs. According to the technical staff, Pegasus Mail was the primary e-mail client. Those who needed encryption used Pretty Good Privacy (PGP) with a plug-in called idw's PGP-Frontend for Pegasus Mail. PGP is a program that implements the OpenPGP standard, helping users to generate a public key and private key pair, to encrypt and decrypt messages, and to attach digital signatures.[6] idw provides a free plug-in that is an interface for using PGP within the Pegasus Mail client.

Although open-source software such as Pegasus Mail can be more difficult to use than commercial software, this was not necessarily a drawback for ActivistCorp in other environments. As a non-profit organization, ActivistCorp technical support tended to avoid software purchases for

[6] Actually, the OpenPGP standard historically derives from earlier versions of PGP, but other programs, such as the open-source Gnu Privacy Guard (GPG) also implement the OpenPGP standard.

pragmatic purposes. Quarterly and annual reports along with taxation filings document organizational expenditures, and one measure of an organization's success is how little an organization spends on administrative costs [21]. Unnecessary software expenditures take money that could otherwise support the group's main causes.

ActivistCorp employees were also highly motivated to protect the organization. Some, although not all, employees participated in direct actions in addition to the work required for their jobs, so working for ActivistCorp was probably also a personal decision rather than a purely professional one. The ideologies of these groups necessarily affect the lives of employees and this effect comes from large events but also the daily rituals that show support for their cause through both organizational purchases and personal behavior. An internal document at ActivistCorp detailed purchasing processes that were consistent with their ideologies. It reminded employees that these procedures avoided undermining the public voice of the organization, assuring they never appear hypocritical. Similarly, employees' personal lifestyles can adhere to similar constraints although observations were only anecdotal. ActivistCorp was more than a means of income. It was probably a personal decision in that those who participate in direct actions can be arrested for their work but it was also a lifestyle decision to be consistent with the group's aims.

Given this work environment, I suspected that ActivistCorp showed potential for widespread adoption of encrypted e-mail. If the employees would believed using encryption would help protect the organization, the employees would try to implement the policy.

3.6.2 Methods

I chose a qualitative approach, using semi-structured interviewing with a base schedule for questions so that we were sure to cover germane topics, but I also had some freedom to explore interesting comments in more depth. The interview schedule used for the work in this chapter is included in Appendix B.

My approach used Burawoy's Extended Case Method for structuring the interview schedule and also guiding the analysis. Prior work in qualitative HCI tends to use Grounded Theory. Grounded Theory refers to qualitative analysis where researchers develop their theories based on their fieldwork, finding regularities within their observations by coding their data and then grouping the codes in a hierarchical fashion. Burawoy explains this approach works well in new areas of study, where few prior assumptions can structure the approach to the field and also the analysis. In contrast to Grounded Theory, with his Extended Case Method, Burawoy encouraged researchers to explicitly describe their assumptions and use these to guide observations: "rather than theory emerging from the field, what is interesting in the field emerges from our theory"[19]. With this method, researchers refine and evolve existing assumptions.

Prior work has indicated three sources detracting from better security practices: 1) difficult user interfaces or extraneous procedures; 2) unreasonable expectations on users based on the security desired by technical staff; and 3) underestimated risk assessment by individual users. Focusing on usability derives from Whitten and Tygar's laboratory evaluation of e-mail encryption, mentioned earlier, and formed the basis of similar works that presumed that fixing user interfaces would make them more attractive for adoption [38, 37]. Schneier has suggested people probably avoid security technology because of the inconvenience to their work and because of inexperience with available technologies [85]. Usability problems were compounded with unfair expectations of users. Adams and Sasse showed password authentication systems and protocols could change to reduce burdens on users. The work demonstrated how poor system architectures and security protocols burden users too far, and users would naturally undermine security procedures under excessive constraints. Similarly, Beznosov et al. highlighted an underlying conflict between the goals of users and the goals of support staff (or security teams) within an organization [11]. Furthermore, excessive precautions could be perceived as paranoid behavior [95]. Finally, users tend to underestimate the likelihood of being targeted for attacks by an adversary. Users would probably see themselves as obscure members of a larger group, like the people interviewed by Weirich and Sasse [95, 96].

Each of these detractions seemed likely influences on adopting encrypted e-mail at ActivistCorp. Software installation and key management are known start-up costs (choosing encryption after setup can just involve ticking a box, and decryption can also be automated). I also wondered if employees would resent a requirement that impedes their work, even though it helped protect the organization. Handling software had little resemblence to the core work of ActivistCorp. The technical support staff already said they had little opportunity to educate users. They also lacked the resources to implement encrypted e-mail for universal use. Finally, while the group might see their actions as targeted, it was unclear at the time that they had evidence of eavesdropping in their e-mail communications. Users might not have felt a threat that messages could be disclosed.

Department	Number of Participants Interviewed	Recorded?
Technical Support	2	
Campaigns	3	√
Media	1	√
Legal	1	√
Human Resources	1	√
Finance	1	√

Table 3.1: Participants Interviewed at Activist Corp with Job Descriptions and Indication if Interview was Recorded

I spent two days visiting ActivistCorp's national office, talking with nine employees from six departments, outlined in Table 3.1. Discussions with two employees from technical support were unstructured. The interviews with the technical staff differed from the others in that the questions were about technical support rather than personal use of encryption. Interviews ranged from ten minutes to an hour and a half. If the interview was recorded, I reviewed the tapes and transcribed the items shown in quotes. Because I originally requested twenty minutes per interview, I asked permission before continuing to interview for longer than the agreed period. I also asked for verbal consent before recording the interviews; all of the employees consented although the free-form discussions with technical support staff were not recorded.

As mentioned previously, some ActivistCorp employees participate in direct actions in addition to their daily jobs. Of the nine employees I interviewed, two had participated in a demonstration the day of the interview, a lawyer and a campaigns worker. Both men from technical support had participated in actions in previous years. One of ActivistCorp's publications features the finance employee in a protest. I assumed one of the men from campaigns regularly participated because he mentioned how he used encryption when he organized direct actions in his spare time. These employees demonstrate that even with jobs outside the actions department, they could still become involved with planning or participating in the riskiest part of the group's work.

3.7 Practices of Adopters and Non-Adopters

The following excerpts highlight a few perspectives from the nine interviewed employees. Each of the four vignettes describes the employee's job and their use of encrypted e-mail in the workplace.

3.7.1 The Habitual User: Cautious or Paranoid?

Woodward worked for the campaigns[7] department as a researcher. While interviewing Woodward, his officemate Stefan (a research manager) periodically interjected comments.

As I walked up to Woodward's desk, I heard a whirring sound and realized he was shredding a document; it was an unusual start to a conversation asking him about his security precautions. Such measures are actually unremarkable given his job. Woodward looks for incriminating evidence against ActivistCorp's opponents. Not all of this information is easily obtained; Woodward might use atypical channels to retrieve it. For example, after recalling the last time he encrypted an e-mail, he described why he used encryption in that message:

> That's not public information. And we got that information through ways—and it's not information that's publicly available and it's known that it is not publicly available, also.

He felt encryption was necessary, in part, because he worried that he might inadvertently reveal his source. With a job so focused on gathering information, he was naturally protective of his own information flow:

> I know how people get information from companies and corporations. That's part of what we do.

He also cited concern over who might be watching:

> I'm aware of the level of government surveillance that's happening. Most people aren't aware of the FBI's Carnivore program, you know.

He added, "there are a lot of people that are interested in what we are doing." Not only did he encrypt messages, but he also encrypted part of his hard drive:

> 'Cause otherwise, if I lost my laptop, you know, then someone would just be able to read everything on it. And so then, what would be the point of—what would be the point of encrypting anything, you know, if—if everything I have is—someone could just read it on my laptop?

[7] Campaigns are "a connected series of operations designed to bring about a particular result" (Merriam-Webster) rather than simply political bodies. Actions or direct-actions are the high-profile events staged by activists. The actions are grouped by goals known as campaigns.

In addition to his cautious approach to storing data, Woodward distrusted plug-ins for e-mail programs, relying on encrypting the text of a message first and copying it into his e-mail client later. He feared a plug-in that simplified the process might be a Trojan horse or an improperly implemented program. Woodward was careful to encrypt a message, so he was also careful to avoid compromising his message through a software hole. He said his vigilance was related to his background in understanding the weaknesses of technology. He encrypted messages whenever he used wireless "and that's just because wireless Internet is so inherently insecure." He knew because he had sniffed traffic in wireless hotspots himself "just for fun."

Woodward's concern may or may not be justified. As discussed in Section 3.6.1, activist groups have had problems with infiltration and monitoring. Woodward himself cited instances where ActivistCorp's offices were infiltrated or raided, indicating that the possibility of espionage made him vigilant.

Woodward was even cautious enough to be concerned about interviewers who claimed to be students:

> It was about a year ago. But, yeah, it was someone asking us a lot about our internal workings and such. And they were being very—...they weren't that great at their story. They were very vague, you know. I'm like, "Who's your professor?"

Although he was cautious, Woodward admitted that he might use encryption more often then needed:

> Oh, sure. I'm sure there's many times where it's not absolutely necessary, but I'd rather err on the side of caution.

He admitted he might use encryption excessively, but he still limited his use. Woodward said he did not encrypt public information. If it was public, he would send the information in plain text.

Woodward was surprisingly more cautious than what I initially expected from employees at ActivistCorp. One of the hosts said his vigilance was extraordinary relative to other employees. Yet, while Woodward was vigilant about protecting secret information, he still limited his use of encryption – his use of encryption did not extend to ubiquitous and universal use.

3.7.2 A Middle Ground

Woodward may have displayed extreme vigilance with information because campaigns and actions provided the visible accomplishments of ActivistCorp. Woodward interacted with confidential aspects of the organization, but a large portion of the employees keep the organization afloat with extensive administration, including departments for human resources, development (fundraising), finance, legal, technical support, and media. Unlike Woodward, their work was less entrenched in the organization's secrets except when they participated in direct actions (if they did at all).

These two vignettes are from administrative employees describing how they have used encryption in ActivistCorp. Less involved with campaigns and actions, they could be more practical about taking security precautions.

The Self-Sender: "Don't Go Overboard"

Abe worked in development, helping ActivistCorp's fundraising efforts. Because he handled financial data, Abe used encryption frequently, particularly when he received records from online donations ("I tend to try and be sure I PGP everything that has a credit card number on it.") He also communicated with an external vendor for recruitment, and the two of them sent each other encrypted copies of the financial records as e-mail messages while they synchronized their records. Abe found this setup simple, but he mentioned having difficulties convincing others that using encryption was easy. He also thought some people in ActivistCorp needed to be more vigilant. He described how he tried to convince the head of campaigns in his home country to use encryption:

> Why? Because it was just good. If the ... police ever come and bust into the office, you shouldn't have a document saying "Hey, I'm discussing how I'm going to campaign against [a controversial issue]." It's not the kind of information you want them to have.

Despite his argument, his colleagues were uncooperative, "most people see this as more work and want things simpler."

Abe saw himself and believed others saw him as someone comfortable with technology:

> I use computers a lot at [ActivistCorp]. I'm—I'm actually considered a "techie"—that's what other people say, "Oh, you're being a techie."

Abe's attraction to adopting encrypted e-mail was related to his love of technology and the excitement or importance implied by using it:

> [When] it wasn't forced upon me, I was willing to try it. For boys at least, there's a "gadget factor" because [I was told] it would take five years for the CIA to decrypt it, so you felt a bit like a secret agent.

He qualified his interest, however:

> I probably don't know how to use it in the best possible way, but I know how to encrypt and un-encrypt. I don't explore, uh, the more complicated things.

The "more complicated things" included digitally signing messages, even though the process of digitally signing has fewer steps than encrypting messages. He explains:

> I figure I'm a hundred times better than most people if I've encrypted. Don't go overboard.

He estimated that encrypting every e-mail message would add another hour to his workday unless it was automated. He said encrypting is like healthy eating and exercise as you admire those who do it because you know it's the right thing to do, but you do not actually do it yourself. He just wanted to be responsible; he would encrypt some things but not everything. Fear of attackers was less important than ease of use. If it was easier to encrypt everything, he would. He likened this to backing up data; there was "no fun factor" and encryption was a chore, "like housework."

Abe was another example of a user who was aware of the secrets he accessed within the organization. He was also technically savvy, although perhaps not as absorbed as Woodward. Unlike Woodward, however, Abe saw the technology as difficult to set up for ordinary users and he saw encrypting messages regularly as a time consuming process.

An Ephemeral User

Jenny worked as a liaison between campaigns and the other departments at ActivistCorp, primarily helping people within the organization. She used encrypted e-mail over two years ago:

> Um, I've used it before. I used PGP—I don't know if that's a certain kind, or that's what we call it here. I used it...before doing some sort of action. We did a whole bunch of direct action and we had, I guess, two of those nationwide ... we were sending encrypted e-mails back and forth ... so, like, leading up into that so people weren't reading what we were doing—or would know when we were going to do it?

Jenny ended with a question, and she might have tried to distance herself from encryption experts. She began the interview by saying, "I hope I can be of help to you because I don't really use it that often." She also refrained from speculating about what encryption does or how people could intercept e-mail communications:

> I have no idea how it works. I guess people can hack into our system, but I have no idea, like all that kind of IT stuff.

She also said it had been so long since she used PGP that it was unlikely she could use it again:

> I use it very [pause] like I did, maybe sent two e-mails, so I didn't use it very much at all. And it's [pause] it was on my computer, but I haven't used it in so long that I probably don't remember how to use it. I'd probably have to get, like, a refresher course.

Although uncomfortable with the technology, Jenny saw encryption as helping to protect the confidentiality of messages. I asked her to speculate on why she had to use PGP two years ago:

> Well, I think the reason we use it is so that we can actually perform the action that we want to do, so that it's not like we get stopped before we've actually been able to, like, you know, put up our banners or things like that.

When I tried to get her to discuss other circumstances where she could use encryption, she did not understand. In fact, Jenny had trouble understanding why we were looking at encouraging people to use encrypted e-mail more often:

> I have a question for you ... why would people need to encrypt their e-mails, like more? Everyone, like corporations and stuff. And why would they ... why would more people want to encrypt their e-mails?

While Jenny was involved with ActivistCorp's visible action work, she rarely needed to use encryption and she only used it for short durations. She was open to receiving help on using encryption, but she was uncomfortable portraying herself as an expert. She was also unable to see why it could be used more generally than just protecting secrets.

3.7.3 The Uninitiated User: Without a Secret

Sandra was a co-author of a writing manual circulated within the organization. This manual advised people on how to present ActivistCorp to the public. What was intriguing about this manual was the following statement:

> Some good advice: if you don't want something you put in an e-mail to get into the wrong hands, don't write it in the first place.

She explained in the interview that was that she was concerned about accidentally sending messages to the wrong people:

> The reason I included that sentence was just, um, to make sure that people are careful because e-mails can be really dangerous. And I know I, in the past, have clicked "Reply all" instead of "Reply" or "Forward" when I thought I was replying. [My advice was] just to keep people from getting into trouble.
>
> I've sent e-mails to people that I thought, "Oh, maybe I shouldn't have sent that e-mail" but that's just 'cause I was angry. [laughs] You know? [I've never been] afraid someone would read it that shouldn't.

She limited her concern to warning against sending something personally offensive. She felt that encryption was unnecessary for her day-to-day work:

> I don't think any of my communication is anything people are dying to get their hands on. I don't—I am not involved in any of the ... protests or that sort of situation we do. So, there's not as much need for, like, me in the organization to use that kind of thing.

Sandra believed what she wrote was uninteresting to eavesdroppers. She, like the others interviewed, saw encryption as a method of protecting secret information. Without secrets, it is silly to assume she would encrypt messages.

Sandra believed she had a low-profile role at ActivistCorp; she warned others about using e-mail inattentively, but her advice was about errors rather than secrecy. She thought it was unlikely that governments or opponents would be interested in observing her communications. Making the effort to conceal messages seemed irrational to her.

3.8 Adoption Criteria

Having introduced a few of the employees at ActivistCorp, we now have their perspective for understanding some of the social context of adopting encrypted e-mail.

3.8.1 Secrets

As mentioned before, employees made the distinction between paranoid use of encryption for all communication and justified use of encryption for protecting secret data. Organizational secrets required higher levels of secrecy (as Woodward explained, "We want to be able to control the release of ... information"). Woodward's officemate, Stefan, illustrated why secrecy was important to ActivistCorp's internal plans:

> You don't want to show your cards. You don't want that stuff out because people's lives are in jeopardy—*really.* I mean, people are taking an action and could be arrested, could be, you know, jeopardized in some way.

ActivistCorp wants to surprise an opponent when direct actions start. This prevents police from blocking their desired location and prevents their opponents from dissolving the conflict before it begins. Thus, information about when an action starts is secret. Once a protest starts, the event must seem tireless with an illusion of boundless resources for the operation. Here, Stefan explained why information about the end of an action required secrecy and why employees protected these plans:

> It's like if you had a strike against a company but you announce that we're gonna give in on May 7th whether or not we've won yet. Then the company would just say, "Well, fuck, we'll just wait ...." What we're doing is holding a protest and we want [pause] the—whoever we're opposing to think that there's no end to this protest until we give in...so if we accidentally announce that ... we're on to the next campaign on May 7th, then it doesn't really help build pressure.

Maintaining ActivistCorp's identity as having inexhaustible power bolsters the group's influence against their opposition. For the same reason, Stefan explained how the group needed to prevent disclosures of any weaknesses:

> We're like a corporation, so if Nike ... said "Damn, Adidas really has us on the running shoe market" and that was printed in the paper, it would crash their stock prices. Things like that. We're in the same game. We're sort of competing for power and—and the illusion of [ActivistCorp's] power is really important, as important as our real power, like corporations and politicians fear what they think we can do.

Both Stefan and Woodward see encryption as a method of keeping discussions about resource limitations and internal criticisms private within the organization. Secrets not only concealed power but also weaknesses, so that adversaries could only guess how large of an arsenal ActivistCorp had built. Stefan and Woodward's comments reflect a desire to control public information about ActivistCorp's private discussions. From a technical standpoint, universal and routine use of encryption would help this aim. Phil Zimmerman, the creator of PGP, saw that universal and routine use of encrypted communications would keep opponents from gleaning information that would otherwise remain private. If people only used encryption when they need it, anyone who discovered people communicating with an encrypted channel would know these conversations must be secret. Encryption that not only concealed secrets but also public information (a technique known as traffic obfuscation) would prevent adversaries from guessing that groups were building an arsenal or planning an event [105]. Traffic obfuscation keeps adversaries for discovering the timing and location of secret events–otherwise, a burst of encrypted communication to a certain location indicates something is about to happen over there.

Stefan and Woodward both felt using encryption was a necessary component of their jobs. Yet, as stated earlier, Woodward differentiated between the necessity of encrypting secret data but the superfluity of encrypting public data. Furthermore, while Stefan and Woodward both worked in campaigns, they also provided examples from direct actions to explain how ActivistCorp needed encryption. They believed themselves to exclusively use encryption within ActivistCorp, guessing they were the only ones who needed it. Most other employees agreed, believing that only the actions department or only the actions and campaigns departments needed encryption.

Yet, passing financial information also required some secrecy and Abe in development admitted to using the technology for protecting banking information about donors:

> We have our supporters out there—our supporters are giving us ... donations. So, they're doing it from ... the most sensitive place they can. There's no commercial reward in it for them.
>
> There's a—there's more of a personal relationship than if you were with Amazon [the online bookstore] or something like that. So then, in that respect, it would be far more damaging for us if something was to happen to those donor's records....You know, we rely on our supporters, they keep our organization running, and whereas [a big online store] could probably buy its way out of it, we're in big trouble. Our supporters would say, [sighs] "you know what? You guys are not responsible—we can't trust you with our—our credit cards. Now I'm not gonna give you any more money." And that means we're finished.

