Privacy Preserving Web-Based Email

Kevin Butler, William Enck, Jennifer Plasterr,
Patrick Traynor, and Patrick McDaniel

Systems and Internet Infrastructure Security Laboratory
The Pennsylvania State University
University Park, PA 16802 USA
{butler,enck,plasterr,traynor,mcdaniel}@cse.psu.edu

Abstract. Recent web-based applications offer users free service in exchange for access to personal communication, such as on-line email services and instant messaging. The inspection and retention of user communication is generally intended to enable targeted marketing. However, unless specifically stated otherwise by the collecting service's privacy policy, such records have an indefinite lifetime and may be later used or sold without restriction. In this paper, we show that it is possible to protect a user's privacy from these risks by exploiting mutually oblivious, competing communication channels. We create virtual channels over online services (e.g., Google's Gmail, Microsoft's Hotmail) through which messages and cryptographic keys are delivered. The message recipient uses a shared secret to identify the shares and ultimately recover the original plaintext. In so doing, we create a wired "spread-spectrum" mechanism for protecting the privacy of web-based communication. We discuss the design and implementation of our open-source Java applet, Aquinas, and consider ways that the myriad of communication channels present on the Internet can be exploited to preserve privacy.

1 Introduction

Internet users hemorrhage personal information. Almost every interaction on the web is scanned (directly or indirectly) by some party other than those directly involved in the transaction. Tracking cookies, web bugs, and other tools are used by advertisers to follow users as they move from site to site across the Internet [21]. Less scrupulous groups rely upon spyware to surreptitiously acquire personal information. Such information can be warehoused, collated with other sources, and stored indefinitely.

Recently, however, a more active means of collecting personal information has become common: users expose their personal communications to service providers in exchange for free online applications such as email and instant messaging. As promoted, access to this information allows online providers to personalize the user experience by offering targeted advertisements [16]. The revenue generated by connecting users and vendors has historically fueled much of the growth of the Internet, and is the major source of revenue for many websites. Hence, user profiling is an often positive and possibly necessary element of online life.

A. Bagchi and V. Atluri (Eds.): ICISS 2006, LNCS 4332, pp. 116–131, 2006.

However, the communications provided by users of these new services such as free email can be used to develop profiles that extend far beyond simply online habits. By allowing these services to scan the contents of every message that passes through their system, they provide commercial interests with insight into their daily and sometimes highly personal lives. The contents of such communications are further susceptible to interception and examination by repressive regimes [1,19]. Such practices are becoming the norm in web-based applications. Unfortunately, the legal devices for protecting user privacy against abuse or sale of this information are few, and those that do exist are often ineffective [7].

We assert that users need not sacrifice their right to privacy in exchange for any service. Just as customers of the postal service have come to expect that their messages will only be read by the intended recipient, so too should users of web-based services be guaranteed privacy in their communications. We demonstrate that strong confidentiality is attainable for all Internet users, regardless of the privacy policy of these online services.

In this paper, we introduce *Aquinas*, an open source tool designed to provide email privacy while maintaining plausible deniability against the existence of the unobservable (covert) communication. The Aquinas client provides privacy using a hybrid scheme; we employ cryptography to secure communication, steganography to hide the existence and substance of ciphertext, and multipath delivery to ensure compromised accounts or intercepted messages provide little information to an adversary. All email messages are initially encrypted and protected with a *message authentication code* (MAC) to ensure confidentiality and integrity. The key and ciphertext are then carefully divided into *shares*. The shares are embedded in emails using steganographic tools and sent to the recipient via multiple email accounts established at competing services such as Yahoo! Mail, Gmail, and Hotmail. When the recipient receives the ciphertext and key shares, Aquinas reconstructs the key and ciphertext. The ciphertext is decrypted and the contents validated to obtain the plaintext message.

Aquinas is an open-source Java applet. While the mechanisms and distributed nature of content delivery make the current iteration of Aquinas highly robust against multi-party collusion and third-party scanning, it is our intention to allow anyone to contribute additional algorithms and functionality to the codebase. This diversity of operation means that ultimately, the ability of any entity to detect or prevent private communications through web-based email services will be severely curtailed.

Through its use of multiple channels for message delivery, Aquinas's design mimics wireless "spread-spectrum" protocols, which use a pseudo-random pattern of radio channels in order to prevent eavesdropping and jamming. Even with the observation of some subset of channels, an adversary gains no usable information about the true nature of a message's contents. In Aquinas, an adversary needs to intercept email on all used mail accounts to gain *any* information about the user communication. Because no web-service can feasibly intercept all communication, user profiling is not possible.

Aquinas differs significantly from existing email privacy tools such as PGP [28]. Existing tools seek to secure the missives between known users typically using highly secure keys, i.e., public keys. Conversely, Aquinas seeks to enable mobile and lightweight communication; users need not have any physical data beyond a single password in their head. Moreover, Aquinas seeks to secure communication in environments where integration with existing tools is not available, e.g., free email accounts. That is not to say that Aquinas provides a superset of features of these tools. Specifically, Aquinas does not provide all the guarantees that other systems may, e.g., non-repudiation. Moreover, Aquinas is robust to compromise due to the generation of new keys for each message. We believe that this forward-security in combination with portability make this mechanism a highly attractive means of addressing the privacy.

The remainder of this paper is organized as follows: Section 2 gives an overview of our approach to solving these issues; Section 4 discusses additional issues facing the use of privacy preserving software; Section 3 examines the specifics of the our implementation; Section 5 examines the related work in this field; Section 6 offers concluding thoughts and future directions for this work.

2 Design

We first define the goals of Aquinas and consider the threats and adversaries we seek to protect against. The latter parts of this section describe the protections in Aquinas and the mechanisms for their implementation.

2.1 Goals

The high-level design goals of Aquinas include:

Confidentiality: No adversary should be able to obtain information about the existence or content of email communication.

Integrity: The integrity of all communication must also be preserved, i.e., any modification of the message should be detectable by the recipient.

Ease of use: Aquinas should not require that the user understand or directly use any sophisticated concepts such as cryptography or steganography. Additionally, the tool should provide a user experience consistent with traditional email applications.

The systemic requirements of Aquinas are somewhat more mundane. We do not want to place a requirement on the user for having to install software beyond a simple web browser, or to provide complex data, e.g., maintain keyrings. The implications of this are that all security-relevant data needed to receive email from a single user should be derivable from a password. The second implication is that the tool should be able to execute on arbitrary platforms.

In addition, we want to maximize the flexibility of the services that can be used; to that end, we wish to be able to easily integrate Aquinas with any communication service available on the Internet. Finally, we require the tool to be extensible in order to accommodate future functionality.

2.2 Threat Analysis

Users of web-based email services are subject to a variety of threats against their privacy and security. Below, we consider possible adversaries and the motivation and attacks they may employ.

Threats may arise from corporate adversaries. For the application providers that run web-based email services, there is a strong interest in profiling their users for revenue generation. Information about users can be sold to marketing agencies or directly to other companies interested in advertising products to their target demographics. The information gleaned about a user through profiling email can be arbitrarily detailed; through sufficiently optimized data-mining techniques, even users reticent to reveal personal information may unwittingly divulge many more personal details than they realize. If information is sent without any form of obfuscation, it is trivial for the adversary to intercept communications; any party between the user and the application provider will also have unfettered access to this information.

There are environments where protections such as message confidentiality may not be allowed: the email provider may disallow encrypted or unrecognizable content, or the network used for information transmission may have similar restrictions. Even when hidden channels are used, vulnerabilities may still be manifested. As information flows to and from an email account, the account will be subject to *channel decay* over time: an adversary collecting copies of the transferred information will be able to use the amassed data to more easily mount an attack against the channel. In addition, the probability of an adversary learning of a channel's existence will increase with time.

An additional adversary with a similar reward model to the application provider can be the webmail user's ISP. Defending against these attacks presents a tangential set of challenges. We consider adversarial ISPs in greater detail in section 2.5.

While the goals of adversarial companies are largely financially-based, political adversaries may represent a greater threat to some users. Repressive political states have shown little compunction about using Internet activity logs to target and persecute dissidents [1,19]. These adversaries can be significantly more determined to discover information about their target than businesses, and have full access to all records and logs of activity. We can consider the political adversary to have all of the same tools at their disposal as the corporate adversary, plus the ability to compel multiple application providers to turn over all information they possess, or force those companies into collusion. This could create very serious consequences for a dissident attempting to keep their communications hidden from a regime.