Abe contrasted ActivistCorp's encryption to an incident with Bank of America's lost tapes of credit card holder's banking information in February of 2005 [50]. ActivistCorp needed every donor's contribution, and while big merchants may suffer embarassment or financial lost, big merchants can more afford to lose a few customers. Abe encrypted donation data to avoid disclosure of donor identities. First, some donors might avoid being publicly linked to ActivistCorp, but, secondly, some people might fear the consequences of such a link:

> There are certain people who we meet on the streets who are like, "No, no. I work for the government. I can't talk to you." And I imagine that [some people] might really want to support us, but they assume that by supporting us, they put a mark

> on themselves as being *bad*. So, if they do, are brave enough to support us, I think, in their minds, they are pretty sure that we are going to protect their confidentiality, you know.
>
> I mean, it's a common thing you hear from people. "Oh, I'd love to support you but I work for [Company X]." So what? You work for [industry Y]. What's the connection? We're not going to put your name on a billboard. But, you know, people do this sort of instantaneous thing that—and whether this might be a great excuse for why they're not going to give us money, but—many people cite that "I can't support you because I'm in an industry (or the government or the military) and this is why I can't support you." So, I do think people in their heads say, eh, if I give money to [ActivistCorp], I'll get into trouble.

Abe wondered if his desire to protect donor confidentiality was justified as he speculated donor identities could be snooping targets:

> Who knows? In a post-9/11 world, [ActivistCorp] doesn't have a lot of friends on the other side, so's to speak. The other side being the administration, industry ... the Homeland Security, you know. We're not, uh, on the same side of the game anymore. We're definitely opposed to what they do. And I think they view us far more as an enemy today than before September 11th. So there's a definite suspicion there. Maybe we're over-inflating our importance, maybe we've got a big ego, but we'd like to believe that they would be very interested to [pause] run through our database and see exactly who supports us. They would be very interested to know who our supporters are. And so, we're obliged to protect it in every possible way.

Many organizations have secrets, and ActivistCorp resembles any other ambitious corporation in this respect. ActivistCorp needs to prevent competitors from knowing about banking information and internal developments. Their media department kept donors, the private industry's equivalent of investors, updated on recent accomplishments, setbacks, budgets, and forecasts. Corporate secrets, which may have just as much financial and survival impact on the organization, can also correspond to the secrets maintained by ActivistCorp's employees in competition for power and influence. The unusual aspect of this group's situation comes from their use of encryption to enforce confidentiality and, furthermore, from members who differentiated between when they used encryption and when they do not. Previous sections have already described limits to using encrypted e-mail, but the next section will show some further restrictions on use.

3.8.2 Paranoia

Some employees interviewed at ActivistCorp limited to their willingness to implement secure practices; moreover, they saw moving beyond that limit as abnormal or paranoid. While Woodward was especially vigilant, even the technical support staff admitted he might be excessively protective. Was the effort justified? Was it reasonable precaution?

Abe explained how someone could "go overboard" when he described how a representative of the PGP Corporation visited ActivistCorp. Instead of a typical password authentication, the representative took off his necklace and used a removable flash drive that held his private key. The demonstration discouraged Abe:

> It was too over-the-top and definitely too complicated. It was like a movie.

He saw the presenter as paranoid. Abe admired this man somewhat, saying carrying a private key around the neck "is really impressive. I take my hats off to that dedication." He also said:

> Yeah, I admire him because he comes in and puts his passphrase [bumps on the table] every single day, three times a day, so that's very dedicated to his stuff. He must either be very scared or very motivated.

He was not sure whether this vigilance was justified. In fact, he associated it with being fearful, perhaps irrationally fearful. Abe reiterated this when speculating on why a colleague sent every e-mail message encrypted. He figured this man has an automated system for encrypting e-mail "or he's nuts."

Someone like Abe, who handled financial data and still was "sure to PGP everything that had a credit card number on it," still acted more cautiously than someone with a more administrative role, like Sandra. Sandra mentioned previously that her e-mail communications were not anything people were "dying to get their hands on," and she explained:

> I'm not paranoid enough to think the CIA is monitoring my e-mails or anything to that effect.

Not only was encrypting messages excessive for someone who had no secrets, it was *paranoid behavior* to assume anyone would be interested in eavesdropping on her communications.

Jenny also found it was abnormal to encrypt non-secret information. When I abstractly explained that people in security suggest all users encrypt all messages, Jenny was baffled:

> So you're saying that ... people should just—even *normal* people? That ... you're sending e-mail to ... your mom, like "Hey, things are going—" That you should encrypt your e-mail. That people should do all that.

Jenny emphasizes "normal people." *Normal* people wouldn't encrypt normal messages.

3.8.3 Flagging

Jenny's quote about "normal people" highlights another association with encryption. Encryption is a flag of message importance or secrecy. Once flagged, people will try to maintain that level of secrecy. For example, an ActivistCorp lawyer said when a message had been sent to her in encrypted form, she would always reply with an encrypted message. Jenny agreed to this as well. Jenny delegated the decision to use encryption to the head of the action, some manager. When she received encrypted messages, she just needed to make sure she could maintain the same level of secrecy someone else declared. In this case, she needs to be able to reply to encrypted messages with encrypted responses.

If fact Sandra also agreed that she would maintain a level of secrecy if ActivistCorp set a policy for it:

> Well, I mean, if it was a policy of the organization, then I would totally do it—and I don't know what it entails to encrypt something, if it was just ticking a box on my e-mail program, then I'd do it just for the added security, but I dunno even how to do it at this point. I don't know if my computer program offers that or anything about it.

Both Jenny and an ActivistCorp lawyer reiterated this, although both had caveats on the necessity of such a policy and whether it would be practical. While considering the practicality of policies, employees are likely to comply with security precautions against reasonable concerns. Conforming to preset policies differed from understanding why such a policy would be instated. For Jenny, universal and routine use was incomprehensible:

> I just don't see people going, "oh, yeah, I should take the extra step to encrypt my e-mail." It's not a hard extra step, but I don't understand why. Like, I can see people saying ... I should protect this against a virus or something like that. But encryption, I just—it doesn't—I don't think people would see that as a—a bonus, like something that they'd really wanna [do]—does that make sense?

Some encrypted messages violated this expectation of secrecy. Abe talked about someone who sent unimportant but encrypted messages. There was a time cost to decrypting received messages, so forcing the recipient to decrypt was considered rude:

> I work with somebody ... and he sends *every—single—message* of his is encrypted. Even if it is just saying to you, "hey, can we have a meeting tomorrow at 2:00?"—it's encrypted. Why? I think he probably has some automated system. That everything he sends gets encrypted automatically. I can't believe he's encrypting manually every time. But to me, it's like—OK, if it's automated—fine. But, it's a bit irritating, you know. I get this message and—oooh, it's encrypted. "Can we have a meeting tomorrow at 2:00?" I'm like, what's the secret?
>
> You got to justify it. I mean—unless it—if it was all happening automatically—great. If it was encrypted on his computer and he sent to my computer, automatically encrypted or decrypted it—fine. Then, encrypt everything you want. But if he's just writing to me something, why put the extra workload on me of tapping in my passphrase and opening it up separately and so forth?

Encryption was a flag that signaled a message was important. If the message was mundane, it was annoying to get over-excited or to spend the time to decrypt the message. Part of the problem lies with the work required, the user intervention that key generation, encryption, and decryption currently require. Protecting the secret keys means that users must continuously re-authenticate themselves to the encryption programs to verify the users are the proper owners. The problem with automating or caching passwords for these systems is that it weakens the trust on ownership of the private keys. The keys might be used to force digital signing of messages or be used to decrypt messages from stolen laptops. It opens possibilities for vulnerabilities that the current structure tries to avoid [36].

3.8.4 Key Management

The whole problem of automating encrypted e-mail really lies in a key management problem as setting the keys and distributing them are barriers to adopting encryption. Schneier writes that key management is "without a doubt, the most difficult issue in cryptographic systems" [34]. Tied to the key management problem is key revocation and regeneration–how do you know if someone compromised the key or if it should be accepted anymore?

From a social context, we expected overhead and intimidation prevented adoption. Abe described it as "the average person doesn't think they can set up encryption." Jenny dismissed the usability problem though:

> No, it was ... I mean, I don't remember having any problems using it ... it wasn't hard. [pause] It was just something—like, you had to ... find the right person, you know ... it wasn't like you could just send an e-mail, you had to definitely ... [pause] find the right key? In order to ... send them something. But, I didn't think it was that hard. It was easy to use, you just—you just had to learn how to use it, I guess you could say, if you remember your password, which I don't remember right now [laughs]. Like it's a different password from all the other stuff. But if I used it more often, I don't think it would be that hard.

Abe believed that encrypting messages is "really simple if it is set up for you" and having technical support staff set up the process would increase use. The technical support staff helped Woodward start using encryption:

> With my job, one day a task arose related to, um, a certain direct action that we were doing. And they were like, "Whoa, well, we can't send you this, you need a PGP key."

> So then I—I walked down to the IT department and said, "guys, I need this PGP thing." They're like, "OK" and they set me up with everything I needed to know.

He gives more details on how this was initiated by someone else:

> Once, they were like, "I need to give you some information but I can't." You know? "Go find IT and get yourself a PGP key and send me your public key and then I can give it to you."

Abe described this situation:

> [PGP is] almost, like, viral: I have got it and I want to use it and you start using it.

Considering these examples, starting to use encryption implies delegating authority [31]: someone else determines encryption is necessary and someone else sets up encryption. As long as someone else does the setup, using encryption is easy.

3.8.5 Security Models

The technical support staff brought up the ideal of universal, routine encryption of e-mail, but they saw current PGP system setups as an impractical burden for a hundred ActivistCorp users. Unlike the support staff, none of the other employees interviewed (not even Woodward), mentioned this argument in their interview. The general users had another model entirely: they were willing to support a policy of encrypting secrets, whether it was information related to an action or to financial data, but it was a huge cognitive leap to go from protecting secrets in an individual message to obfuscating secrets using everyone else's messages.

Encryption was equated with stopping opponents from discovering secrets, as Stefan elaborated:

> I think the only people who do encryption in the organization are people who have been trained to be—who are associated with—specifically the [direct actions] we do. So we keep that stuff encrypted. And that's [pause] I think probably anybody outside of [ActivistCorp] would assume that's what we do.

Woodward also emphasizes encryption is for people who manage secrets:

> It's more if someone is in a role that requires them to be quiet about something.

When the information is not a secret, everyone outside of technical support sees encryption as excessive. Some employees, like Jenny, find encryption has a specific purpose: protecting secrets. Jenny agreed when explaining why she does not use encrypted e-mail more often:

> Um, I'm not really involved in the planning of that kind of stuff [in direct actions]. And I don't know of any other reason I would need to encrypt my e-mails 'cause most of my e-mails are just ... things public. People could learn, I mean, people could read my e-mails, they wouldn't see anything [pause] incriminating? I guess that's the only thing. [Outside of that], I don't understand why anyone would need to [pause] encrypt their e-mails, I guess, in the organization.

Equating encryption with confidentiality might disappear if encryption was invisible to the user. It also might not. In the case of digital signatures, most of the process can happen automatically. Once a user has set up a public and private key pair, an e-mail client can automatically and routinely sign messages. The recipient can check the signature if they want but they can ignore it as well; furthermore, e-mail clients can also automatically check the the signature. The sender (actually, the sender's software) does all the work and the recipient benefits if they understand and observe the digital signatures.

I expected employees would value message integrity comparably to message confidentiality and would value the utility of the cryptographic methods for demonstrating integrity. Although I have not explored the topic in depth, digital signatures seemed relatively unimportant to the employees I interviewed.

Both Abe and Woodward encrypted e-mail regularly, so they both had public and private key pairs. Neither routinely signed their messages. Woodward saw signing messages as a feature bundled with encryption. He knew that there was a button to encrypt and sign, but he only signed messages that were encrypted. I speculate that since Woodward avoided encryption when sending public information, signing messages was extra effort; he only semi-regularly used the software that signed messages. For Abe, digitally signing required more cryptography than he was comfortable with:

> I probably don't know how to use it in the best possible way, but I know how to encrypt and un-encrypt. I don't explore, uh, the more complicated things.

Even with two technically savvy users, I was unable to see universal, routine use of a technology for demonstrating integrity of messages. The imminent danger was snooping [66]; discussions about catching (or warning about) forgery and tampering never started. Tamper-evident communications, though, could be helpful for the group; for example, Abe could benefit from checking for tampering in his financial reports. Most of the focus in the discussion was about message confidentiality, not message integrity.

3.9 Study Limitations

The study in this chapter relies mostly on interviews of people who worked for a group that needed encryption. Interviewing also limits the results. I have participants' reflections and speculations on using a technology but I did not view actual practices. When someone discussed the secrecy of an operation or the precautions taken, I made assumptions about the accuracy of their perception of threat. When they claimed how easy a process was, I did not cross-verify their claims. I also met with only the participants introduced by my hosts, rather than randomly sampling employees at the organization or similar organizations. My hosts, the technical staff, scheduled meetings. My results are then limitations based on restricted access to participants and based on relying on interviewing alone.

A larger problem is how my results tie to a specific technology and that technology at a specific organization. This constrains the impact of the results in that their selected protocol, OpenPGP, is considered both unusable and obscure [98, 37]. Most software programs for communication support SSL connections for encrypted transport, but few e-mail clients support encryption with an implementation of OpenPGP (although many plug-ins or software add-ons fill this gap). In contrast, popular e-mail clients like Mozilla Thunderbird, Apple Mail, and Microsoft Outlook and Outlook Express do include, by default, implementations of the S/MIME protocol. Support for OpenPGP generally requires a plug-in, such as EnigMail for Thunderbird. More troubling for our area of interest, however, is that e-mail clients themselves are becoming less popular with an increased popularity of web-based e-mail services such as Microsoft's MSN Mail, Yahoo

Mail, Google's GMail, and Facebook's mailbox. None of these services provide features for e-mail encryption (although GMail's corporate services offer e-mail encryption through what appears to be a proprietary protocol).

In this study, however, the group chose this technology for themselves. It was not imposed on them, as in experimental studies. Also, as mentioned earlier, their choice was consistent with the choice advocated by activist handbooks. While the results of the technology choice may not apply more generally to all e-mail users, they reflect the experience of this motivated subset of e-mail users.

3.10 Conclusions

Critics may argue that ActivistCorp had adopted the least usable form of the technology. Had employees adopted implementations like HushMail, CryptoHeaven, or S/MIME support in e-mail clients like Thunderbird or Outlook, perhaps they would have encrypted more frequently or with fewer complaints. Furthermore, as Edwards et al. have suggested, the social meaning associated with using encryption probably derives from its visibility [33]. Automated handling of this process could remove stigmas associated with its use so that people could have increased security without appearing paranoid. I talk about one way of automating encryption for group discussion in the design of EMBLEM (Chapter 4), drawing from the experiences described here and I also discuss some of the pragmatic issues in redesigning encrypted e-mail.

The security community has long recognized the utility of encrypted e-mail: it protects the contents of messages, and universal, routine use obfuscates e-mail traffic. There is also a recognized usability problem: adopting encryption incurs overhead cost and sending encrypted e-mails is less efficient than sending plaintext e-mails. My work has contributed to prior work that qualitatively studies HCI-Sec, specifically illuminating the social setting affecting adoption of encrypted e-mail. Utility and usability influence adoption, but they are not the sole criteria.

The perspective and examples provided by my interviews offer insight into a previously unstudied group. First, ActivistCorp was a rare organization in that secret plans constituted a major component of the organization's mission. Second, many employees personally supported the activist group's causes, and, thus, these employees had more incentive than ordinary industry workers when considering the protection of organization secrets. Lastly, ActivistCorp also granted us access to see inside the organization. These circumstances came together to make the quotes and descriptions from the interviews a portrait of how encrypted e-mail could be used in the workplace. While the findings are tied to the specific technology used by participants, they nonetheless provide insight into the non-technical aspects affecting adoption.

Part II

System Design

Chapter 4

Redesigning Encryption for Group Discussions

In Chapter 3, I presented a study of an activist group whose members occasionally used encrypted e-mail in order to protect the confidentiality of organizational secrets. The employees limited their use of encrypted e-mail; while some might expect the primary issue to be whether or not employees knew how to use the technology, the main question remained whether or not the employees saw the technology as a necessary step for the kind of message they were sending. This contrasts sharply with the universal and routine use of encrypted e-mail advocated by Phil Zimmerman, creator of PGP [105].

Whereas the previous chapter focused on the adoption of technology for protecting message confidentiality by an *individual within a group*, in this chapter I explore designs that consider *group adoption* of a technology for protecting communications about a given project. As mentioned in my introduction, when examining how individuals adopt technology, communication protocols (such as the OpenPGP standard) are subject to the influence of network externalities [58]. Katz and Shapiro discuss this "positive consumption externality" in terms of the adoption of telephones:

> The consumption externalities may be generated through a direct physical effect of the number of purchasers on the quality of the product. The utility that a consumer derives from purchasing a telephone, for example, clearly depends on the number of

> other households or businesses that have joined the telephone network.

The more users a product develops, the greater the benefit an individual user reaps from the product. Network externalities are especially obvious in encryption software: *I can set up encryption software on my computer, but that only means other people can encrypt messages to me.* If no one else uses encryption software, I will never receive an encrypted message and I will never be able to send one.

Katz and Shapiro further state, “the utility that a given user derives from the good depends upon the number of other users who are in the same ‘network’ as is he or she.” Any benefit an individual user receives from adopting encryption software depends on the social network that encourages use of the software. If other users exist, there is an opportunity for use. The difficulty, of course, remains in forming such networks of use. Creating a successful software product that gains global popularity is, of course, unpredictable [82]; however, if we envision adopting encryption software as adopting an application for computer-supported cooperative work (CSCW), it may alleviate some of the barriers facing widespread use of such technology.

My last chapter argues that availability and usability are not the only factors that play into a user’s decision to adopt encryption software. People weigh their decision based on numerous factors including their ability to use the software, the impact such use will have on their organization, and, what is most important for our present study, the impression their action conveys to colleagues. This chapter sets out improve our understanding of the multiple factors surrounding a users’ decision and focuses on the role peers play in transforming encryption software into more than a single-user application.

The decision to consider encryption software as a CSCW application sheds light on the problems associated with adoption, particularly the startup costs of supporting group communication; in other words, many people must work in order to ensure that the secrets belonging to a few select users remain protected. Grudin generalizes this argument in terms of CSCW applications [48]:

> . . . many CSCW applications will directly benefit certain users . . . while requiring additional work from others. A traditional method of coping with such problems is to create new jobs or ‘redesign’ existing jobs—in short, to require people to do extra work,. . . .[but] CSCW applications will not have recourse to changing job requirements

> to the degree that often occurs when entire systems are installed. The investment and commitment are smaller and the organization won't tolerate significant disruption for each new application acquired. CSCW applications will have to be more 'group-friendly' than systems have been.

In the extreme case where information disclosure puts someone's life at risk, the extra work required to install and use encryption software represents a small investment and time commitment. ActivistCorp, the activist group I studied in Chapter 3, generally followed this paradigm; however, ActivistCorp was unwilling to change its relationship to volunteers or other outsiders. The software used to protect message confidentiality demanded full commitment from every member of a group communicating on a particular project. In Grudin's terms, encryption software has evolved around single-user adoption rather than becoming 'group-friendly.' Given these software costs, it is not surprising that the organization limited their adoption to the most clear-cut cases.

Habitual use of encryption may only occur with seamless industry standardization or other protocol standardization for every user. However, such standardization remains unlikely given the today's information hungry technology economy, where a provider's incentive to offer free online services generally stems from behavioral marketing and data harvesting potential. In addition, standardization also appears unlikely given government interest in monitoring Internet traffic [100].