2.3 Email Protection

Figure 1 provides an overview of how email messages are protected by Aquinas. After a message is composed, the email is encrypted and steganographic techniques are applied to conceal the nature of the information being sent. We use *symmetric cryptography* as the encryption mechanism, in contrast to alternative

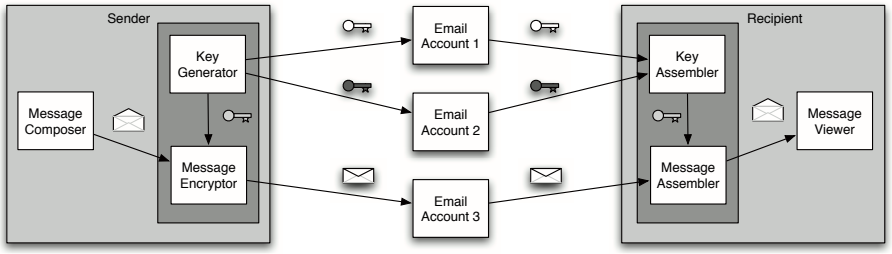

Fig. 1. A sample message and key delivery flow. The sender encrypts the plaintext and then embeds it into carefully select covertext using steganography. The message containing the hidden content is then sent as shares to one or more accounts owned by the recipient. Each of the key shares used to create the encryption key are then sent to different destination email accounts. The recipient's client checks all of the accounts, reassembles the key and ciphertext from the shares, and recovers the plaintext. The separate emails from different SMTP servers to prevent reassembly by adversaries.

email schemes, which use public-key cryptography. Because public-key systems require the use of a trusted third party for endorsing user identities, selecting parameters for key encryption, and proving credentials—a non-trivial problem that has not been entirely solved in a satisfactory manner [8]—as well as a full associated infrastructure, we found that this architecture would not fit within the goals of our system. While use of symmetric cryptography necessitates initial establishment of a shared secret (typically in an out-of-band fashion), we felt this was an adequate tradeoff.

Symmetric cryptography requires both the sender and recipient to agree on a key. Obviously, we do not want to send the key in the same email as the ciphertext. A simple solution is to send the key and ciphertext is separate emails, but if both are sent through the same mail service, the adversary still has access to both. The solution is to split both the key and ciphertext into multiple shares and send each part through multiple mail services.

The encryption process is straightforward. The sender begins by creating some number of keys. These keys are combined via XOR (herein noted as $\bigoplus$) to create the encryption key[1]. Using some symmetric cryptographic algorithm, e.g. AES, the message ciphertext is created. However, encryption alone is not sufficient protection for the email, as a service provider could easily detect that an encrypted message was sent. A sender may wish to plausibly deny that sensitive information has been transmitted, and the presence of ciphertext in a message alludes to the transmission of unknown information. To make the emails appear innocuous, the message and key shares are passed through a steganographic filter (e.g., SNOW [26]), obscuring the email with *covertext* that provides no insight as to the real message contents.

Once the message has been encrypted and protected with a MAC, it is steganographically obscured with covertext. The resulting message is sent in an email to

[1] The encryption key cannot be determined unless *all* of the key shares are known.

one of the recipient's accounts. The key shares are also hidden through steganographic techniques, and these messages are sent to different accounts. At this point, the message and key shares are distributed among multiple, independently administered email servers, and the message contents, mass collusion notwithstanding, are secured from unauthorized observers.

The recipient begins the decryption process by downloading both the message and key shares. From the recipient's point of view, the key to decrypt the message is the recipient email accounts. Once downloaded, the recipient applies the steganographic filter to eliminate the covertext and retrieve the ciphertext and key shares. The key shares are combined with $\bigoplus$ to create the decryption key, and the ciphertext is decrypted.