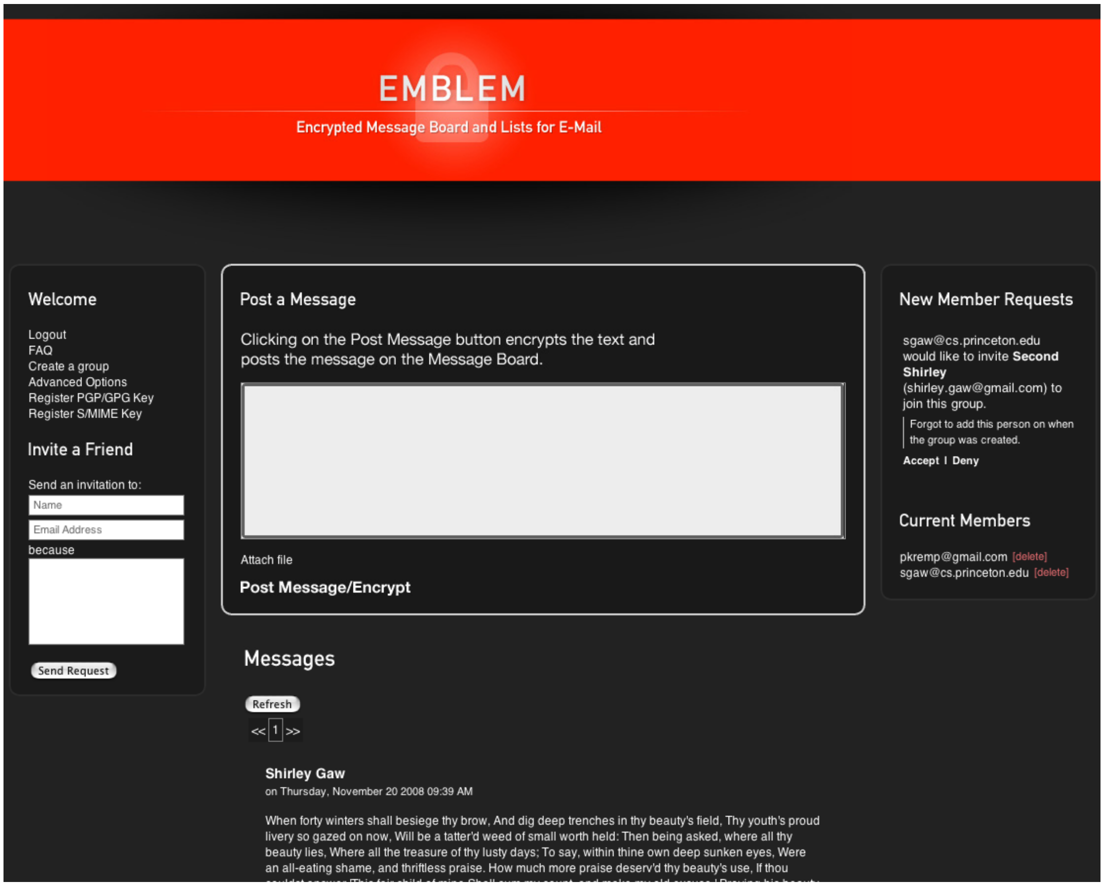
Without standards to require enforcement of message confidentiality, a possible next step between the cases represented by ActivistCorp's use and normalized cleartext e-mail is one that encourages more borderline use, where protections would be prudent but conservative estimates of the cost of introduction currently discourage use. Examples of such cases include situations where a large number of novices need to invest in new technology, where the user population has different familiarity with the technology, or where dedicated technical support is not available.

For example, at ActivistCorp, a lawyer emphatically stated her refusal to use encryption software in conversations with pro bono lawyers who volunteered their legal services. Not only was this demographic unfamiliar with the technology, but ActivistCorp had little leverage to request such a time investment from volunteers. Another example featured a man in ActivistCorp's development office discussing the difficulty of adoption, even within the organization. Colleagues in his home country were even more resistant to adopting the encryption software, as were certain offices outside the headquarters. Satellite offices had no technical support and had to resort to learning from their colleagues; in such cases, software adoption was less important than organizing direct actions.

The cases above offer examples of what Grudin describes as unfair balances of work [48]. ActivistCorp's volunteers all worked for other offices. Therefore, while ActivistCorp would have benefited from increased protections against data disclosure, pro bono lawyers, otherwise unaffiliated with the group, had less obvious benefits for their time investment. The work required of employees in satellite offices exceeded that of headquarters employees who could rely on technical support staff to ease the transition. Thus, we are left with Grundin's prediction: "can a CSCW application succeed if doing the extra work is left to individual discretion? Unfortunately, probably not" [48].

This chapter presents a design for a system that provides protected message transport and storage. In particular, I emphasize the need to support a diverse group of users that is similar to Grudin's suggestion for improving the adoptability of CSCW applications: "a typical CSCW application will be used by a range of user types – people with different backgrounds and job descriptions, all of whom may have to participate in one way or another for the application to succeed" [48]. I decrease some of the investment required for novice users, and I emphasize using encryption for specific short-term projects, rather than habitual use. Much of my focus considers interconnectedness, with minimal requirements for changes even from experienced encryption users. In sum, I envision that the new design will be more palatable for group discussions requiring greater message confidentiality enforcement but not as strenuous a commitment as demanded of ActivistCorp users.

The spirit of my system resembles other usable security systems: extraneous details are hidden from novices. My system enables novices to communicate without having to learn cryptography. EMBLEM, an Encrypted Messaging Board and Lists for E-mail, provides a web application interface for novice users and a mailing list interface for expert users. EMBLEM's design has a key feature: it splits users into different categories based on expertise, so that experts do not impose as much work on novices, and novices can also help enforce message confidentiality when their projects require it. Differentiating between types of users accomplishes two things: it streamlines the cryptographic processes for novices with an online discussion board, shown in Figure 4.1, and retains a familiar mailing list interaction for experts using their normal e-mail clients. The latter



Clicking on the Post Message button encrypts the text and posts the message on the Message Board.

sgaw@cs.princeton.edu would like to invite **Second Shirley** (shirley.gaw@gmail.com) to join this group.

> Forgot to add this person on when the group was created.

pkremp@gmail.com [delete]
sgaw@cs.princeton.edu [delete]

Shirley Gaw
on Thursday, November 20 2008 09:39 AM

When forty winters shall besiege thy brow, And dig deep trenches in thy beauty's field, Thy youth's proud livery so gazed on now, Will be a tatter'd weed of small worth held: Then being asked, where all thy beauty lies, Where all the treasure of thy lusty days; To say, within thine own deep sunken eyes, Were an all-eating shame, and thriftless praise. How much more praise deserv'd thy beauty's use, If thou

Figure 4.1: Screenshot of the EMBLEM online discussion board. The right-most panel is only shown to message board hosts. New posts are automatically encrypted with the correct group's public key when messages are stored. New messages are shown in the lower part of the screen, decrypted correctly without any user involvement.

can use whichever encryption tools they want as long as they adhere to the supported protocol (currently, OpenPGP, but this could be extended to support, for example, S/MIME as well). Thus, in contrast to other encryption systems, EMBLEM supports multiple modes of interaction where one set of users sees e-mail and another set sees an online forum. In addition, the system includes a path toward expert use without imposing such a path on novice users.

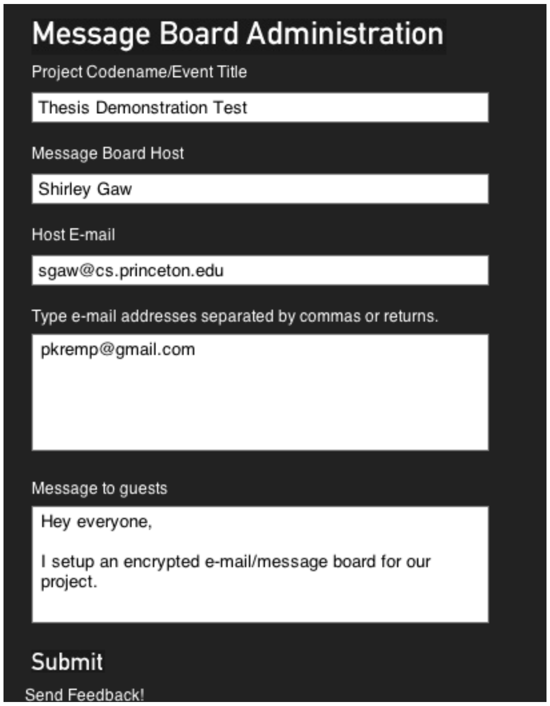
4.1 EMBLEM User Experience and Architecture

In the EMBLEM system, group hosts and technically savvy users are primarily responsible for choosing data encryption, selecting software, and installing software. Novice users have nominal usage requirements, no software installation requirements, and yet still see a novel system for special messages. Novice users may not fully understand the technology, but the system provides these users with hints that they are not using normal e-mail.

The EMBLEM system provides two modes of interaction. A group host sets up an EMBLEM server and sends invitations to the group. The users then have a choice: they can register as web users or as e-mail users. In web-based interaction, users visit the EMBLEM site through their normal web browser whenever they need to see messages posted or post messages. A web application in the webpage and the EMBLEM server handle the cryptographic operations on behalf of the user whenever messages are posted or received. A second, e-mail-based type of interaction, allows current OpenPGP users (those who have PGP or GPG keys) to register their public keys with the EMBLEM system. Instead of going to the EMBLEM website, these users receive messages from the group in their normal e-mail clients and use their own OpenPGP software to handle the cryptographic operations.[1] Whenever OpenPGP users want to post new messages, they use the public key of their EMBLEM group to encrypt messages and send this encrypted e-mail to the mailing list.

The following section describes how the system works for our three types of users: group hosts, novice users, and expert users. I use the term novice users and expert users to differentiate between levels of expertise among users. Novice users have little or no experience with cryptography and expert users already have (or are willing to have) PGP or GPG setup. I describe how each type

[1] Web-based users can also transition to being OpenPGP users if they wish.



Message Board Administration

Project Codename/Event Title

Thesis Demonstration Test

Message Board Host

Shirley Gaw

Host E-mail

sgaw@cs.princeton.edu

Type e-mail addresses separated by commas or returns.

pkremp@gmail.com

Message to guests

Hey everyone,

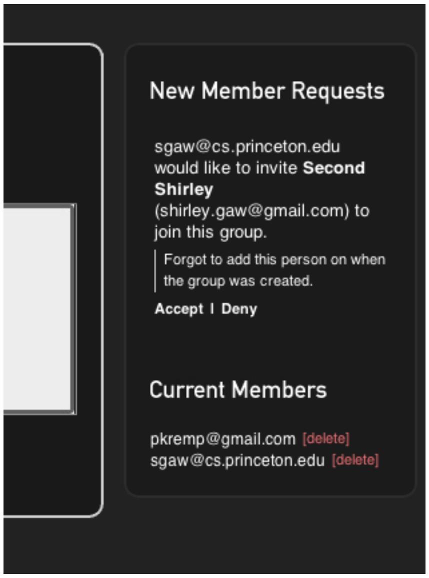
I setup an encrypted e-mail/message board for our project.

Figure 4.2: Screenshot for a group host who is creating a new EMBLEM group.

of user interacts with EMBLEM in the User Experience section and explain the functions and workings of the system in the System Design section.

4.1.1 User Experience

To create a group, the group host connects to a known EMBLEM server (or installs one herself). She creates a new messaging group, using the form in Figure 4.2. After creating a list of all of the desired members, EMBLEM sends e-mail invitations to the potential members. In addition, when the group host uses EMBLEM, her view displays some administrative tasks, see Figure 4.3. Administrative tasks include receiving notifications when members want to invite new people to the group and seeing a list of current members in case the host wishes to revoke access. Revoking access removes web users from the system and prevents them from seeing the online message archive or future posts. Revoking expert users denies access to future posts; however, these users may already have a local archive of previous posts and could therefore still be able to access the archives.



New Member Requests

sgaw@cs.princeton.edu would like to invite **Second Shirley** (shirley.gaw@gmail.com) to join this group.

> Forgot to add this person on when the group was created.

Accept | Deny

Current Members

pkremp@gmail.com [delete]
sgaw@cs.princeton.edu [delete]

Figure 4.3: Group Host's User Experience: Zoomed in administrator's view, highlighting pending invitations and current memberships.

A novice begins using EMBLEM in the following manner.

1. He receives an e-mail from the EMBLEM system inviting him to join a new group.

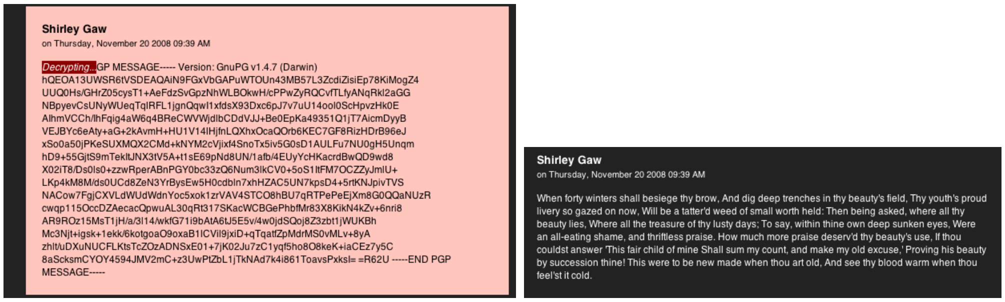
2. He clicks on the message's embedded link or opens his browser and navigates to the EMBLEM page.

3. EMBLEM guides him through a registration step to setup his login information.

4. EMBLEM displays the discussion board (shown in Figure 4.1), which periodically updates with new messages.

Novices can then post replies through the EMBLEM website, and the web application will automatically forward messages to the EMBLEM server for encryption. The server will then send this post to the mailing list. Eventually, the novice user will receive an e-mail notification that a new message has been posted.

It remains significant that a novice retains his normal e-mail account and/or e-mail client for receiving invitations and notifications but uses his web browser to see and post messages. My system includes no new software applications for a novice. In the discussion board, the novice periodically sees new messages. The system hints at the cryptographic operations performed



Shirley Gaw
on Thursday, November 20 2008 09:39 AM

```
Decrypting...GP MESSAGE----- Version: GnuPG v1.4.7 (Darwin)
hQEOA13UWSR6tVSDEAQAiN9FGxVbGAPuWTOUn43MB57L3ZcdiZisiEp78KiMogZ4
UUQ0Hs/GHrZ05cysT1+AeFdzSvGpzNhWLBOkwH/cPPwZyRQCvfTLfyANqRkl2aGG
NBpyevCsUNyWUeqTqlRFL1jgnQqwl1xfdsX93Dxc6pJ7v7uU14ool0ScHpvzHk0E
AlhmVCCh/lhFqig4aW6q4BReCWVWjdlbCDdVJJ+Be0EpKa49351Q1jT7AicmDyyB
VEJBYc6eAty+aG+2kAvmH+HU1V14lHjfnLQXhxOcaQOrb6KEC7GF8RizHDrB96eJ
xSo0a50jPKeSUXMQX2CMd+kNYM2cVjixf4SnoTx5iv5G0sD1AULFu7NU0gH5Unqm
hD9+55GjtS9mTekltJNX3tV5A+t1sE69pNd8UN/1afb/4EUyYcHKacrdBwQD9wd8
X02iT8/Ds0ls0+zzwRperABnPGY0bc33zQ6Num3lkCV0+5oS1ltFM7OCZZyJmlU+
LKp4kM8M/ds0UCd8ZeN3YrBysEw5H0cdbln7xhHZAC5UN7kpsD4+5rtKNJpivTVS
NACow7FgjCXVLdWUdWdnYoc5xok1zrVAV4STCO8hBU7qRTPePeEjXm8G0QQaNUzR
cwqp115OccDZAecacQpwuAL30qRt317SKacWCBGePhbfMr83X8KikN4kZv+6nri8
AR9ROz15MsT1jH/a/3l14/wkfG71i9bAtA6tJ5E5v/4w0jdSQoj8Z3zbt1jWUKBh
Mc3Njt+igsk+1ekk/6kotgoaO9oxaB1lCVil9jxiD+qTqatfZpMdrMS0vMLv+8yA
zhlt/uDXuNUCFLKtsTcZOzADNSxE01+7jK02Ju7zC1yqf5ho8O8keK+iaCEz7y5C
8aScksmCYOY4594JMV2mC+z3UwPtZbL1jTkNAd7k4i861ToavsPxksl= =R62U -----END PGP
MESSAGE-----
```

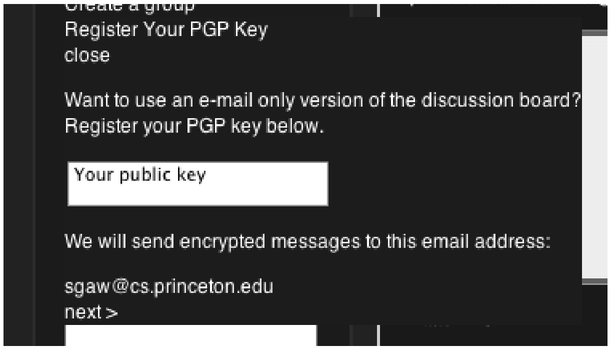
Shirley Gaw
on Thursday, November 20 2008 09:39 AM

When forty winters shall besiege thy brow, And dig deep trenches in thy beauty's field, Thy youth's proud livery so gazed on now, Will be a tatter'd weed of small worth held: Then being asked, where all thy beauty lies, Where all the treasure of thy lusty days; To say, within thine own deep sunken eyes, Were an all-eating shame, and thriftless praise. How much more praise deserv'd thy beauty's use, If thou couldst answer 'This fair child of mine Shall sum my count, and make my old excuse,' Proving his beauty by succession thine! This were to be new made when thou art old, And see thy blood warm when thou feel'st it cold.

Figure 4.4: Novice User's View of New Messages. When new messages arrive, the messages are initially shown as ciphertext, accompanied with the warning "decrypting." When the decryption finishes, novices are shown new messages with readable text.

on behalf of the user, with an example shown in Figure 4.4. New messages flash in pink, and novices see that messages initially appear as ciphertext gibberish with a warning that the system is "decrypting." Some moments later, a novice notices the cleartext message appear. Novices can also add posts to the board, although they will briefly see a warning that the message is being encrypted. A ciphertext version of the message flashes instead of the novice's message. Instead of the message appearing as a new post, a pending message in ciphertext appears above the discussion board. This disappears when EMBLEM detects that the message has been successfully received by the group. A few minutes later, a novice sees his new post flash in pink in the message board. The notification system also tells him when new messages are posted to the board. Finally, a novice can also ask the group host to invite others to join the group, in case the host had omitted anyone in the initial round of invitations or in case group members want to expand the group (see the left hand side of Figure 4.1).

Initially, an expert's user experience resembles that of a novice's. An expert receives a notification that a group host has invited her to join an EMBLEM group. After completing the authentication step, an expert can choose the Advanced Options on the message board, shown in Figure 4.5. She can then copy a text version (called an *ASCII-armored* version) of her OpenPGP key and paste it into the website, which will enable her to register her public key through the EMBLEM system. The EMBLEM website will then show her the group's e-mail address for the mailing list along with an ASCII-armored version of the public key that she needs to use for encrypting e-mail messages to the EMBLEM group. She can store a copy of the group's public key on her local machine's keystore. After the public-key registration, an expert will receive copies of



Create a group
Register Your PGP Key
close

Want to use an e-mail only version of the discussion board?
Register your PGP key below.

Your public key

We will send encrypted messages to this email address:

sgaw@cs.princeton.edu
next >

Figure 4.5: Experts's Advanced Options. People who already know how to use PGP or people who would prefer use a single e-mail inbox can export their public keys to EMBLEM and import the group's public key.

new posts in her normal e-mail account. These copies will be encrypted with her public key. Thus, an expert can recover the original text of the message after using her own OpenPGP application. She is able to decrypt messages by using her own process. For new posts, she resorts to her own process to generate a ciphertext message with the group's public key and sends an e-mail through her own e-mail client (or her own webmail account) to the group's mailing list.

These dual modes of interaction, one for experts and one for novices, are mutually compatible. EMBLEM provides novices with a level of indirection that only fills their inboxes with notification messages, while the actual messages remain in EMBLEM's mail server. In addition to a normal e-mail program, the novice has a virtual inbox for encrypted messages in the EMBLEM system. I have designed EMBLEM to encourage interested users to fuse their inboxes. Anyone willing to become an expert in the system by generating a key pair and registering a public key, would receive the benefit of having only one inbox even as they join more EMBLEM groups or as the traffic for an EMBLEM group increases. In this sense, becoming an expert could be a convenience, and pursuing convenience also increases practical security. EMBLEM serves as a mediator between novice and expert users. However, if all users transferred to expert mode, EMBLEM would become unnecessary.

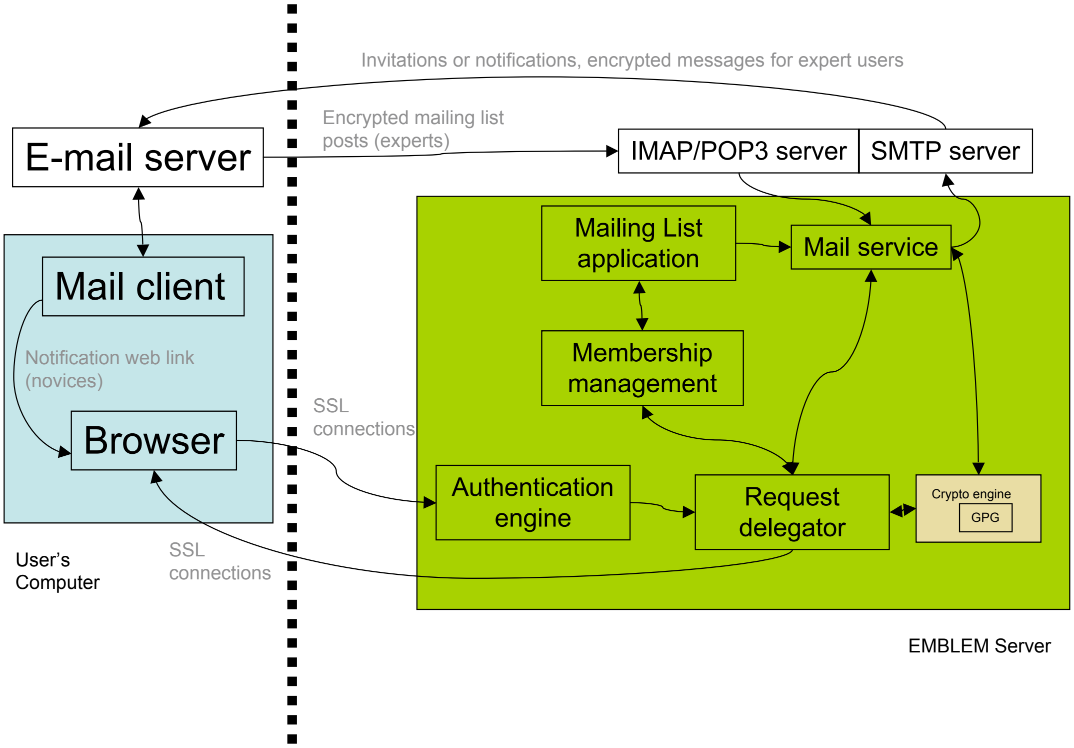


Figure 4.6: EMBLEM system design assuming a novice user with an e-mail client application. A novice receives invitations and notifications through his normal e-mail client and e-mail server. These messages provide links for visiting the EMBLEM server. A novice connects to EMBLEM over an SSL connection in his normal web browser. The system automatically handles cryptographic operations for the user. The mail server stores an archive of encrypted messages.

4.1.2 System Design

Figure 4.6 features the EMBLEM server's system design.[2] Novice users see a discussion board in their browser, and the web application sends requests and receives responses from the EMBLEM server over an SSL connection. The EMBLEM server has five main components:

1. Authentication engine
2. Request delegator and web server
3. Mailing list application service
4. Cryptographic engine and key manager
5. Group membership manager

When the server receives a new request, an authentication engine determines whether the user is a member of the system and, if so, which groups the member belongs to. The authentication engine also checks the legitimacy of the request (see Section 4.3.3 for a discussion of cross-site request forgeries). After the authentication step, the request delegator becomes the primary interface for all client transactions. Most administrative actions, such as creating a new group, approving memberships, and removing memberships are passed on to the group membership manager, where the host's web application requests these services. The web applications of group members periodically request the EMBLEM server to check and download e-mail messages for a specified mailing list group. New messages are sent to the client, who responds with requests to decrypt these messages (this inefficiency could be removed in future implementations). Each decrypted message is sent back to the client over an SSL connection. Similarly, new posts are sent over SSL to the EMBLEM server, where the cryptographic engine encrypts the message before shipping a copy to the mailing list (although the client also briefly displays the ciphertext message for the user as well). When new messages are posted, group members receive notification messages from the server.

An application version of the EMBLEM system periodically checks the mail servers for new messages. For expert users, the application decrypts the new message with the group's private key

[2] Technical details concerning EMBLEM's implementation, such as the libraries and software packages, are in Appendix C.

and re-encrypts one copy for each expert user. Each copy is encrypted with the user's registered public key. The re-encrypted messages are then sent by e-mail to the expert user.

4.2 Deployment Issues

Most of burden in deploying EMBLEM falls on the group host and advanced users. The following section provides a detailed survey of the issues facing all users, starting with novices.

4.2.1 Novice Use Issues

Novice users in EMBLEM have one main responsibility: they must create a new online account. The group host invites the novice user, which signals that the novice is expected to accept the host's security decisions about the confidentiality required for the communication.[3]

The novice avoids decisions on all cryptographic operations. When the user creates an account by finishing the one-time registration step, EMBLEM (unbeknownst to the user) generates the key pair for encryption. Every time the novice user posts a message through his web application, EMBLEM automatically bundles message posting with cryptographic operations (by using a shared group public key). Novices do not decide whether or not to use encryption; instead, the only decision facing novices is whether or not to use EMBLEM. By handling the cryptographic operations automatically with message posting, the system prevents novice users from making mistakes in key selection: they do not have direct access to the private key, nor can they use the wrong encryption key.

Learning how to use EMBLEM requires minimal effort on the part of novices. The system appears in a familiar interaction environment: e-mail and web browsing. This approach avoided designs that required installing software or plug-ins. The e-mail notifications warn the novice about new messages in the EMBLEM system. Apart from the authentication mechanism, the user is free from any interaction other than checking messages and posting new messages, just as one does on any other web forum or blog comment page.

For increased security, browsers may block execution of the Javascript in EMBLEM webpages. If the user has turned off Javascript, we can assume that the user is comfortable and knowledgeable

[3] A disagreement about whether such vigilance is needed (see Chapter 3) not only makes the system an ill-suited solution, but also a system that even users with encryption experience will avoid.

enough to turn on Javascript. In a worst case scenario, a novice may have a pre-configured environment managed by their organization. In this case, the user may need guidance to turn on Javascript for EMBLEM or may give up using EMBLEM altogether.

As Chapter 2 demonstrates, increasing a user's number of online accounts seems to encourage password reuse. Avoiding user login authentication remains preferable, but there are certain pragmatic trade-offs. Login authentication asks users to choose a password and users tend to choose poorly for the sake of convenience. I had considered implementing the ESCAPE system to use in EMBLEM for access control [6]. ESCAPE issues invitations to group members. When a group member visits the ESCAPE website, ESCAPE installs a client certificate in the user's browser. The system then employs this certificate for client authentication in the SSL/TLS protocol on every subsequent time the user visits the website, so that the ESCAPE system can have access control without password authentication. This system, however, limits a user's access to a single browser on a single machine. As Chapter 2 mentions, the benefit of using password authentication for an online application is portability. Any browser can support basic login authentication, and users only need very little information to access the account. Therefore, by supporting password authentication in EMBLEM, the system supports greater portability. Users can share computers (so long as they are automatically logged out of their accounts or are diligent about logging out of their accounts) and can use the system on multiple machines and public terminals.

4.2.2 EMBLEM Server Deployment

The centralized design of EMBLEM requires a group host to find a vendor who already runs EMBLEM as a web service or that a group host chooses to run EMBLEM herself. Less technically savvy users can readily use EMBLEM as a web service much like Facebook and E-vite, which provide web services for gathering network information, posts, and inviting new users. However, unlike most services which are not overly concerned with security, we must consider that the EMBLEM server is particularly vulnerable to attack. Thus, EMBLEM is best replicated for each discussion group.[4] Running EMBLEM requires the following packages:

EMBLEM Web Application Archive (WAR): Install a bundled distribution of the web scripts,

[4] Although this approach can lead to problems for users in distinguishing trusted and untrusted EMBLEM servers, see Section 4.3.3.

pages, and images and library

Tomcat: Install a Tomcat server for running the EMBLEM web services

GPG: Install the GPG application (already included on Unix and Linux distributions)

JNI library and GPGME: Install a library for accessing GPG functions (GPG Made Easy, GPGME). Also install a JNI library that communicates between the EMBLEM servlets and GPGME

mail servers: Create an account with a webmail service (for example Yahoo Mail) that supports SMTP and POP3 or IMAP access. Alternatively, install and run a mail server.

In addition, running an EMBLEM server requires the following configuration steps:

1. Configure EMBLEM for the correct mail servers and library search paths.
2. Replace Tomcat's default configuration with a requirement for SSL connections.
3. Generate or purchase a certificate for server authentication in SSL connections.
4. Setup virtual hosting to resolve an alias URL to the IP address of the Tomcat server.

These steps are far more complicated than the easy startup typically offered by web services; however, a production version of the system could simplify the steps by bundling the software packages together. It is also important to remember that a single technical person can deploy EMBLEM for all participants. Rather than burdening everyone with the complexity of system, as many distributed encrypted e-mail systems do, EMBLEM only asks that a select group agree to help support the system. Since this bears similarity to online software mirrors, it is reasonable to hope that more than one trusted EMBLEM server exists for general use.

4.3 Security Analysis

While automated systems and centralized systems can mitigate some of the demands placed on users, the process introduces security problems. This section describes some of the weaknesses in the EMBLEM design. Overall, having privileged users as hosts provides a point of contact for diagnosing problems or reporting compromises that other systems may not support. Furthermore, the EMBLEM approach has the ability to increase the security of communications in groups when the convenient alternative is cleartext messaging. Although there are some weaknesses introduced with EMBLEM, the system remains preferable to unencrypted traffic. If users are capable of using more complex systems, they can still use these systems in conjunction with EMBLEM. If all members use distributed cryptographic systems such PGP, then EMBLEM becomes unnecessary. The benefits of EMBLEM are apparent when at least one member in the group lacks experience with encrypted e-mail. Today, this situation is more common than having a group of expert (and dedicated) users.

4.3.1 Server Key Store Attacks

As would be expected from a centralized system, the EMBLEM server is the primary and weakest point of the system. The problem is more than denial of service, discussed later. The biggest problem lies in the fact that the server decrypts messages with a group's private key. Ideally, the communications between individuals would rely on distributed private keys, much like the traditional PGP system. A production version of the system should store keys on individual machines; however, a trade-off exists between constraining requirements of software installation and distributing private keys. A centralized key store simplifies the design; EMBLEM uses this approach for prototyping.

Such centralization is embedded in transactions with web applications and transactions with expert users who received individualized re-encrypted messages. The process leads to two problems: malicious EMBLEM hosts and compromised keystores. A maliciously deployed EMBLEM server could capture all traffic through EMBLEM groups and see messages in the clear. Similarly, a compromised EMBLEM server leaves archives of messages vulnerable to decryption, as the Hushmail case reveals (discussed in Related Software, see Section 4.5). Given that malicious EMBLEM servers are problematic, suspicious experts would need to serve EMBLEM themselves.

Distributed private keys would render the web application in EMBLEM more powerful. The web application, rather than the EMBLEM server, would execute cryptographic operations such as generating keys, importing keys, and encrypting and decrypting messages (or, alternatively, instead of a web application, the system could be implemented as an applet, following Hushmail's example). This would increase the importance of digitally signing messages, with the intention of proving message integrity.

The EMBLEM design factored in the expectation that the system would serve project-based discussions, where long-term key usage is less necessary; nevertheless, the design does not prevent users from spending more time on the system. Another solution to offset the risk facing EMBLEM's keystore would involve using authentication agents for key management. These systems could protect private keys with the passwords cached for a limited period of time. In that case, the host of the group or the host of the server would be responsible for assuring that the agent is initialized to recall the key passwords before people use the system.

4.3.2 Monitoring Attacks

A user's normal browser is part of the EMBLEM trusted computing base, which can lead to misplaced trust. A malicious browser or a plug-in that monitors the page display will likely intercept all messages in the clear. Similarly, keyloggers would be able to intercept cleartext. Although browsers present a particular vulnerability for EMBLEM, keyloggers are a problem for most encrypted communication systems.

However, monitoring the SMTP server is a specific vulnerability for EMBLEM. An attacker can use traffic analysis to determine a group's membership as well as the specific timing of a planned event. Normal e-mail conversations distributed across groups (and, presumably, across SMTP servers) would not be vulnerable to this problem, but any listserv would have suffer from similar vulnerability.

4.3.3 Browser-related Attacks

As EMBLEM relies on novice user interaction through a browser, any logged-in user opens EMBLEM to common browser attacks, and the system requires defenses against things such as cross-site scripting and cross-request forgeries. First, the web application avoids presenting HTML encoded data to protect against cross-site scripting attacks (XSS). Currently, the system only supports plaintext encoded messages and prevents the display of user submitted HTML; therefore, for instance, cleartext messages are not interpreted as HTML. EMBLEM displays ciphertext for reference – users can only create or alter cleartext posts.

Logged-in users also render EMBLEM vulnerable to cross-site reference forgery (XSRF); The system is hardened against cross-site reference forgeries using the suggestions from Zeller and Felten [104]; however, unclosed sessions remain problematic for the system, since any logged-in user has full access to all current postings on the discussion board. Expiring the page can mitigate this occurrence, but identifying users is also troublesome. For example, using password authentication introduces all of the problems germane to dictionary attacks. Portable browsers with a certificate could be installed on USB keys for public terminal use; however, this poses a problem if the USB key is compromised. Completely avoiding password authentication without significant user burden is unlikely in most design alternatives.

Another problem facing the EMBLEM website is that users need to trust the EMBLEM server. Since any group may use a different EMBLEM server, attackers could launch phishing attacks. Websites with similar names or similar notification/invitation messages could lure users into submitting their login information into an attacker's website instead of the group's EMBLEM website. Prior work in phishing, namely security skins [28], may help with this problem. However, urging the novice user to subscribe to more than one EMBLEM server easily runs the risk of becoming unmanageable for the user.

4.3.4 Denial of Service Attacks

Attackers who simply take down the EMBLEM server will easily deny service for all mailing lists on that particular server. In addition, simply deleting messages from the mail server effectively eliminates the entire message archive for all web users. Although this is a vulnerability in the system, I imagine this feature might be beneficial should the host want to destroy the archive. Depending on the mail server, however, it may take some time for the complete destruction of certain archives.

An EMBLEM attacker could also deny service to individuals or to all users. Attacking the membership manager can remove users from the system silently. Members might only notice a lack of traffic in the system but should realize the problem when they login. Another remaining problem is that EMBLEM functions an invitation-based system, and, similar to ESCAPE's invitation system, there is no guarantee that the first person to respond is the intended recipient. EMBLEM is also particularly vulnerable to anybody monitoring the SMTP server. If the initial invitations are intercepted, the intended users are blocked from service but may be unaware of the fact. The attacker can use the invitation to lurk on the list and intercept traffic in in decrypted form; however, Balfanz argues that the increased practical security gained by adopting his system outweighs the costs of more secured implementations that will likely be abandoned; this could also apply to the EMBLEM system [6].

4.4 User Experiences

Grudin states that the difficulties in evaluating CSCW applications contribute to a lack of widespread success in CSCW application development ("it is difficult if not impossible to create a group in the lab that will reflect the social, motivational, economic, and political factors that are central to group performance. In addition, group observation must extend over a longer period of time [than evaluation of single-user applications]" [48].) The concept of mixing both novices and experts and urging them to use the EMBLEM prototype instead of a mailing list for a project will be ideal, especially if they were a subset of activists from ActivistCorp (see Chapter 3) and if they could use the system for several months; however, this kind of evaluation would be more reflective of that type of user. Therefore, it would be appropriate to monitor wider use by releasing the software to anyone who wanted to try it, or by recruiting people to use it for their projects. This approach would be time-consuming, however, and it is unclear that the results would be promising given the specificity of the intended use case.

It remains difficult to evaluate security prototypes in realistic settings, which only adds to the problem. One developer in the activist group security software explains, "we've been working on [our own project], and are struggling with the same conundrum: we know that there are problems with our setup, but at the same time, we know that the more users we get, the better it will become. But how do you ethically attract users to a system that you know is flawed?" Obtaining realistic evaluations seems to put participants at risk; I would want for the participants to commit and get more involved with the design in order to receive better feedback. However, a commitment would put their communications at risk. For people who truly needed increased confidentiality, maybe I would be better off teaching them how to use encrypted e-mail for secured communications rather encouraging use of a prototype system.

In light of these constraints, from a security and resource perspective, I opted for a more exploratory and lightweight evaluation. I asked two groups of graduate students to try using EMBLEM. The first group included three sociologists at Princeton who used EMBLEM for their dissertation support group e-mails over the course of eleven days. The second group consisted of seven security students (including myself) who spent three days using EMBLEM for testing and chatting.

The first group (two females and one male) used EMBLEM in lieu of the messages they sent back and forth for their dissertation group. Most of the activity for the group occurred on the first day (six of the eight messages posted to the board). Afterwards, the activity died out. Most of their feedback related to user interface ambiguities or places for improvement, but one interesting question arose. A participant asked: "why do you see weird text when messages are sent?" She had difficulty conceptualizing what took place invisibly behind the scenes and what benefit could be derived from encrypting messages. Her response to the system, even after a conversation that explained the design and motivation, consisted of providing me with data rather than feeling EMBLEM was useful.

The second group (initially four males, but eventually six males and also myself) consisted of members from my security research group. The group sent 30 messages on the first day, but dropped to 10 messages on the second day and nothing on the third day. My postings were limited to responses to participant questions (three posts). Four of the males had used GPG for a group project they had worked on together. Much of their experience using EMBLEM centered around testing (such as seeing if they could send large messages and if the system cleaned against scripting attacks), and one of the big complaints was the slowness of the system. They complained about the time it took for messages to appear after posting and how slowly the system started loading messages after an initial login. They also expressed annoyance at the excess of notifications (every post generated a new notification). Most felt that the system, while usable, was not beneficial, as summarized by one: "if you were already paranoid enough to encrypt your e-mail, you would be worried about using your scheme [EMBLEM]." In particular, the centralization of the encryption worried them, since this provided a single point of attack; however two of the participants said it would be good for a group that included novice users.

Given that the system was designed for groups with members of mixed experience with encryption, both groups of students represented far from ideal case-studies. Though the inner workings of EMBLEM remained unclear, the first group represented novice users and the purpose of enforcing message confidentiality made sense to at least two of the members. Additionally, their traffic in the discussion group related to a specific project, which reflected the design goal of the system. The second group represented a much more experienced and much more wary set of users than would be expected for EMBLEM; their cautious nature is reflected in their concerns about the security of the system. For a set of experienced encryption users or a set of highly motivated security users, EMBLEM's design is a poor fit.

4.5 Related Software

First, I address how EMBLEM compares to end-to-end encryption offered by S/MIME and OpenPGP systems. Then, I discuss other systems and interfaces related to encrypted e-mail.

As mentioned earlier, Whitten and Tygar initiated the analysis of encrypted e-mail usability in 1999 [98]. Through a laboratory user study, they demonstrated that users newly exposed to a data encryption software, Pretty Good Privacy (PGP) 5.0, largely failed to correctly complete the requested cryptographic tasks, including the following:

1. Generate a public-private keypair.

2. Acquire public keys from all group members.

3. Distribute their own public key.

4. Compose a new e-mail message, encrypting the body with the recipient's public key.

EMBLEM hides nearly all of these tasks from novice users. Keys used by novices are generated and stored on the EMBLEM server. New messages are automatically encrypted with the group's public key. Expert members already familiar with PGP (or the Gnu equivalent, Gnu Privacy Guard, GPG) receive correctly encrypted messages with the recipient's private key, since EMBLEM automatically re-encrypts messages for these users.