2.4 Design Detail

Our key distribution approach is an example of *multipath delivery*. This method leverages the distributed nature of Internet services to create and multiplex orthogonal channels in the form of multiple email accounts. An analogous means of communications, known as *spread spectrum*, has been used for more than fifty years. Given some range of radio spectrum with x discernible frequencies, messages are transmitted using some pseudorandom sequence of frequencies known only to sender and receiver. An adversary attempting to eavesdrop on communications has a probability $(1/x)^p$ of overhearing the entire message over p time periods. As x and p increase, the ability of an attacker to successfully intercept communications quickly approaches zero. The application of such a technique to the Web makes interception by an adversary an even more daunting task. While the radio spectrum arguably has a limited number of frequencies, the number of channels in which data can be injected into and across the Internet are arguably infinite. We demonstrate the use of Aquinas with key shares carried across multiple email addresses; however, with little additional extension, we can store key shares and messages in web log comments, chat rooms, newsgroups, and a variety of other locations. If we consider each of these particular channels equivalent to a different frequency in the spread-spectrum analogy, then we see the vast number of virtual frequencies afforded to us.

Each of these email accounts used to send the shares should be located at domains operated by different providers. This method of key delivery is robust to collusion for a number of reasons. Competition will deter collusion: any information about a user that a provider is able to garner or derive that is not known to the provider's competitors generates a competitive advantage. Because providers are competing for revenue from advertisers, having unique insights into customer profiles will be rewarded by allowing more targeted marketing to those users, making advertising more lucrative and profitable. Hence, providers desire to keep this information as private as possible, and colluding with other providers would necessitate providing information on the user. This creates a *competitive disincentive* for the provider to engage in collusion. Additionally, even if an adversary is to discover that a message is hidden within an email, they must still

recover all n key shares along with the message in order to decrypt it, making this system robust to the compromise of up to $n-1$ key shares.

The recipient, using the Aquinas client, checks her disparate *message* and *key* email accounts for shares. Aquinas downloads all of the messages and then searches through the headers for a flag identifying the keys for a specific message. Demultiplexing via the $\bigoplus$ operation is performed on all n key shares, providing the recipient with key K. The actual data contained within the email is then uncovered and decrypted using K. The real message from the sender is then displayed for the recipient.

The communication process is no more difficult from a user's standpoint than using a traditional mail program. Specifically, a user must enter the multiple outgoing (SMTP) and incoming (POP3) email servers that are to be used to deliver messages. With the *address book* feature in Aquinas, allowing storage of multiple users per email address, this information only needs to be entered once.

2.5 Adversarial ISPs

Many users rely on a single service provider to transit their information to the greater Internet. The consequence, however, is that this ISP has access to all of the information sent through its network. By implication, this means that all of the messages sent by the Aquinas user will pass through their home provider who can collect data, even though the destinations of these messages may be disparate email services providers.

Key management does not help in this case because all n channels are implicitly revealed. However, the user has recourse through use of the SSL protocol. SSL provides end-to-end data protection between the user and the email provider, making information unreadable to an ISP attempting to passively eavesdrop on messages. Aquinas supports the use of SSL in order to thwart the ISP threat. With SSL, however, there is some information leakage; the adversary can learn the destination of the packets (but not the destination of the email) by examining the IP header. Thus, while the content of the messages will be unknowable, the fact that information is being transferred to an email provider will be leaked. By observing this information, the ISP could learn all of the providers used and instantiate collusion with them. To hide evidence of the destination, the user could make use of proxies, such as anonymous remailers and other anonymous routing services [27,9]. Additionally, to lower the probability of an adversary detecting the existence of a channel formed by the email account, the user can periodically abandon their accounts and set up new ones for communication.

An alternative solution to the ISP threat exists that does not require the use of SSL between a user and their email provider. Security can be implemented through *chaffing and winnowing* [20] with email accounts. By including email accounts not used during the email communication, the adversarial ISP will have to choose the correct subset of accounts that correspond to a message. A brute-force approach based on combinatorics rapidly becomes infeasible for the adversary. For example, if the user transmits a message with 40 shares, but only

20 of those are used to construct the message, the adversary will be required to search through the $\binom{40}{20}$, or nearly 138 billion, combinations.

2.6 Key Negotiation and Management

Bootstrapping communication between users requires a mechanism outside of Aquinas to be used. Out-of-band key communication through methods such as speaking over the phone or meeting in person is possible; alternately, a mechanism such as PGP could be used for the initial setup. While the user would have to be on a trusted machine that has PGP installed to perform this transaction, once the initial key setup was complete, the user can then communicate using any terminal with the recipient.