Garfinkel and Miller argue that most of the problems faced by Whitten and Tygar's users "simply do not exist in programs like Microsoft Outlook, Outlook Express (OE), and Netscape

Communicator, all of which have integrated support for the S/MIME email security" [38]. Further, they explain how S/MIME's design ameliorates or removes some of the issues encountered by Whitten and Tygar's users.

Even with this support, S/MIME still fails to attract a significant user base. The fact that more users turn to webmail over traditional e-mail clients only exacerbates the problem: webmail clients generally lack both S/MIME and OpenPGP support (although some plug-ins exist for adding this functionality, see below). Garfinkel et al. argue that having a few vendors supporting S/MIME could remove this barrier [37]. Even if browsers made such encryption available by default, initializing the system remains difficult. The process of obtaining the key pair required for cryptographic operations still discourages novices.

My contribution to the work in encrypted e-mail offers a system that removes the initialization barrier and simplifies interaction for communication. While S/MIME or OpenPGP (as it was originally designed) may present a better protocol for users already initiated to cryptography, most users are completely unfamiliar with encryption. Our system expects that while the novice users lack understanding of the importance of encryption, they may be open to trying the system in special cases. An EMBLEM group resists reduction to the lowest common denominator, regular e-mail. An essential design factor in EMBLEM offers people who believe they have secret data and require confidential conversation an inconspicuous means of introducing technology. Novices also benefit from limiting their exposure: they simply visit a website; terminating use of the system leaves no imprint on their computers since software was never installed.

Systems similar to EMBLEM are already available and, as mentioned above, plug-ins can lighten the requirements for using and starting to use encrypted e-mail. Often, these systems require complete homogeneity in deployment: control of traffic on both user and recipient machines and similar software installation on both machines or similar web applications and accounts. Such an expectation is unreasonable outside of a single organization that already has technical support or technology policy enforcement [40].

When implementing the OpenPGP standard in systems such as PGP or GPG, data recipients have to install an application to handle data encryption, decryption, and signing. Identity Based Encryption (IBE) avoids this problem by allowing the sender to use arbitrary strings, such as e-mail addresses, for the public key [14]. Yet, both the sender and the recipient need to install the IBE system in order for this to work. In contrast, though S/MIME solves the implementation problem, the initialization remains a hurdle for users. While theoretically more usable, IBE and S/MIME are only practical for the technically savvy who already know they would like their messages encrypted and who agree on the protocol and software implementation for the procedure.

Even webmail users can add encryption support with plug-ins that either operate through the browser or inject scripts into specific webmail applications. Enigmail and FireGPG are examples of application plug-ins. They require the installation of key manager (GPG) as well as a plug-in. In addition, they still require the user to generate a key pair and distribute the public key. Other plug-ins such as Gmail Encrypt, freenigma, and Gmail S/MIME for Gmail use script injection to add support to Gmail's webmail interface. These plug-ins require that every recipient install the plug-in and, except for Gmail Encrypt, procure a S/MIME certificate or GPG key pair. Gmail Encrypt only works with other users of the same systems and likewise requires similar key generation. If users have unsupported webmail services (such as internal webmail clients for their organization), they are stuck. Furthermore, many plug-ins only support the Firefox browser, requiring a browser install for most web users.

PGP Universal Server and Hushmail serve as primary examples of server solutions. PGP Server works by filtering traffic between the mail servers and the Internet cloud [73]. Encryption and decryption of messages are automatically handled, provided that a PGP Server exists on the sending and recipient sides and both sides properly configure their systems. The invisible nature of the secure system is helpful when ubiquitous message encryption occurs since it enables users to expect what is happening; silent encryption systems completely remove the user from the system, which complicates matters if, for instance, the user wants to know if the system has failed.

Hush Communication produces tools that are most similar to EMBLEM's service [54, 53]. Like EMBLEM, Hushmail (in the webmail version) supports asynchronous messaging through a web application and supports compatibility with current PGP/GPG users. Hushmail's basic service directly transfers the e-mail interaction to a web interface, complete with an inbox, however. Direct translation of e-mail to e-mail is less useful than translation of e-mail to web forums for short-term projects (although the experiences in the second user study contradict this). A discussion board or web forum is conducive to broadcast messages where few users actively provide content; this is, in fact, the way most online forums and mailing lists behave. Finally, in Hushmail, as in EMBLEM, users may invite others to join the service.

In addition to the interaction issues, sole dependence on a single vendor, such as Hush Communications, frequently complicates privacy issues as the vendor holds the private keys or potentially accesses cleartext versions of communications. For example, recent press has revealed that users were vulnerable to the Hushmail company's willingness to turn over decrypted communications to the US government in 2007 [87]. EMBLEM addresses this by supporting multiple EMBLEM deployments, but it remains to be determined whether this is a practical solution.

4.6 Conclusions

Many tools, plugins and additions to traditional e-mail clients have simplified secure messaging. Such tools can only automate cryptographic steps when users have already properly configured and loaded the system. Unfortunately, many of the hurdles occur *before* users reach a state where they can easily encrypt and decrypt.

This chapter introduces EMBLEM as a system that supports novice users with using encrypted message storage and transmission. My system relies on a trusted server to manage discussion groups. The server automates cryptographic operations for novice users. Expert users, who are already familiar with OpenPGP, see a mailing list interface in their e-mail client. Furthermore, if expert users are wary of trusting the server, they are free to provide their own deployment of the service. While this system is less secure than distributed use of OpenPGP or S/MIME messaging, it provides uninformed users with a familiar interface that one can adopt more readily.

I use a level of indirection for novice users which enables them to receive notifications instead of actual ciphertext messages. Potentially, this requires two applications instead of just one (the e-mail client and the web browser). Automated solutions would be more convenient. If a few of the major webmail services agreed to turn on encrypted message storage, the message confidentiality problem would disappear and EMBLEM would become superfluous. However, this outcome is currently unlikely: webmail providers have an incentive to store cleartext messages for both data mining and system flexibility.

Web applications "computing in the cloud" provide users with novel and easily managed software. They need not worry about updates or installs. Web applications remain flexible to lightweight packaging in widgets and regrouping into novel features or portals. This flexibility as well as desirability has increased the prevalence and adoption of web applications. Unfortunately, it has also increased the dependence on external data storage, which creates potential issues for data privacy and may increase observation of individuals. Companies in possession of the data generated by their users own it. Admittedly, most individuals may not be overly concerned with private ownership of user data, but a possibility remains that this issue will grow in importance.

EMBLEM offers an example that shows how the increasing power of emerging web applications can enable individual resistance to observation and data mining. The wide browser support for web applications leaves the possibility that widget versions of cryptographic engines could enable a layman's use of data encryption. Currently, SSL solves encrypted transit for ordinary users. Systems such as EMBLEM can move toward solving encrypted storage for ordinary users. EMBLEM enables a layman to encrypt his own messages, his own data storage, and his own digital conversations. The underlying question remains whether such systems would be beneficial over the currently available tools and if this empowerment would truly address the problems that people may or may not have with eavesdropping. The design of EMBLEM, however, addresses a specific case and it provides an example of designing between two extremes, the nearly impenetrable system and the easily compromised system. Hopefully, this approach to design will prove more effective at promoting security when required.

Chapter 5

Conclusion

In a series of lectures to U.S. national intelligence employees, David G. Boak presented motivations for working on communications security. His foresight regarding the issues facing deployment and adoption of security systems now strikes us as prescient [13]:

> Those of you who are cryptoanalysts will find yourselves in an environment that is necessarily cautious, conservative, and with security per se a truly paramount consideration. But do not lose sight of the real world where your ultimate product must be used, and beware of security features so intricate, elaborate, complex, difficult, and expensive that our customers throw up their hands and keep on communicating in the clear.
>
> From this, we can conclude that, to carry out our job we have to do two things: first we have to provide systems which are cryptographically sound; and second, we have to insure that these systems can and will be used for the purpose intended.
>
> If we fail in the first instance, we have failed those customers who rely on our security judgments and put them in a disadvantageous position with respect to their opposition. But if we fail to get the systems used—no matter how secure they are—we are protecting nothing but our professional reputation.

While his lectures are over forty years old, Boak brought to light an issue that is still central for research in computer security today: the practical deployment of security systems in the real world. At the beginning of his lecture series he emphasized the possible disconnect between securing communications and real-world work behind securing those communications. The important metric, according to Boak, was not merely theoretical security that could be achieved ideally but the level of security achieved from successful deployment and adoption of security systems.

By studying people's security habits, the results of my thesis describe cases in which more secure practices could increase the level of achieved security in theory. Realistically, however, such a level of security is rarely found or sustained. My work highlights the fact that secure practices are not adopted by individuals simply because they are available. In this conclusion, I will frame my work in HCI-Sec on the adoption of secure technologies and practices with an analogy to Becker's work on *deviant careers* [10].

5.1 Security Practices and Career Contingencies

In his monograph, *Outsiders*, Becker focuses on the careers of marijuana users. While marijuana use and computer security practices differ radically, they nevertheless both diverge from normal and accepted behaviors, and Becker's description of how deviant careers develop is applicable to my analysis. After a brief summary of relevant results from Becker's study, I will explore the parallel further. Becker explains that access to the drug, while influential, only represents one element in the career of a marijuana user. Becker describes that the development of a marijuana user derives from socially learned enjoyment – a marijuana user can access marijuana, but transitioning from access to habitual use requires more (pages 30-31) [10]:

> Before engaging in the activity on a more or less regular basis, the person has no notion of the pleasure to be derived from it; he learns these in the course of interaction with more experienced deviants. He learns to be aware of new experiences and to think of them as pleasurable. What may well have been a random impulse to try something new becomes a settled taste for something already known and experienced. The vocabularies in which deviant motivations are phrased reveal that their users acquire them in interaction with other deviants. The individual *learns*, in short, to participate in a subculture organized around the particular deviant activity.

According to Becker, a "normal" person does not simply become a deviant marijuana user or fall into the practice overnight. People who become deviant do not necessarily display underlying deviant individual characteristics or predispositions. Rather, deviant practices are *learned* and *reinforced* through more experienced individuals who pass along knowledge of how to execute the practices as well as how to derive enjoyment from these practices.

This model of career contingencies that support transition from one practice to another can be applied to the development of security practices. Factors in the development of deviant behavior can translate to certain factors in the development of secure practices, and the model also addresses our commonly held, yet often erroneous, beliefs of what these influences represent. *Access to technology* and usability of technology are factors that influence adoption but, but like access to marijuana, this availability does not translate directly into use. People must *learn about security threats and responses to security threats.* Furthermore, they must both execute secure practices and also *derive a level of enjoyment, righteousness, or responsibility* toward the practices. People may learn of secure technologies and practices (and how they learn these is an interesting question), but exposure alone will not induce change in their habits. These three factors – access to resources, knowledge of security threats and defenses, and perceived benefit from increased security – all contribute to the successful adoption of security practices. Reviewing my research in terms of the influence of these factors helps illustrate the cases where secure practices are or are not successfully adopted.

5.1.1 Surveying Password Management Practices for Website Logins

Chapter 2 considered password management practices where people had very few passwords and everyone reused passwords. Participants had significant knowledge of security threats from poor passwords as well as what could increase their level of password security. They knew what constituted a strong, random password. Many also had relatively sophisticated models of attack. Participants saw that both technically savvy attackers as well as personally known attackers could compromise their accounts. They also considered motivation as a stronger impetus for attack than ability.

In terms of consumption and access to resources, very few participants used any external tools to help them manage their passwords, despite having password managers embedded in the very browsers used to visit websites. Additional resources for better password management could include password generators (for stronger password construction and for easing the burden of generation in lieu of reuse); enforcement of strong password choice by website administrators could also be a technical resource, but this mechanism as well as password generators do not directly address reuse, nor do they measure it. Therefore, there are limitations in addressing the problem of reuse via currently available technology, although this problem is tied to how often people (re-)generate their passwords.

A few participants cited avoiding reuse as a reason they opted for different passwords on different websites, but, as the numbers on reuse demonstrate, the perceived value of avoiding reuse is low: pitting convenience against security rarely results in security prevailing on a regular, habitual basis. People have their habits grounded in convenient password management strategies and though they often know another way to improve the level of security achieved, they are basically "locked-in" to what they already have. Since they already have passwords set on certain accounts, the more they use these passwords, the easier these passwords become to remember. Convincing users to change from three passwords reused over nine websites to nine passwords used across these same accounts is a nontrivial overhaul. Furthermore, it remains unclear whether they would see any clear benefit to such changes.

5.1.2 Barriers to Adopting Encryption for Group Communication

Peer-influence is another way for people to become aware of the benefits associated with more security practices. This bears some similarity to Becker's depictions of how more experienced users disseminate their experiences to novice and occasional users. The study in Chapter 3 on the adoption of encrypted e-mail most clearly demonstrates this diffusion of technology; *there was no self-discovery of more secure practices.* People learned how to use the systems from each other and justified their precautions as necessary under specific circumstances.

Knowledge of security threats became a predominant motive for seeking encryption software. The security threats stemmed from known incidents of infiltration of the organization to general wariness of government surveillance and included cooperation from technology developers, such as when someone from the PGP Corporation demonstrated the technology for the organization. However, since the non-profit remained focused on non-technical goals, their budget limited their available resources for consuming technology, and they had to rely on low-cost or free versions of encryption.[1] Likewise, limited access to technical support staff inhibited the adoption of encryption.

The interesting cases featured circumstances that adopted encryption versus circumstances that ignored encryption. The perceived value of encryption was tied to a litmus test of whether or not the data contained organizational secrets. Focusing only on confidentiality of secrets, without worrying about traffic obfuscation, lowered the value of regular, habitual use of the technology. Special circumstances where confidential data required protection necessitated the use of encryption and encouraged people to depart from their normal practices.

The method of introducing the technology also influenced the perceived value of the software. Adoption of encrypted e-mail in ActivistCorp indicated using this technology was, as one participant described, "viral." It required people to act as disseminators of the technology rather than waiting for individuals to discover it on their own. In a similar way, Brown and Duguid suggest that interactions with colleagues help circulate technical information in the context of work in a manner that is more easily absorbed than other types of training (their book describes the apprenticeship of technical representatives that repaired Xerox copiers by observing more experienced people in the situation [17]). My research shows that interaction between colleagues confirmed the use of encryption software as a worthwhile practice.

5.1.3 Redesigning Encryption for Group Discussions

Assuming that people learn about secure practices from their social interactions and begin to recognize the benefits of secure practices, the natural follow-up question for technology designers becomes whether technology design could influence or facilitate this interaction and dissemination of knowledge. I explored this issue when designing the EMBLEM system in Chapter 4. Rather than considering adoption through a single-user's independent discovery of the technology (which, given the current rarity of the technology as well as the presence of network externalities, seems unlikely), I expected that people would need an introduction to the technology.

The design of the EMBLEM system tried to reduce the burden of knowledge dissemination of the technology. For example, I expected that the creator of an EMBLEM group would act as

[1] Wariness of Trojan horses might also also encourage organizations to avoid proprietary software.

a security delegate, wary of surveillance and desiring to protect secret discussions, rather than requiring many people to be knowledgeable of security threats. I assumed that a delegate could pass this knowledge to other members of the group (or that information about security threats and defenses could be abstracted for novices); furthermore, this delegate could reduce the necessity of having available technical support staff, a concern in the adoption of encrypted e-mail at ActivistCorp.

My assumption that a delegate (or even that the software itself through visualization) could disseminate knowledge of security threats and defenses remains problematic for the design and reduces the perceived value of the system. Users who are unaware of encryption software for e-mail will require more than a simple introduction to the EMBLEM system. They need more background on the kind of protection it offers and the reasons behind it. On the other hand, users who are familiar with encryption software may continue to be wary of systems that weaken currently available end-to-end protection.

EMBLEM's design primarily accounted for multiple types of users with differing levels of experience with encryption for communications. While EMBLEM's design may or may not be the best approach, the general issue of developing security technologies *assuming group adoption* is an compelling area for future work. Furthermore, as Chapter 4 explains, the subfield of CSCW may provide a basis for developing these technologies in the future.

5.2 Security Practices and Symbolic Interactionism

My work also contributes to security system development by analyzing why users consciously reject more secure practices. People are wary of appearing paranoid, especially in front of their colleagues. Encouraging people to change their habits from less to more secure practices would correspond to encouraging people to increase displays of paranoia. Thus, encouraging this change in security practices requires a change in the perception of how desirable these practices are. Becker's description of marijuana users revealed that people needed to detect the effects of using the drug as well as to learn to perceive these effects as desirable (in contrast to non-users' perception of these effects as undesirable or self-discovery of these effects without an expectation of deriving pleasure from them). For security systems, we need to address the stigma non-users attach to more secure practices, making sure that users execute practices correctly and that users perceive benefit from correct execution. Becker indicates that this perception of benefit will come from the influence of peers, but HCI-Sec researchers have so far failed to capitalize on this influence.

The HCI-Sec community needs to appreciate that people's practices are guided, at least in part, by the meaning people attach to them. Infusing meaning into practices is analyzed from the sociological perspective of *symbolic interactionism.* Blumer explains the three premises of perspective of symbolic interactionism (emphasis added) [12]:

> The first premise is that *human beings act towards things on the basis of the meanings that the things have for them.* Such things include everything that the human being may note in his world—physical objects, such as trees or chairs; other human beings, such as a mother or a store clerk; categories of human beings, such as friends or enemies; institutions, as a school or a government; guiding ideals, such as individual independence or honesty; activities of others, such as their commands or requests; and such situations as an individual encounters in his daily life. The second premise is that *the meaning of such things is derived from, or arises out of, the social interaction that one has with one's fellows.* The third premise is that *these meanings are handled in, and modified through, an interpretive process used by the person in dealing with the things he encounters.*

In short, people inject meaning into things and actions through their interactions with others and interpret their own actions based on such interactions. Becker's work on deviant behavior and the development of deviant careers offer examples of symbolic interactionist perspective. In the case of computer security, we need to account for the attributions of meaning and the interactions between people when we attempt to restructure currently deviant behavior into desirable habits.

As Boak argued, computer security tends to encourage almost myopic focus on the technology developments without much regard for the people that need to deal with problems created by following organization protocols and adopting the designated technologies. Issues concerning deployment are, however, actions infused with meaning, and the development of security goals must account for the fundamentally human and social nature of technology adoption. It remains too easy to focus on targets that will not substantially increase the real-world security achieved – novel technologies and cryptography protocols or usable technologies may solve problems unrelated or orthogonal to the real adoption of better security practices.

My thesis addresses the human side of computer security by realizing that the people who deal with security systems are frequently in a position to undermine or foster successful deployments. The password study results indicate that the gains from knowing what constitutes good practices are frequently compromised by lack of knowledge about the power of attackers, especially in the case of the huge number of attack attempts generally used in dictionary attacks. The ActivistCorp study results indicate that a visible security technology will be limited to the context that people associate with heightened security. The EMBLEM work reiterates the necessity for designers to match the goals of the security system with the needs and concerns of its potential user population.

The cases of security use in the real world demonstrate how security system deployments fundamentally rely on their audiences. More broadly, computer security work in the future ought to investigate instances where people have adopted security systems. In addition, while much talk focuses on security models in terms of risk and potential threats, it would be beneficial to develop more sophisticated models of users in terms of the social interactions that encourage technology adoption. Lack of adoption is, in and of itself, a threat to security systems.

My approach may seem redundant to scholars in computer science. Small actions, such as using a piece of software for the first time, would likely not have such large negative connotations associated with them; however, Goffman's work on the presentation of self indicates that even minute details are important to an individual's construction of the impression one conveys to the outside world. He describes the impression as a "performance" than one acts in from of an "audience" [42]:

> Performers commonly attempt to exert a kind of synecdochic responsibility, making sure that as many as possible of the minor events in the performance, however instrumentally inconsequential these events may be, will occur in such a way as to convey either no impression or an impression that is compatible and consistent with the overall definition of the situation that is being fostered. ...as students of social life we have been less ready to appreciate that even sympathetic audiences can be momentarily disturbed, shocked, and weakened in their faith by the discovery of a picayune discrepancy in their impressions presented to them. Some of these minor accidents and "unmeant

> gestures" happen to be so aptly designed to give an impression that contradicts the one fostered by the performer that the audience cannot help but be startled from a proper degree of involvement in the interaction, even though the audience may realize that in the last analysis the discordant event is really meaningless and ought to be completely overlooked.