We propose that a directory of users be stored in a publicly accessible repository. Each set of email addresses associated with a user can be stored within this space. The addresses can be public because it is their particular combination used for an email transmission that is the secret. Part of the initial communication between two users can include transmission of a shared secret between the two parties. This can be very simple, such as the word "dog". A permutation sequence can then be calculated by using this secret as a key. For example, AES-128 has a keyspace of 2^{128} entries. Encoding the secret as a value (e.g., converting "dog" in its decimal representation) allows us to use it as a key. If there are 40 email addresses associated with a user, the keyspace can be binned into 40 intervals, and the generated number will fall into one of these bins, generating one of the email addresses that will comprise the key share. The resulting value is then encrypted with the key and another interval is selected based on the new output. This process is repeated until there are 20 unique addresses selected. By negotiating a new secret (for example, through email communication), a new combination of addresses used as key shares can be selected. The following matrix illustrates the series of transformations that generates the values to be binned:

$$\begin{bmatrix} k_0 = h(\text{“}dog''\text{)} \\ k_1 = E(k_0, k_0) \\ k_2 = E(k_1, k_1) \\ \quad \vdots \\ k_{20} = E(k_{19}, k_{18}) \end{bmatrix}$$

Note that email is not the only method by which key and ciphertext can be delivered. The open functionality inherent to the Internet allows any means of sending data to become a covert channel for communication. A combination of keys placed in weblog referrer logs, instant messages, BitTorrent [2] and other P2P file sharing systems, streaming audio and video, newsgroup postings, and any number of disposable or community email accounts can be used to keep the contents of any message secret. This method of key and content distribution creates a wired "spread-spectrum" effect, effectively using servers across the Internet like unique "frequencies". This technique thereby obfuscates the

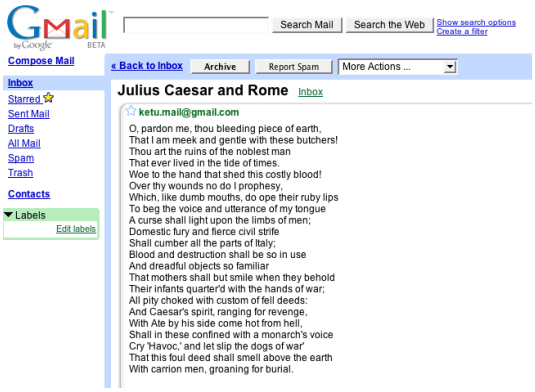

Fig. 2. A screenshot of the content of an email sent from Aquinas to a Gmail account

ability to determine that communication has occurred at all. Because of the sheer vastness of the web, the ability to prohibit privacy on this medium is *virtually impossible*.

3 Implementation

Aquinas is principally designed to support a simple and user-friendly interface. In order to retain the convenience of web-based email, Aquinas is required to be accessible via the Internet. Ideally, this portability should be machine independent to allow use by the widest possible community. For these reasons, we developed Aquinas using Java. Our goals, however, were not merely to allow use on their primary home or work machines (although this use is encouraged); rather, we wanted to ensure that users could protect their communications no matter where they were or what machine they were using, such as a terminal at an Internet cafe[2]. Accordingly, we have designed Aquinas to run as an applet. The Aquinas Java applet and source-code are freely available from:

`http://siis.cse.psu.edu/aquinas.html`

For reasons of space, the complete details of the implementation have been made available in the technical report [3], which is also available at the above address.

Figure 2 shows a screenshot of what the Gmail scanner sees as the content of an email sent using Aquinas. The plaintext of the message, however, is displayed in Figure 3. We performed extensive tests with emails protected by different steganographic covertexts, to determine how they would be handled by Gmail and other providers. While Gmail sometimes showed advertisements pertaining

[2] Note that users must still be cognizant of their surroundings and the machines they use if Aquinas is used in an untrusted location such as a remote kiosk. We cannot and do not protect against physical attacks such as keystroke loggers on remote terminals.

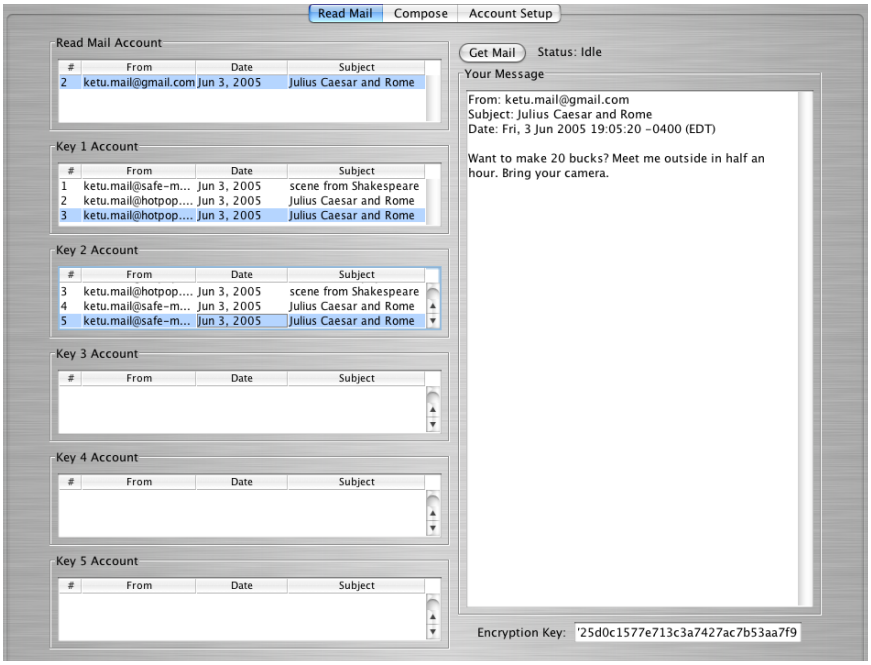

Fig. 3. A screenshot of the recovered plaintext of the email displayed in Figure 2

to the content of the covertext, none of these advertisements reflected the keywords or terms found in the plaintext message. This indicates to us that the real message transmitted stayed private and was protected from profiling.

4 Discussion

Aquinas extends the confidential nature of email by allowing message contents to remain secret until being read by the intended recipients thereby redefining the endpoint of web-based email as the user. Its portability, imperceptibility and forward-security through unique session keys make the use of Aquinas more attractive than many more traditional schemes. We therefore consider several issues of the secure use and implementation of Aquinas in the following subsections.

4.1 Preserving Privacy

Although the mechanisms discussed in this paper can provide security against profile generation and data mining, users of these solutions must still be cognizant of other privacy issues. Specifically, in spite of the use of encryption and steganography, it is still possible for information leakage to occur. The selection of cover text, for example, provides data that can be scanned and associated with a user. If a user were to select text from a website with radical political

statements or adult material, that information may still be affiliated with the user in spite of there being no actual relationship between the two parties in the real world. To mitigate this threat, we suggest using neutral text, such as the "Terms of Service" or "Frequently Asked Questions" pages available at the websites hosting the email. By doing this, a user exposes only the fact that they use a service (which is already known to the service provider).

The sender should also be aware of the paths that key shares take. For example, if all data were to cross a particular domain either during the sending or receiving process, all of the data necessary to create the keys for decryption would be readily available. It is therefore critical that users take advantage of as many unique channels as possible to provide maximum security.

Users should take additional precautions when deciding upon names for email accounts. While identically named accounts at a number of major free email providers would be easy for people to remember, they also increase the ease with which collusion between providers can occur. The tradeoff between ease and security must be carefully considered by each user. Much of this tradeoff can be mitigated by using the address book feature provided in Aquinas. As a standard security practice, the use of unique passwords across accounts in also highly recommended. In addition to providing robustness to a single compromise, the use of unique passwords also prevents one service provider from logging in to a user's account at another provider (i.e., unapproved collusion [11]). Simple methods to increase the security of password re-use include browser extensions such as those presented by Ross et al. [22].