Goffman argues that one small misstep, one inadvertent flag into the inner paranoia of an individual, can unravel the otherwise normal facade that an individual casts to peers and colleagues. Goffman's advice to social scientists is nonetheless applicable to scholars in security and HCI-Sec domains: heed the meanings associated with actions, no matter how small. The stigma associated with being overly cautious, taking the most secure path possible and sacrificing personal convenience for security gains, may discourage, in a single interaction, these exact practices. Recognizing this stigma and working to relieve the barriers it presents may be one of the more fruitful areas of work in the future.

5.3 Implications for Future Work

My thesis discusses the issues facing real users in their interactions with security systems and protocols. While many researchers are interested in the failures of already deployed protocols, technologies, and systems, few researchers consider analyzing these systems in terms of their deployment and adoption by large groups of users. Many new project designs abstract real world constraints, which prevents significant progress in the field.

For example, as mentioned in Chapter 2, many projects explore password authentication and alternatives to password authentication, but few examine how to introduce these systems in realistic settings. (There are, of course, practical resource constraints, for example access to participants and deploying untested systems to large organizations.) Yet, research settings rarely replicate the real problems that users face including, for instance, creating a password as a secondary task to a primary task of accessing a resource and recalling a password with recall interference (a task of correctly recalling a specific password when different passwords are used in different accounts). How to operationalize tasks that reflect the difficulties faced by users would be an productive area to explore in HCI-Sec.[2] In addition, future work could focus on re-creating and testing deployment issues to investigate these problems realistically in conjunction with work on creating new alternative systems.

One possible venue for investigation considers "training wheels" versions of security interfaces. Carroll and Carrithers pioneered work in HCI that demonstrated that "training environments" could reduce learning time for complex software applications. When users are overwhelmed with choices, the notion of limiting user actions – while coming at the cost of reduced feature sets – actually guides users more quickly to their tasks with fewer errors (and specifically fewer unrecoverable errors). While EMBLEM remains an imperfect design, it partially addresses the problem of introducing new users to security systems without overwhelming them with cryptographic details. Work on scaffolding and training wheels environments, and more generally, the area of computer-supported learning may help guide future work in developing adoptable systems.

While other fields, such as computer networks research, have had to consider backwards compatibility issues and incremental deployment in a realistic setting, these same ideas have not been applied to users in computer security. Each new security system is considered in a bubble of its own, which, though practical for researchers, leaves a serious gap between research and real life. Pragmatically, bringing these fields together will likely have gains for both sides. What is created in research but does not see the light of day in real systems can deaden the field's innovation—why work on theoretically "solved" problems? That is not to say that what is created now must be successfully adopted tomorrow. Naturally a lag between research and real life exists, but current security threats (such as phishing, malware, and botnets) spark interest in the research community and drive new solutions. Working on current deployment and adoption issues broadens the lens of interest beyond computer security threats alone. Deployment issues are perhaps more closely an area of market research right now, but the research community could learn more about systems that have succeeded or failed in real organizations; my work in Chapter 3 reflects a blending of the two approaches.

Gaining access to participants as well as an entry into organizations concerned with real security problems, however, remains problematic for future work investigating security in the wild. Understandably, there are some organizations, such as governmental agencies and military sec-

[2] Such research could draw on the experience of social psychologists who have had to deal with constructing experiments of this kind.

tors where security issues are so paramount that they forbid access to those conducting openly published research. Nevertheless, other sectors, such as health systems, remain relatively open since they deal with security and privacy with more public scrutiny. While those in security related fields consider more discreet populations than HCI researchers generally analyze, they have common experiences with HCI researchers in participant access issues.

Recent interest in security and human behavior, as demonstrated by the workshop organized by Anderson and Schneier [3], reflects prospects for future collaborations between HCI-Sec researchers and other social scientists. In particular, economists have often analyzed individual decision-making processes, and their models of human behavior could help frame future research on the adoptability of software systems. Specifically, I have discussed network externality constraints on the adoption of software systems, but the applications of prospect theory to computer security may prove more helpful for understanding and modifying security decision-making processes in the future.

Chances are people are not able to accurately estimate the gains from more secure practices versus the risks from less secure practices. Economists Kahneman and Tversky argue that people tend to overestimate prospective gains and underestimate prospective losses in their work on prospect theory for analyzing decision making [56]. While this theory usually applies to decision making for financial risk, the general idea is applicable to the adoption of secure practices, such as password management practices. People can realize two possible gains for their management of passwords: convenience or increased security. Arguably, people realize *no gain at all* when they avoid reuse and construct randomized, hard-to-guess passwords. Few, if any, users will ever know if their practices have thwarted an attacker or prevented a break-in. Even if people could weigh the benefits from a thwarted attack very highly in their estimations of expected utility, the prospect has no value. The only influences on the estimation of expected utility, then, become convenience and risk (where people usually underestimate losses). Since convenience would outweigh predicted losses from compromised accounts it is unsurprising that few people see much value in improving their password management strategies.

My work does not consider economists' theories of collective action, though I believe it will be an essential consideration for future work on 'group-friendly' security software. Economists such as Mancur Olson have analyzed how groups form and act in a cooperative interest [71].[3] Specifically, the voting paradox applies to adoption of secure technologies and practices: while the group receives a benefit collectively, the individual benefit is tiny in comparison to the cost incurred against the individual actor. This is particularly interesting because organizational security inherently relies on collective action: most individuals sense no loss from a security breach but must do work for the sake of the group (as attackers choose the path of least resistance), and, furthermore, the level of security achieved relies on all actors acting in the best interest of the organization. Future work in the field could apply collective action theories, particularly the impetuses for collective action, to encourage successful adoption of security practices.

Finally, the thematic focus this thesis has drawn on sociology, particularly symbolic interactionism. Work in the field of HCI has regularly drawn on sociological theory such as Erving Goffman's *Presentation of Self*, but HCI-Sec tends to analyze user practices in terms of psychology (quantitative analysis and evaluation within a laboratory setting) rather than sociology [42]. Sociology's consideration of individuals in a social context appears to be more helpful when considering the real life deployment of security systems. How users reflect on their actions, how they appropriate technology, and how they apprentice others to use technology are no less important than metrics of task completion and errors rates.

Future work could also apply sociological organizational theory, such as mimetic isomorphism, to further analysis from individual actors to organizations. Mimetic isomorphism describes how, when uncertain about a decision, organizations try to mimic the decisions of other organizations they perceive as successful. Adoption of technology and the development of security protocols within an organization may reflect the standard adopted by a larger group of organizations. Comparing the use of technology in different organizations highlights how technology design and organizational culture can intersect and evolve for successful deployment of security systems.[4]

My thesis has focused on how security technologies and practices are adopted in realistic settings. The broader problem of security system deployment and adoption is a well known problem but one that has not seen much enthusiasm in the research community. Although researchers may find new ideas for alternative security systems far more seductive, I argue that analyzing successful

[3] For a contrasting view, see Garrett Hardin's tragedy of the commons, which speculates on how actors acting in their individual interest can undermine their collective interest. [51].

[4] Le Dantec and Edwards provide an example of cross-organizational field work on the use of non-security technology [65].

and failed security system deployments may also be inspirational for researchers. Furthermore, several fields in the social sciences such as social psychology, economics, and sociology can provide useful concepts as well as theories for framing research projects. Applying theories of collective action or social interactionism to computer security is an atypical approach, but one that has potential to ground design and evaluation in more realistic estimations of adoptability.

5.4 Closing Remarks

This thesis explored the development of secure habits and how ideal practices are often not realized in real life. I explored the areas of password authentication and encrypted e-mail, two mechanisms that can prevent message disclosure. My study of the adoption of encrypted e-mail in an activist group was the first of its kind to examine how adoptability, rather than pure usability, influences the uptake of a security technology. My work on password management practices similarly considered security from real users' perspectives. The study was the first to show how users weighed threats from outsiders, gauging both ability to attack and motivation to attack. It also measured their passwords based on real use of online accounts.

I based my investigations on the idea that changes in secure practices are developing career contingencies, moving from older, normalized habits to newer, deviant ones. Understanding the impetus, or generating an impetus, for these changes is an area for future work; the contribution of this thesis lies in framing views of people's practices in terms of their interactions with others and their self-presentation. I have tried to encompass these considerations in a new system for online discussions (EMBLEM), but more work on system design with considerations of adoptability remains an area for future work.

The desire to adopt more secure habits derives from an individual's learned realization of their benefits; researchers need to foster this value in the systems they design. However, it is important to remember that this value can be negative, such as when people associate more secure practice with being paranoid. This stigmatization of adopting security technologies or more secure practices discourages the very practices that we strive to encourage. Future work in the field could further investigate security in the wild, particularly from the perspective of group adoption of security technologies and protocols.

Appendix A

Survey Questions for Chapter 2 Study

A.1 Quantifying Password Reuse

The first part of the task provided a list of 139 websites in twelve categories. Websites were collected from the researcher's web surfing histories. Additionally, the sites were collected from results by searching "login", "password", and "username" in Google. Below the list is an example of the instructions for the second part of the task where participants tried to login to their indicated websites. The third part of the task asked participants to count their passwords and the instructions for this task are also shown. Finally, the fourth part of the task asked about what makes passwords strong.

A.1.1 Part 1: Selecting Websites

Participants were instructed as follows:

> Please answer as accurately as possible.
>
> Do you have an account on the following websites? Place a check next the website name.

Shopping	Services	Travel
1-800-Flowers.com	iVillage.com	Alaska Airlines
Amazon.com	CampusFood.com	American Airlines
Barnes and Noble	Comcast	Amtrak
BizRate	Food Network	Continental Airlines
Blue Nile	Keen.com	Expedia
Buy.com	Match.com	Extended Stay America
CVS/pharmacy	Meetup	United Airlines
Crate and Barrel	Monster.com	Mariott Hotels
E-Bay	Net-Temps.com	NJ Transit
Epinions.com	Opera.com: My Opera	Orbitz.com
Flower.com	Phone Shark	Priceline.com
Fogdog Sports	RelayHealth.com	Student Universe
IKEA	Sprint.com: My PCS	Travelocity
Kmart	T-Mobile	Hilton Hotels or Hampton Inn
Office Max	TV Shows on DVD	
Overstock.com	UniversalClass.com	
Starbucks	Verizon	
Target	WebMD	
The Sports Authority	Working For Change	
Walmart	Microsoft Passport [Hotmail, MSN, MSN Messenger, MSN Money, and MSN Encarta]	

Journals & Magazines	Computers & Internet	Entertainment	Finance
Discover Magazine	AvantGo	Audible.com	Bank of America
HighWire.com	Backflip.com	GamesMania	CBS Marketwatch
Magazines.com	Dell Computers	MTV.com	E-Trade
Ms Magazine	ExtremeTech.com	MovieTickets.com	Financial Times
PC Magazine	HP Computers	Netflix	Fleet
People Magazine	IBM Computers	RPG.net	Merrill Lynch Online
Reader's Digest	InterVideo.com	TicketsClub.com	Morningstar.com
Scholastic.com	O'Reilly Books	Tower Records	PNC
Scientific American Digital	Slashdot	Wizards of the Coast	Princeton Credit Union
Science Magazine	SourceForge	iTunes/Apple Store	Wells Fargo
Z Magazine			

Sports	Shopping: Clothes	Reference
CBS Sportsline Fantasy Sports	Abercrombie & Fitch	AccuWeather.com
WhatIfSports.com	Old Navy	Books 24x7
NBA	Nordstrom's	Wikipedia
ESPN.com: My ESPN	GAP	Traffic Pulse
MLB	J Crew	Weather.com
ESPN Fantasy Games	American Eagle Outfitters	Interlibrary Loan
NFL.com Fantasy	Banana Republic	
NHL Chat	Macy's	
Baseball Mogul	Victoria's Secret	

Communication	News
Blogger	Women's eNews
Coollist.com	LA Times
Friendster	NY Times
GMail	Newsday.com
Live Journal	Salon.com
Lycos Mail	Washington Post
Mail.com	Guardian (UK Newspaper)
MyWay.com	
Orkut	
Princeton Webmail	
QuickTopic.com	
Yahoo	
ScreenName.com [AOL Netscape	
AIM Compuserve]	

A.1.2 Part 2: Logging into Websites

Participant instructions for the login task. The example shows the case where the participant has indicated he or she uses Gmail. This task is repeated for all other websites the user indicated they sued.

> Login to the Gmail website. Use the link below which opens a new browser window. You will have 90 seconds to try to login to the website. When you have finished, close the GMail window to return to this webpage.
>
> `https://gmail.google.com`
>
> Were you able to login to the website on your first attempt? [Yes, No]
>
> If had a successful login, please write down your password for the GMail website.
>
> If you were not able to login successfully, what are some reasons why you believe you were not successful?
>
> - Couldn't access the website

◇ Forgot my username for this website

◇ Forgot my password for this website

◇ Forgot my e-mail address for this website

◇ Made a typographical error in entering the login data

◇ Other (please specify) [*Free form response.*]

If participants were unsuccessful in logging into a website within 90 seconds, they were shown the following instructions (similar to the self-report instructions).

> The 90 seconds has passed. Before we move on to the next task, could you please select some reasons why it was difficult to login successfully in 90 seconds for this website?

◇ Couldn't access the website

◇ Forgot my username for this website

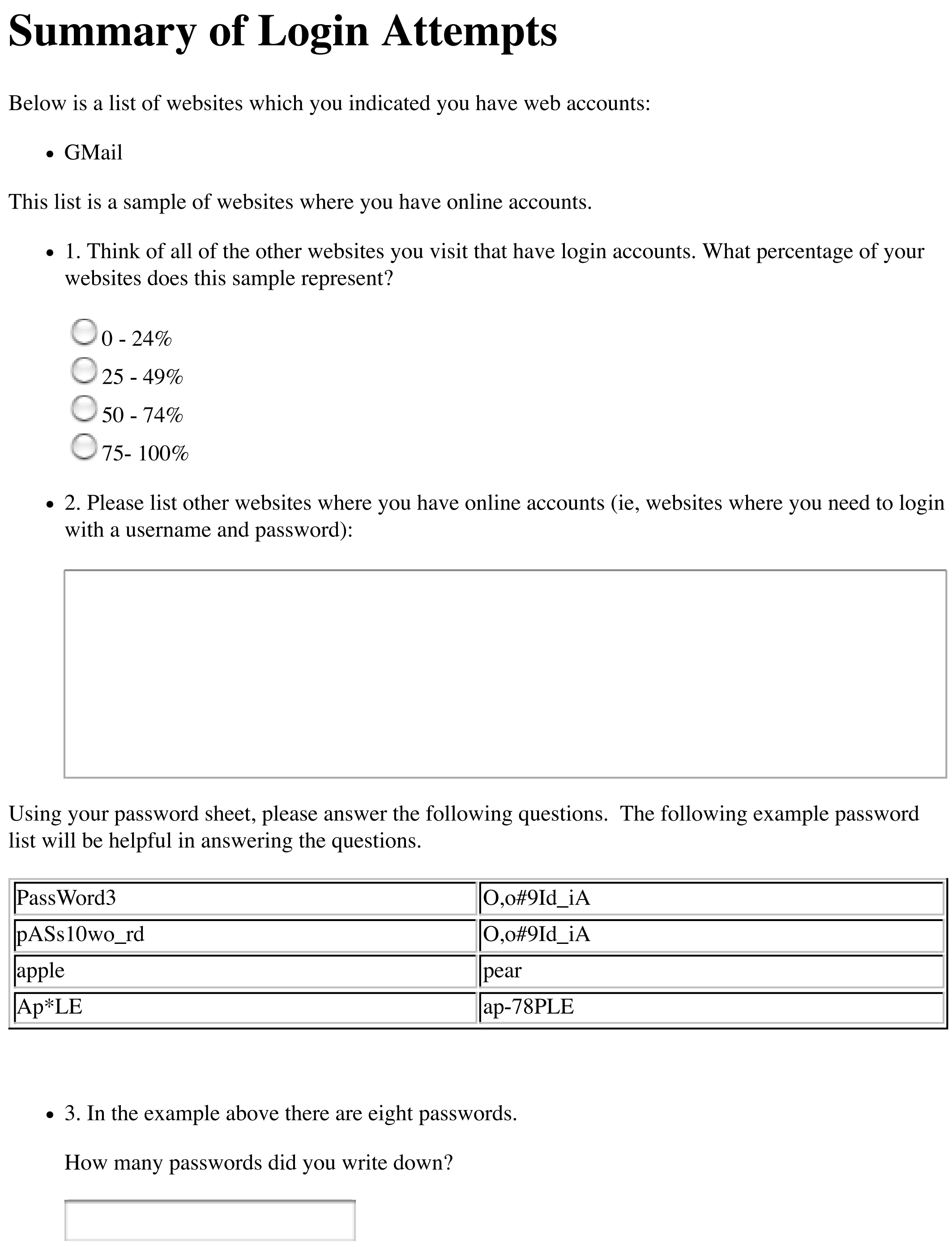
◇ Forgot my password for this website

◇ Forgot my e-mail address for this website

◇ Made a typographical error in entering the login data

◇ Other (please specify) [*Free form response.*]

A.1.3 Part 3: Counting passwords



Summary of Login Attempts

Below is a list of websites which you indicated you have web accounts:

- GMail

This list is a sample of websites where you have online accounts.

- 1. Think of all of the other websites you visit that have login accounts. What percentage of your websites does this sample represent?

  ○ 0 - 24%
  ○ 25 - 49%
  ○ 50 - 74%
  ○ 75- 100%

- 2. Please list other websites where you have online accounts (ie, websites where you need to login with a username and password):

Using your password sheet, please answer the following questions. The following example password list will be helpful in answering the questions.

PassWord3	O,o#9Id_iA
pASs10wo_rd	O,o#9Id_iA
apple	pear
Ap*LE	ap-78PLE

- 3. In the example above there are eight passwords.

  How many passwords did you write down?