The number of accounts used to achieve privacy can be set by the user and should be based upon their perceived threats. For example, someone simply wanting to avoid being profiled by free web-based email providers and advertisers may decide to rely upon two accounts. Because it is extremely unlikely that competing forces including Hotmail and Gmail will willingly share trade secrets (for economic and potentially anti-trust reasons), the effort required to protect the average account using Aquinas is minimal. If the consequence of content compromise is more dangerous, the number of accounts used should be increased. While the Chinese government was able to put pressure on Yahoo! Mail to turn over information on suspected members of the political opposition, the ability of a government to achieve the same if Aquinas is used is minimized. Because it is unlikely that every provider will be compliant with foreign governments, communications can be protected from this sort of interception. One way to realistically implement a significant increase in the number of accounts would be for users to aggregate and share accounts within larger communities. In a design similar to the Crowds [18], users could receive and forward mail on behalf of other users within their community while maintaining plausible deniability of the communication details.

Techniques leveraging the temporal spacing of messages can also help to protect against traffic analysis attacks. As mentioned in Section 2.5, a user can include chaffing and winnowing techniques to increase their security. For example, slowly sending shares over the course of an hour forces an adversary to

consider all egress traffic during that period. A small alteration to the current version of Aquinas would allow it to continuously emit low volumes of traffic to randomly chosen websites and accounts. Shares included within this stream would be significantly more difficult to detect.

Due to the nearly infinite number of ways in which data can be injected into the Internet, the probability of an adversary selecting all of the correct repositories is incalculably small. Even in the unlikely event of an adversary having perfect knowledge of the accounts used for communication, a user can still be protected. Assuming that 40 messages are again used, but that the number of keys used is decided out of band (perhaps as part of account selection as in Section 2.6), an adversary is would be required to try up to $2^n - 1$, or nearly 1.1 trillion, combinations of messages. The action of selecting accounts therefore becomes equivalent to encryption by an additional, unrelated key. If the accounts are unknown, the size of this key is arguably infinite. In the worst case, the key size of the secondary in this example is 40-bits. Users uncomfortable with such a key length can increase robustness by changing the algorithm used to generate the encryption key from the key shares. If the $\bigoplus$ operation is replaced by an order-dependent technique (such as alternating multiplication and division of key shares according to the account selection scheme in Section 2.6), the adversary will instead have to try $\sum_{k=1}^{n} {}_nP_k$ permutations, as between 1 and n shares in the correct order could be required to reassemble the key. This operation has time complexity $O(n!)$. With 40 messages, more than $1.6 * 10^{48}$ permutations would be required to uncover the key. As this is much larger than the number of brute-force attempts to recover a 128-bit key, a user is sufficiently protected against even the strongest adversaries.

4.2 Resiliency

While offering robustness to the collusion of multiple service providers, the multi-path key and message delivery mechanism described in this paper is not without its own limitations. For example, if an email service provider were to determine that a message contained a key, simply deleting the message would prevent the intended recipient from decrypting and reading their mail. A message mistakenly classified as spam would have similarly deleterious effects, as the user would have difficulty differentiating real messages amongst the torrent of spam messages most email users receive.

Shamir's *threshold secret sharing* [25] could be used to make Aquinas robust against share-loss. This technique works by creating the key K from the combination of n key shares. K can be reconstructed as long as k key shares (where $n = 2k - 1$) are in the possession of the recipient. The advantage to this scheme is that it allows for $k-1$ key shares to be lost (or delivered late) without affecting the ability of the recipient to decrypt and read their email. If spam filtering were to become an issue, this scheme would be more robust, as it would allow the intended recipient to still read their encrypted messages without all n keys. While this approach is secure to the compromise of up to $k - 1$ key shares, if $k < n$, messages can be decrypted with fewer keys than in the currently implemented scheme.

Robustness based upon the perceived threat of an adversary could also be incorporated as a keying mechanism. For example, a user may decide that the overhead of increasing the number of email accounts is greater than the protection offered from a keying scheme based on threshold secret sharing. One simple extension to the multipath mechanism is to increase the number of accounts to which copies of key shares are sent. A user could opt to send the same key share to multiple accounts. In so doing, fewer cooperating adversaries would be necessary to reconstruct keys. A more elegant solution would be to use a mechanism based on *error correcting codes* (ECC). By attaching tags containing a few extra bytes to the end of each key, it becomes possible to reconstruct K with only a subset of all n key shares. The size of this subset (and the attached ECC) needed to recreate K can be adjusted to suit the specific expected adversary. The threshold secret sharing, multi-share delivery and error correcting code alternatives are all under consideration for future versions of this software.