- 4a. Permutations of phrases share common sub-phrases. Permutation does not refer to a shared meaning between phrases; for example, "School" is NOT a permutation of "Princeton". You can create a permutation of a phrase in five ways:

1. Repeating a previously used phrase (nothing changes)	"Princeton" and "Princeton"
2. Changing the capitalization of letters within a phrase	"Princeton", "princeton", "prINcEtoN"
3. Adding numbers or characters to a phrase	"Princeton9","A-Princetonian", "PrincetonPrinceton", "AP~rinceton3"
4. Removing numbers or characters from a phrase	"Pton", "Printon", "Prince"
5. Combining some of the previous methods	"pRince#ton", "pr(INce)ton3", "P*in73t0n"

In the example at the top of the page, there are four FAMILIES of passwords ("apple", "PassWord3", "O,o#9Id_iA", "pear"):

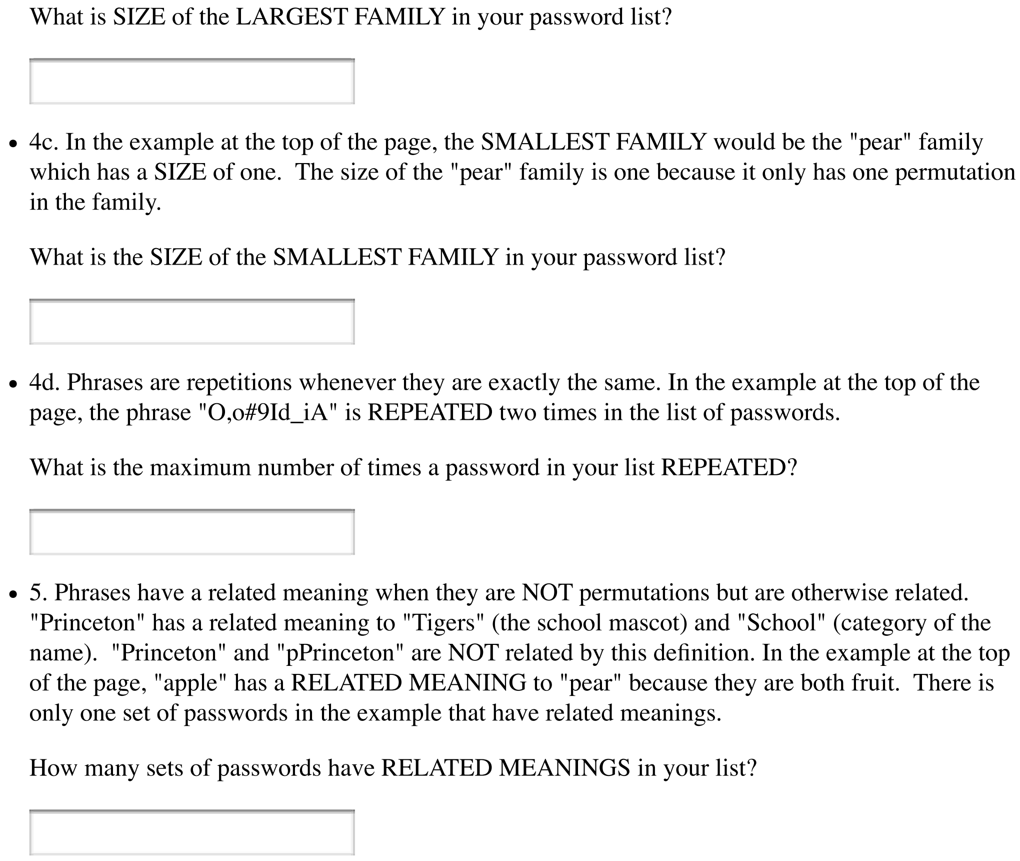
FAMILY	PERMUTATIONS
"apple"	"apple", "Ap*LE", and "ap-78PLE"
"PassWord3"	"PassWord3" and "pASs10wo_rd"
"O,o#9Id_iA"	two identical entries of "O,o#9Id_iA"
"pear"	"pear"

Note that the number of families is equal to the number of unique passwords (four, in the example).

How many FAMILIES of passwords did you write down?

- 4b. In the example at the top of the page, the LARGEST FAMILY would be the "apple" family which has a SIZE of three. The size of the "apple" family is three because it has three permutations in the family.

FAMILY	PERMUTATIONS	SIZE
"apple"	"apple", "Ap*LE", and "ap-78PLE"	3
"PassWord3"	"PassWord3" and "pASs10wo_rd"	2
"O,o#9Id_iA"	two identical entries of "O,o#9Id_iA"	2
"pear"	"pear"	1



What is SIZE of the LARGEST FAMILY in your password list?

- 4c. In the example at the top of the page, the SMALLEST FAMILY would be the "pear" family which has a SIZE of one. The size of the "pear" family is one because it only has one permutation in the family.

  What is the SIZE of the SMALLEST FAMILY in your password list?

- 4d. Phrases are repetitions whenever they are exactly the same. In the example at the top of the page, the phrase "O,o#9Id_iA" is REPEATED two times in the list of passwords.

  What is the maximum number of times a password in your list REPEATED?

- 5. Phrases have a related meaning when they are NOT permutations but are otherwise related. "Princeton" has a related meaning to "Tigers" (the school mascot) and "School" (category of the name). "Princeton" and "pPrinceton" are NOT related by this definition. In the example at the top of the page, "apple" has a RELATED MEANING to "pear" because they are both fruit. There is only one set of passwords in the example that have related meanings.

  How many sets of passwords have RELATED MEANINGS in your list?

Participants were asked to repeat questions 3 - 5 with the following instructions.

> The previous questions may have missed some of the passwords you use. Now write down all of your other passwords that you can recall and answer the questions below. Please use any tools (memory, e-mail archives, pieces of paper, etc.) that will help you recall your passwords.
>
> Using your password sheet, please answer the following questions. The following example password list will be helpful in answering the questions.

A.1.4 Part 4 Constructing Strong Passwords

In the fourth part of task, participants were asked about what they considered to be a strong passwords. They were given the following instruction:

> Many websites have tips and rules for creating strong passwords. Pretend your friend

> Eve Jones (evjones@princeton.edu) is also a student at Princeton and she is having trouble understanding these rules. For each rule or tip, she's provided three example passwords with an explanation of how she created her password. Help her learn what makes a strong password by ranking her examples from strongest to weakest and explaining your ranking.

After each rule, participants were asked to rank the password options from "Most Secure" to "Least Secure" and had a free form space for an explanation. Again, overall all the passwords were relatively weak as they were all short passwords constructed with very little randomization.

1. Use a password of at least six characters.

   HomerSimpson I concatenated the first and last name one of the characters in my favorite TV show.

   snyfe I took the first or last letter of words at the end of paragraphs in an excerpt from the Undergraduate Announcement: (s = interests, n = information, y = year, f = field, e = education).

   evjones03 This is my username with the number '03' appended to the end.

2. Use uppercase and lowercase letters in the password.

   Buster It's my dog's name.

   poTion I took this word out of my favorite novel and capitalized the 3rd letter in the word.

   kwcmyd I selected lowercase letters from different areas of the keyboard.

3. Create an acronym from an uncommon phrase.

   MadeNChina I saw "Made in China" on the bottom of my mug and substituted the capital letter 'N' for the word "In" and removed all spaces from the phrase.

   Goa7DbFits Since today had lots of snow outside, I thought of this phrase: "Go out after 7:00. Don't bother FREEZING in the snow.". Next, I took the first letter of each word.

   MSFTWMTMCD I took the acronyms from some stocks that I own (MSFT = Microsoft, WMT = Walmart, MCD = McDonald's Corporation).

4. Mix up two or more separate words.

   PrincetonNJ I took my address and concatenated the city and state names, abbreviating New Jersey to NJ.

   0moDtAhDer0 I took the word "mother" and put the word "DAD" inside. I then appended and prepended zeros.

   garbageball When my friends and I try to toss trash, we pretend it's basketball and call this game "garbageball" taking the word "basketball" and replacing "basket" with "garbage".

5. Drop letters from a familiar phrase.

   fur-hldrvhc I read the newspaper and took the phrase "four-wheel drive vehicle". I removed the letters in the following order: o, w, e, e, i, e, v, e, i, l, e.

   IiauaiooIai I took the chorus from the Destiny's Child song "Soldier" and used the vowels only from the line "If his status ain't hood, I aint..."

   ThDlyPrinctnn I took out all of the vowels from and spaces from the name of the school newspaper, "The Daily Princetonian".

6. Avoid common literary names.

   rodnoffirG I took this name out of a Harry Potter book (Griffondor). I then reversed the word.

   Brklyn1234 I visited Brooklyn today, so I took the name of this borough. Next, I dropped the vowels (Brklyn). Finally, I appended the string '1234'.

   5DI4cn0pSm I took the last name of the author Charles Dickens. Next, I dropped 'k' and 'e' (Dicns). Then, I capitalized 'I' and 'S' (DIcnS). Finally, I inserted the current time (5:40 pm).

7. Avoid abbreviations of common phrases or acronyms.

   FSPMGLGA I took the first letter of names of characters from Lord of the Rings (F = Frodo, S = Sam, P = Pippin, M = Merry, G = Gandalf, L = Legolas, G = Gimli, and A = Aragorn).

imho&lol I used the abbreviation of the phrases “in my humble opinion” and “laughing out loud” from instant messaging.

dYs∼oWS$ First, I took the last letter from words in the phrase “read my lips, no new taxes” (dysows). Next, I capitalized ’Y’, ’W’, and ’S’. Finally, I inserted some punctuation (∼ and $).

8. Use homonyms or deliberate misspellings.

   whinnyDaPooh I took the name of a cartoon character (Winnie the Pooh) and replaced the word Winnie with similar sounding whinny. Next, I replace “the” with “Da”.

   urmysunshine First, I used the phrase, “you are my sunshine”. I substituted the word “you” for ’u’. Next, I replaced “are” with ’r’.

   drowssapseve I spelled ”eve’s password” backwards. I then removed the apostrophe.

9. Avoid passwords that contain your login ID.

   ejones I took my login ID (evjones) and dropped one letter (v).

   felten I took the login ID of someone else at the university.

   vo7ne\ I took my login ID (evjones) and dropped the ’e’ (vojones). Next, I substituted ’n’ with 7 (vo7nes) and ’s’ with ‘\’.

10. Use numbers in the password.

    2581796 That’s my office telephone number.

    d5ri7ve1 I just drove from the airport, so I thought of the word ’drive’. I then tried to put some digits (5, 7) inside the word.

    2gether I took a number that sounds like a word (‘two’ sounds like ‘to’) and thought of a word that incorporates this sound (“together”).

11. Use punctuation in the password.

    01/12/85 This is my birthday.

    sk?inniest I took a word from my fashion magazine and added a question mark (‘?’) after the 2nd letter in the word.

!nter$tate I took a word off of a map and substituted letters with similar looking punctuation characters, 'i' looks like '!' and 's' looks like '$'.

A.2 User Priorities

Password management survey. Possible responses for multiple choice questions shown in square brackets.

1. Please enter your participant ID
2. Are you a student? [Yes, No, I do not wish to answer]
   (a) If YES: What is your class year? [Freshman, Sophomore, Junior, Senior, Graduate Student (Please specify how many years of study)]
   (b) Students: What is your major?; Faculty/Staff: What department or office are you affiliated with?
3. What is your gender?
4. How old are you?
5. How would you rate your experience with computers? [1 = Novice, 2, 3 = Average, 4, 5 = Expert]
   (a) When you need a software program, do who usually installs the software on your computer? Examples: you, a sibling, technical staff for your department.

Are there two websites where you use the SAME password? [Yes, No, I do not wish to answer.]

If YES: Why do these websites have the same password?

If NO: Why do all websites have different passwords?

The following statements relate to reasons why you might use the same password for different online accounts. In your experience with online web accounts, indicate the extent to which you agree and disagree with the statements. [Strongly Disagree, Slightly Disagree, Neither Agree nor Disagree, Slightly Agree, Strongly Agree, Refrain from Answering]

1. I reuse a password if it is unimportant to me
2. It is easier to reuse a password than create a new one
3. If I reuse a password, it is easier for me to remember it
4. I use the same password for all websites I use
5. I do not see a need to create a different password websites that require logins when I access information (online newspapers, online discussion groups, etc)
6. It is unrealistic for me to use different passwords for different websites
7. I do not understand why I would want to use a different password for different websites
8. Only nerds or paranoid people think they need more than one password
9. I only need one password because no one knows it
10. I reuse a password when there isn't much financial information (bank account, credit card number, etc) about me on a website
11. I reuse a password when there isn't much personal information (sexual orientation, health status, etc) about me on a website
12. I reuse a password when I use a website for routine communication (e-mail, chat, etc)
13. I reuse a password when I don't think anyone would be interested in the information related to me on a website
14. A hacker couldn't do much damage to me if they compromised my password
15. A hacker couldn't do much damage to anyone else if they compromised my password
16. No matter how good my passwords are or how many passwords I have, a determined attacker would compromise my passwords anyway
17. No one would want to attack one of my website accounts
18. A hacker would choose someone more important than me when attacking an online account.

19. Since many other people have accounts on the same websites I use, a hacker is unlikely to attack one of my accounts.

20. Please add any other reasons you have for using a the same password for different websites [*Free form response.*]

Are there two websites where you use DIFFERENT passwords?

If YES: Why do these websites have different passwords?

If NO: Why do all websites have the same password?

The following statements relate to reasons why you might use different passwords for different online accounts. In your experience with online web accounts, indicate the extent to which you agree and disagree with the statements. [Strongly Disagree, Slightly Disagree, Neither Agree nor Disagree, Slightly Agree, Strongly Agree, Refrain from Answering]

1. I use different passwords for different accounts because I am worried that someone might steal one or more of my passwords

2. I have heard it is more secure to use different passwords for different websites

3. It is harder to guess my password if I use different passwords on different websites

4. It is harder to gather information about me if I use different passwords on different websites

5. I wasn't allowed to use one of my passwords because it was too easy to guess

6. I wasn't allowed to use one of my passwords because it wasn't in the correct format (too long, too short, did or didn't have numbers, did or didn't have punctuation, did or didn't use capitals, etc)

7. If a password is generated for me by a website than I use this password instead of one of my normal passwords

8. It is easy for me to use different passwords for different websites because I use a password generator to create my passwords

9. It is easy to remember different passwords for different websites because I only use a few websites

10. I use unrelated passwords for unrelated websites. For example: all websites related to cars use one password and all websites related to tennis use another password

11. I have different passwords for different security levels of websites. For example, I have a generic password for online newspapers but I have a special password for my online bank account

12. I have forgotten one of my original password and had to reset it to something different

13. Please add any other reasons you have for using different passwords for different websites [*Free form response.*]

There are many ways a password can be compromised. The following are scenarios where someone could compromise a user's password:

- A hacker could use a software program to extract passwords.
- A roommate could read a written list of passwords.
- A stalker could guess a password after learning some personal facts, like a pet's name or a social security number.
- A passerby could observe a password as it is typed.

1. IGNORING THEIR MOTIVATION, for each person listed below, write at least one scenario where this person could obtain one of your passwords. [*Free form responses.*]
   - Someone you know well (significant other, friend, relative, etc)
   - Someone without computer expertise that you have met (classmate, neighbor, etc)
   - Someone with computer expertise that you have met (computer support person, tech savvy friends, etc)
   - Someone from your organization that you do not know (your school, your club, etc)

- Someone from a competiting organization that you do not know (school rival, employer's competitor, etc)
- Someone that is unaffiliated that you do not know (hacker, etc)

2. IGNORING MOTIVATION AND CONSIDERING ABILITY ONLY, rank people by their ability to access information without permission from one of your web accounts.

   - 1 - (Greatest ability) [*Options for ranking:* Someone you know well, Someone without computer expertise that you have met, Someone with computer expertise that you have met, Someone from your organization that you do not know, Someone from a competing organization that you do not know]
   - 2 [*Options for ranking*]
   - 3 [*Options for ranking*]
   - 4 [*Options for ranking*]
   - 5 [*Options for ranking*]
   - 6 - (Least ability) [*Options for ranking*]
   - Please add any comments about your ranking [*Free form response.*]

3. IGNORING ABILITY AND CONSIDERING MOTIVATION ONLY, rank people by their motivation to access information without permission from one of your web accounts.

   - 1 - (Most motivated) [*Options for ranking:* Someone you know well, Someone without computer expertise that you have met, Someone with computer expertise that you have met, Someone from your organization that you do not know, Someone from a competing organization that you do not know]
   - 2 [*Options for ranking*]
   - 3 [*Options for ranking*]
   - 4 [*Options for ranking*]
   - 5 [*Options for ranking*]
   - 6 - (Least motivated) [*Options for ranking*]

- Please add any comments about your ranking [*Free form response.*]

4. CONSIDERING MOTIVATION AND ABILITY, rank people by their likeliness to access information without permission one of your web accounts.

    - 1 - (Most likely) [*Options for ranking:* Someone you know well, Someone without computer expertise that you have met, Someone with computer expertise that you have met, Someone from your organization that you do not know, Someone from a competing organization that you do not know]
    - 2 [*Options for ranking*]
    - 3 [*Options for ranking*]
    - 4 [*Options for ranking*]
    - 5 [*Options for ranking*]
    - 6 - (Least likely) [*Options for ranking*]
    - Please add any comments about your ranking [*Free form response.*]

Some people need help to remember their passwords. They may write down their passwords on a piece of paper, have their web browser store their passwords, or save their passwords in a file. What kinds tools do you use? Check all that apply (examples for each type in parenthesis) [*Choices shown as following enumerated list*]

◇ Password Software (Password Agent, Password Tracker, Any Password)

◇ Password Website (Gator eWallet, PasswordSafe.com, RoboForm.com)

◇ Website Cookies (Website checkbox: "Remember my password on this computer")

◇ Web Browser (Internet Explorer AutoComplete, Netscape Password Manager, Firefox Saved Passwords)

◇ Computer File (Word document, Excel sheet, text file)

◇ Paper (post-it notes, notebook, day planner)

◇ Password Reminder (Website feature: "Forgot your password?")

◇ Human Memory

◇ Other [*Free form response.*]

What kinds tools have you used in the past but no longer use? Check all that apply (examples for each type in parenthesis) [*Choices shown as following enumerated list*]

◇ Password Software (Password Agent, Password Tracker, Any Password)

◇ Password Website (Gator eWallet, PasswordSafe.com, RoboForm.com)

◇ Website Cookies (Website checkbox: "Remember my password on this computer")

◇ Web Browser (Internet Explorer AutoComplete, Netscape Password Manager, Firefox Saved Passwords)

◇ Computer File (Word document, Excel sheet, text file)

◇ Paper (post-it notes, notebook, day planner)

◇ Password Reminder (Website feature: "Forgot your password?")

◇ Human Memory

◇ Other [*Free form response.*]

What information do you write down or store to help you remember your passwords for websites? Check all that apply. [*Choices shown as following enumerated list*]

◇ Username

◇ Website name (eg, Princeton University)

◇ Website address/URL (eg, www.princeton.edu)

◇ Password

◇ Clue (Word or phrase related to the password)

◇ E-mail address

◇ Secret question (eg, what's your mother's maiden name?)

- ◇ Answer to a secret question
- ◇ Other [*Free form response.*]

- When was the last time you created a new password for logging into a website? [Less than 1 week, More than 1 week, Less than 1 month, More than 1 month, Less than 6 months, More than 6 months, Less than 1 year, More than 1 year]
- How did you create a new password?
- What are other methods you have used to create a new password?
- Where did you store your new password?
- What are other methods you have used to store passwords?

The following statements relate to reasons why you use the same password without changing it. In your experience with online web accounts, indicate the extent to which you agree and disagree with the statements. [Strongly Disagree, Slightly Disagree, Neither Agree nor Disagree, Slightly Agree, Strongly Agree, Refrain from Answering]

1. If I haven't used some passwords for very long, I don't need to change it
2. If I infrequently use a password, I don't need to change it
3. I don't change a password because I wouldn't be able to remember a new password
4. I try to use the same password for multiple websites, so it would be inconvenient to change the password
5. I only reset the password after I've forgotten it
6. The process for changing a password is bothersome, difficult, or unintuitive
7. It is unrealistic for me to periodically create new passwords
8. I will change a password if have been asked to change it
9. I will only change a password if I know the password has been compromised

10. If I am not protecting anything with my password, it doesn't matter if I change it or not

11. I don't have a reason to change the password on the websites I use

12. There's never a need to change passwords for online accounts

13. Please add any other reasons you have for keeping, changing, or resetting passwords [*Free form response.*]

Appendix B

Interview Schedule for Chapter 3 Study

1. RESPONSIBILITIES AND TECHNICAL BACKGROUND

    - Tell me a little about yourself and how you arrived at your present position.
    - What have you done for work to day? What do you plan to do? Is this reflective of what you typically work on?
    - What do you need a computer for when you work?

2. PERCEPTION OF ENCRYPTED E-MAIL

    - What benefit did you think there was in using encrypted e-mail?

        (a) For yourself?

        (b) For those you work with?

        (c) For your organization?

    - What problems are there with using encrypted mail?

        (a) For yourself?

        (b) For those your work with?

        (c) For your organization?

- In your experience, what's a situation where someone would want to use encrypted e-mail?
- In your experience, what's a situation where someone would not want to use encrypted e-mail?
- Have you been asked/have you requested someone to use encrypted e-mail?
  (a) What was the motivation for this request?
  (b) What was the reaction to the request?
- Many people do not use encrypted e-mail.
  (a) How would you encourage someone to use it?
  (b) What would you do to make it easier for them to use encrypted e-mail?

3. UNDERSTANDING ENCRYPTED E-MAIL

- How did you learn about PGP?
- What do you think it does for you?
- Are there alternative to PGP?
- How did you get setup for encrypted mail?
- Have you encrypted an e-mail message to someone? Why?
  (a) (If no) Why not?
  (b) (If no) Have you tried to send an encrypted message?
  (c) (If no) What happened?
  (d) (If no) Who was involved?

4. CHOOSING TO USE ENCRYPTED MAIL (If relevant)

- Describe the last time you used encrypted mail.
- How did you convince the recipient to use encrypted mail?
- What was the recipients reaction? (Problems/Benefits)
- How did the recipient get setup for encrypted mail?
- Why did you think you need encryption?

5. CHOOSING NOT TO USE ENCRYPTED MAIL

    - When you sent e-mail to me, you did not encrypt the message. Why not?
    - (Lawyer only): At the end of your message, you have a message:

      > This message contains information that is privileged and confidential attorney-client communication and attorney work product. No unauthorized use, distribution or disclosure of any information contained herein is permitted.

      Why do you use this?
    - What other situations do you use unencrypted e-mail?
    - What is your reasoning about why unencrypted e-mail is OK in these cases?
    - Speculate on how you would use encryption in this scenario. What would have to be different?

6. AVOIDING ENCRYPTED MAIL (If relevant)

    - Have you ever sent a message w/o encryption but think it would have been better to use it?
    - Why did you think it would have been better to encrypt?
    - Who do you think you would be discouraging if you encrypted the message?
    - Did you take an alternative precaution?
    - How did you justify your decision not to use encryption?

7. OTHER (If relevant)

    - Digital signatures are also related to encrypted e-mail. Have you had any experience with digital signatures?
    - Walkthrough sending messages
        (a) Could you walk me through sending an encrypted message to me? If not, why not?
        (b) What do you need from me?
        (c) What do you need to give me?
    - What other questions do you think I should be asking people about the practice of using encrypted e-mail?

Appendix C

EMBLEM Implementation Details

I implemented the EMBLEM server as a Tomcat 6 servlet instance. Authentication is handled as a Java class that implements CSRF nonce checking (see Zeller and Felten [104]) and also implements password authentication. The group membership manager is a simple Java module with synchronized writes to an XML file. EMBLEM uses a configuration file that specifies the incoming and outgoing mail servers. The implementation supports SMTP and either IMAP or POP3 mail servers, using the JavaMail library. I exploit the multiple account mapping addressing scheme to have different EMBLEM groups' mail sent to the same account, so the mail server polling service also has to filter messages for a particular group on each request. This simplifies the configuration for creating groups and also allows users to overload their existing mail accounts with encrypted mailing lists.

The client is a Google Web Toolkit 1.6 web application for AJAX (Asynchronous JavaScript and XML) interactivity with web users. The toolkit compiles Java source code into Javascript. The web application communicates back to the Tomcat servlet over an SSL connection. A JSON (JavaScript Object Notation) library helps with generating and parsing JSON-formatted responses from the server to web application clients. Visual effects, like the message fade, are executed through the GWT Library 0.2.0 with Scriptaculous 1.8.1.

The server uses several libraries to handle both web application requests and cryptographic operations. I deployed EMBLEM on an Ubuntu Hardy Heron (8.0.4) server, which was actually virtually hosted through Slicehost. The server presents a StartCom Ltd. X.509 certificate for SSL connections.[1] I also use GPG 1.4.7 for the cryptographic engine and perform the cryptographic operations through a GnuPG Made Easy (GPGME) 1.1.4 library wrapped with a Java Native Interface (JNI) module. For ease of display, I use ASCII armored output instead of binary output. E-mail attachment uploading and downloading from web applications to the server are handled with help from the Apache libraries Commons FileUpload 1.2.1 and Commons IO 1.4.

[1] On reflection, I should have purchased an SSL certificate from a known issuer. The unknown issuer for the SSL certificate creates a few extraneous popup windows in the browser, warning users that the certificate authority is untrusted. IE7 will completely block access until the user click the red, "Continue to this website (not recommended)." Firefox 3 similarly blocks SSL connections to websites that present certificates from unknown issuers. This can be avoided if the EMBLEM server presents a certificate signed by a recognized certificate authority.

Bibliography

[1] A. Adams and M. A. Sasse. Users are not the enemy. *Commun. ACM*, 42(12):40–46, 1999.

[2] American Civil Liberties Union. The foreign intelligence surveillance act - news and resources. `http://www.aclu.org//safefree/general/17321res20030408.html`. Accessed 4 November 2008.

[3] R. J. Anderson, B. Schneier, A. Acquisti, and G. Loewenstein. Interdisciplinary workshop on security and human behaviour. `http://www.cl.cam.ac.uk/~rja14/shb08.html`, 2008.

[4] P. M. Aoki and A. Woodruff. Making space for stories: ambiguity in the design of personal communication systems. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 181–190, New York, NY, USA, 2005. ACM.

[5] L. Baker. Facebook ban makes British MP doubt his existence, December 2007.

[6] D. Balfanz. Usable access control for the World Wide Web. In *Proc. of ACSAC*, pages 406–415, Las Vegas, NV, December 2003. IEEE.

[7] D. Balfanz, G. Durfee, R. E. Grinter, D. Smetters, and P. Stewart. Network-in-a-box: How to set up a secure wireless network in under a minute. In *Proc. of the 13th USENIX Security Symposium*, pages 207–222, San Diego, CA, August 2004. USENIX.

[8] S. R. Barley. Careers, identities, and institutions: the legacy of the Chicago School of Sociology. In M. Arthur, T. Hall, and B. Lawrence, editors, *The Handbook of Career Theory*, pages 41–65. Cambridge University Press, Cambridge, England, 1989.

[9] L. Bauer, S. Garriss, J. M. McCune, M. K. Reiter, J. Rouse, and P. Rutenbar. Device-enabled authorization in the Grey system. In *Information Security: 8th International Conference, ISC 2005*, volume 3650 of *Lecture Notes in Computer Science*, pages 431–445, Sept. 2005.

[10] H. S. Becker. *Outsiders: Studies in the Sociology of Deviance.* The Free Press, Collier-Macmillan Limited, 1963.

[11] K. Beznosov, M. E. Zurko, S. Chan, and G. Conti. Usability of security administration vs. usability of end-user security. SOUPS 2005 Conference Panel, 2005.

[12] H. Blumer. *Symbolic Interactionism: Perspective and Method*, page 2. University of California Press, Berkeley, CA, 1969.

[13] D. G. Boak. A history of U.S. communications security (Volumes I and II); the David G. Boak lectures, National Security Agency (NSA). `http://www.governmentattic.org/2docs/Hist_US_COMSEC_Boak_NSA_1973.pdf`, 1973. Accessed 16 February 2009.

[14] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229, London, UK, 2001. Springer-Verlag.

[15] T. Bridis. Hacker impersonated palin, stole e-mail password. *Associated Press*, September 18, 2008.

[16] A. S. Brown, E. Bracken, S. Zoccoli, and K. Douglas. Generating and remembering passwords. *Applied Cognitive Psychology*, 18(6):641–651, 2004.

[17] J. S. Brown and P. Duguid. *The Social Life of Information.* Harvard Business School Press, Boston, MA, USA, 2002.

[18] BugMeNot.com. Frequently asked questions. `http://bugmenot.com/faq.php`. Accessed 7 February 2009.

[19] M. Burawoy, editor. *Ethnography Unbound: Power and Resistance in the Modern Metropolis.* University of California Press, 1991.

[20] J. M. Carroll and C. Carrithers. Training wheels in a user interface. *Commun. ACM*, 27(8):800–806, 1984.

[21] Charity Navigator. How do we rate charities? `http://www.charitynavigator.org/index.cfm?bay=content.view&cpid=35`, 2006. Accessed 17 Nov 2008.

[22] L. Church. End user security: The democratisation of security usability. Security and Human Behaviour, `http://www.lukechurch.net/Professional/Publications/SHB-2008.pdf`, 2008.

[23] J. R. Crandall, D. Zinn, M. Byrd, E. Barr, and R. East. ConceptDoppler: a weather tracker for internet censorship. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 352–365, New York, NY, USA, 2007. ACM.

[24] L. F. Cranor and S. Garfinkel. Guest editors' introduction: Secure or usable? *IEEE Security & Privacy Magazine*, 2(5):16–18, 2004.

[25] L. F. Cranor and S. Garfinkel, editors. *Security and Usability: Designing Secure Systems That People Can Use.* O'Reilly, 2005.

[26] P. A. David. Clio and the economics of QWERTY. *American Economic Review*, 75(2):332–37, May 1985.

[27] R. Dhamija and A. Perrig. Dejà vu: A user study using images for authentication. In *Proc. of the 9th USENIX Security Symposium*, 2000.

[28] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM Press.

[29] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590, New York, NY, USA, 2006. ACM.

[30] P. Dourish. Implications for design. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 541–550, New York, NY, USA, 2006. ACM.

[31] P. Dourish, E. Grinter, J. D. de la Flor, and M. Joseph. Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal Ubiquitous Comput.*, 8(6):391–401, 2004.

[32] W. K. Edwards, M. W. Newman, J. Z. Sedivy, T. F. Smith, D. Balfanz, D. K. Smetters, H. C. Wong, and S. Izadi. Using speakeasy for ad hoc peer-to-peer collaboration. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, New Orleans, LA, November 2002.

[33] W. K. Edwards, E. Shehan, and J. Stoll. Security automation considered harmful? In *NSPW '07: Proceedings of the 2007 workshop on New security paradigms*, New York, NY, USA, 2007. ACM.

[34] N. Ferguson and B. Schneier. *Practical Cryptography.* John Wiley & Sons, Inc., New York, NY, USA, 2003.

[35] E. Gabber, P. B. Gibbons, Y. Matias, and A. J. Mayer. How to make personalized web browising simple, secure, and anonymous. In *FC '97: Proc. of the First International Conference on Financial Cryptography*, pages 17–32, London, UK, 1997. Springer-Verlag.

[36] S. L. Garfinkel. *PGP: Pretty Good Privacy.* O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1996.

[37] S. L. Garfinkel, D. Margrave, J. I. Schiller, E. Nordlander, and R. C. Miller. How to make secure email easier to use. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 701–710, New York, NY, USA, 2005. ACM Press.

[38] S. L. Garfinkel and R. C. Miller. Johnny 2: a user test of key continuity management with S/MIME and Outlook Express. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 13–24, New York, NY, USA, 2005. ACM Press.

[39] S. Gaw and E. W. Felten. Password management strategies for online accounts. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 44–55, New York, NY, USA, 2006. ACM.

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

[41] J. Gideon, L. Cranor, S. Egelman, and A. Acquisti. Power strips, prophylactics, and privacy, oh my! In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 133–144, New York, NY, USA, 2006. ACM.

[42] E. Goffman. *The Presentation of Self in Everyday Life*, page 51. Anchor, Doubleday, New York, 1959.

[43] N. S. Good, R. Dhamija, J. Grossklags, D. Thaw, S. Aronowitz, D. Mulligan, and J. Konstan. Stopping spyware at the gate: a user study of privacy, notice and spyware. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 43–52, New York, NY, USA, 2005. ACM Press.

[44] N. S. Good, J. Grossklags, D. K. Mulligan, and J. A. Konstan. Noticing notice: a large-scale experiment on the timing of software license agreements. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 607–616, New York, NY, USA, 2007. ACM.

[45] N. S. Good and A. Krekelberg. Usability and privacy: a study of kazaa p2p file-sharing. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 137–144. ACM Press, 2003.

[46] Google. Privacy overview - Google privacy center. `http://www.google.com/intl/en/privacy_highlights.html`, 2008. Accessed 17 Nov 2008.

[47] R. Grinter and M. Eldridge. Wan2tlk?: everyday text messaging. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 441–448, New York, NY, USA, 2003. ACM.

[48] J. Grudin. Why cscw applications fail: problems in the design and evaluationof organizational interfaces. In *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 85–93, New York, NY, USA, 1988. ACM.

[49] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: Cold boot attacks on encryption keys. In *Proc. 17th USENIX Security Symposium*, 2008.

[50] S. Hansell. Bank loses tapes of records of 1.2 million with visa cards. *The New York Times*, page A9, February 26 2005.

[51] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.

[52] Hepting v AT&T. 439 F. Supp. 2d 974 (N.D. Cal., 2006).

[53] Hush Communications Corp. Hush encryption engine white paper. `https://www.hushmail.com/public_documents/Hush%20Encryption%20Engine%20White%20Paper.pdf`. Accessed 21 Nov 2008.

[54] Hush Communications Corp. Webmail using the hush encryption engine. `https://www.hushmail.com/public_documents/Webmail%20Using%20the%20Hush%20Encryption%20Engine.pdf`. Accessed 21 Nov 2008.

[55] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, 2007.

[56] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–91, 1979.

[57] A. H. Karp. Site-specific passwords. Technical report, Hewlett-Packard Laboratories. `http://www.hpl.hp.com/personal/Alan_Karp/site_password/site_password_files/site_password.pdf`.

[58] M. L. Katz and C. Shapiro. Network externalities, competition, and compatibility. *American Economic Review*, 75(3):424–40, 1985.

[59] Katz, Michael L. and Shapiro, Carl. Technology adoption in the presence of network externalities. *The Journal of Political Economy*, 94(4):822–841, 1986.

[60] J. J. Kaye, J. Vertesi, S. Avery, A. Dafoe, S. David, L. Onaga, I. Rosero, and T. Pinch. To have and to hold: exploring the personal archive. In *CHI '06: Proceedings of the SIGCHI*

conference on Human Factors in computing systems, pages 275–284, New York, NY, USA, 2006. ACM.

[61] D. Kirk, A. Sellen, C. Rother, and K. Wood. Understanding photowork. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 761–770, New York, NY, USA, 2006. ACM.

[62] D. Klein. Succumbing to the dark side of the force - the Internet as viewed from an adult website. Notes from an Invited talk at USENIX, SANS, and O'Reilly's Open Source Conference. `http://www.klein.com/dvk/publications/darkside.pdf`, 1998. Accessed 18 Nov 2008.

[63] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 13–19, New York, NY, USA, 2007. ACM.

[64] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Protecting people from phishing: the design and evaluation of an embedded training email system. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 905–914, New York, NY, USA, 2007. ACM.

[65] C. A. Le Dantec and W. K. Edwards. The view from the trenches: organization, power, and technology at two nonprofit homeless outreach centers. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 589–598, New York, NY, USA, 2008. ACM.

[66] E. Lichtblau. Large volume of FBI files alarms some US activist groups. *New York Times*, page A.12, July 2005.

[67] May First. How can I use gpg to both encrypt my email and prove my identity? `https://support.mayfirst.org/wiki/gpg`, May 2008. Accessed 17 Nov 2008.

[68] L. I. Millett, B. Friedman, and E. Felten. Cookies and web browser design: toward realizing informed consent online. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 46–52, New York, NY, USA, 2001. ACM Press.

[69] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM.

[70] D. A. Norman. *The design of everyday things*. Doubleday, 1990.

[71] M. Olson. *The logic of collective action; public goods and the theory of groups*. Harvard University Press, Cambridge, Mass., 1971.

[72] H. Petrie. Password clues. `http://www.centralnic.com/news/research`. Accessed 2 May 2005.

[73] PGP Corporation. *PGP Universal 2.0 – Technical Overview*.

[74] F. Pratt. *Secret and urgent; the story of codes and ciphers*. The Bobbs-Merrill Company, 1939.

[75] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: beyond human-computer interaction*. John Wiley And Sons Inc., 2002.

[76] Princeton Office of Information Technology. Password practices to avoid. `http://www.princeton.edu/itsecurity/basics/passwords/bad-pw-practices/`. Accessed 7 February 2009.

[77] Reporters without Borders. Handbook for bloggers and cyber-dissidents. `http://www.rsf.org/IMG/pdf/handbook_bloggers_cyberdissidents-GB.pdf`, September 2005. Accessed 17 Nov 2008.

[78] J. Ridgeway. Exclusive: Cops and former secret service agents ran black ops on green groups. `http://www.motherjones.com/news/feature/2008/04/firm-spied-on-environmental-groups.html`, April 2008. Accessed 17 Nov 2008.

[79] S. Riley. Password security: What users know and what they actually do. `http://psychology.wichita.edu/surl/usabilitynews/81/Passwords.htm`, February 2006.

[80] J. Rode, C. Johansson, P. DiGioia, R. S. Filho, K. Nies, D. H. Nguyen, J. Ren, P. Dourish, and D. Redmiles. Seeing further: extending visualization as a basis for usable security. In

SOUPS '06: Proceedings of the second symposium on Usable privacy and security, pages 145–155, New York, NY, USA, 2006. ACM.

[81] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. *14th Usenix Security Symposium*, 2005.

[82] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. *Science*, 311(5762):854–856, 2006.

[83] M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link' a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–131, 2001.

[84] M. A. Sasse and I. Flechais. *Security and Usability: Designing Secure Systems That People Can Use*, chapter 2. O'Reilly, 2005. Eds. Lorrie Faith Cranor and Simson Garfinkel.

[85] B. Schneier. *Secrets and Lies : Digital Security in a Networked World.* Wiley Computer Publishing, New York, NY, 2004.

[86] R. Singel. Yahoo outed chinese dissident knowing investigation was political, documents show – updated. *Wired.com*, July 31 1997. Accessed 28 September 2008.

[87] R. Singel. Encrypted e-mail company hushmail spills to feds. `http://blog.wired.com/27bstroke6/2007/11/encrypted-e-mai.html`, November 2007. Accessed 28 February 2008.

[88] S. Singh, A. Cabraal, C. Demosthenous, G. Astbrink, and M. Furlong. Password sharing: implications for security design based on social practice. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 895–904, New York, NY, USA, 2007. ACM.

[89] D. K. Smetters and R. E. Grinter. Moving from the design of usable security technologies to the design of useful secure applications. In *Proc. of the 2002 workshop on New security paradigms*, pages 82–89. ACM Press, 2002.

[90] Smyth v. Pillsbury. 914 F. Supp. 97, 100-01 (E.D. Pa. 1996).

[91] Student Computing Initiative. SCI: Frequently asked questions. http://www.princeton.edu/sci/help/faq.htm#eligible, May 2007. Accessed 11 Nov 2008.

[92] A. S. Taylor and L. Swan. Artful systems in the home. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 641–650, New York, NY, USA, 2005. ACM.

[93] United States Computer Emergency Readiness Team. Home network security. http://www.us-cert.gov/reading_room/home-network-security/#IV, December 2001. Accessed 28 October 2008.

[94] United States v. Nicodemo S. Scarfo. United States Court of Appeals for the Third Circuit. 263 F.3d 80 (U.S. App. 2001).

[95] D. Weirich and M. A. Sasse. Persuasive password security. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 139–140, New York, NY, USA, 2001. ACM Press.

[96] D. Weirich and M. A. Sasse. Pretty good persuasion: a first step towards effective password security in the real world. In *Proc. of NSPW 2001*, pages 137–143, New York, NY, USA, 2001. ACM Press.

[97] J. J. White. E-mail@work.com: Employer monitoring of employee e-mail. *Alabama Law Review*, 1997. 48 Ala. L. Rev. 1079.

[98] A. Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proc. of 8th USENIX Security Symposium*, pages 169–184. USENIX, 1999.

[99] A. Woodruff, S. Augustin, and B. Foucault. Sabbath day home automation:"it's like mixing technology and religion". In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 527–536, New York, NY, USA, 2007. ACM.

[100] L. Wright. The spymaster; a reporter at large. *The New Yorker*, 83(44):42, January 21 2008.

[101] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610, New York, NY, USA, 2006. ACM.

[102] K.-P. Yee. Aligning security and usability. *IEEE Security and Privacy*, 2(5):48–55, 2004.

[103] K.-P. Yee and K. Sitaker. Passpet: convenient password management and phishing protection. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 32–43, New York, NY, USA, 2006. ACM.

[104] W. Zeller and E. W. Felten. Popular websites vulnerable to cross-site request forgery attacks. `http://www.freedom-to-tinker.com/blog/wzeller/popular-websites-vulnerable-cross-site-request-forgery-attacks`, September 2008. Accessed 29 October 2008.

[105] P. Zimmerman. *The official PGP user's guide*. MIT Press, 1995.