5 Related Work

Privacy on the Internet is not guaranteed for users in general, and can be ambiguously defined even where it exists [15]. Often, users believe that they have online privacy but really have no guarantees to that effect [14]. To mitigate these shortcomings, many privacy-preserving tools have been created and deployed, protecting numerous aspects of a user's online activities.

Methods of securing non-web-based email have been extensively studied. Solutions such as Privacy Enhanced Mail (PEM) [12] and its successor, Secure MIME (S/MIME) [17], provide confidentiality, integrity, and non-repudiation for email messages. With PEM, this is accomplished through the construction of a full certificate hierarchy within a public key infrastructure (PKI); this has proven to be unwieldy in practice. For S/MIME, cryptographically transformed messages are sent as attachments within email, with key validation performed through a PKI. Pretty Good Privacy (PGP) [28] is another system for providing confidentiality and integrity of email that does not rely on the use of a PKI. A user forms a *web of trust* by trusting certain entities she communicates with, which in turn has other trusted relationships. The transitive certification paths of trust among these relationships are used to authenticate the source of email. Confidentiality can be provided by the mailer itself, with tools such as *ssmail*, a patch for the *sendmail* [5] mail transfer agent.

The *Off-the-record Email* (OTR) system [10] works at the user level, with dynamic key management performed between the two parties using it. Additionally, OTR provides non-recoverability of email messages once they have been deleted, even if the private keys used to generate the cryptographic operations have been revealed. However, while forward secrecy is assured, plausible deniability is not: an agent monitoring traffic will observe that encrypted information is being transmitted to the recipient.

While privacy within web-based email services has been largely absent, one solution is offered by SAFe-mail.net [23]. This system supplies confidentiality

and integrity through the use of a PKI that is run by SAFe-mail themselves. Because the service handles both certificates and user email, however, it has access to all of a user's information, allowing them to arbitrarily link and use this data.

Secure publication of data is another area where privacy can be crucial, in order to protect the authors of controversial documents from reprisal. The ability to publish without the fear of retribution has been tremendously important to citizens throughout history. The Federalist papers in the United States brought forth the ideals that ultimately became enshrined in the Constitution, but many of the authors published anonymously to avoid reprisal. More recently, the former Soviet-bloc countries witnessed the rise of *samizdat*, the process of anonymously publishing and distributing information banned by the government [24]. Publius [13] is a tool that facilitates secure publishing on the Internet, using threshold keying (discussed further in Section 4) to preserve anonymity. Other systems, including Free Haven [6], provide anonymous storage and retrieval. Similarly, Freenet [4], a distributed system for storage, provides anonymous content storage and dynamic growth of the network through the addition of new nodes.

Many of these tools have been useful in keeping communications private and secure; in particular, PGP has been extensively used by human rights organizations around the world. However, in virtually all cases, the fact that communication has taken place can be divined through the presence of encrypted data, or information has been transferred through private services. To this point, there have not been any solutions that allow for encrypted and steganographically concealed communications that transmit information solely through public channels and publicly available services.

6 Conclusion

This work has introduced Aquinas, an open source tool for preserving the privacy of user communication carried by web-email services. Each message is initially encrypted with a random symmetric key. The resulting ciphertext and key are both divided into shares. Each share is hidden in randomly chosen cover-text using steganography and sent through an independent web email account. Clients reconstitute the ciphertext and keys from shares received via the appropriate accounts. The result is decrypted to obtain the original message. We use email accounts in an analogous manner to the multiple channels employed in spread-spectrum communications. More generally, we show that the retention of one's privacy is possible regardless of the policies imposed by the providers of these web-based services.

Future extensions to this work will incorporate a variety of new image and linguistic steganography techniques, allowing users to more fully obfuscate their communications. Additionally, we will implement features that support the distribution of ciphertext shares across multiple accounts, and will continue to improve the usability of our interface as directed by user input. Such an approach also begs extension to the panoply of channels available throughout the Internet.

Our future work will not only explore these diverse channels, but also develop a formal framework for reasoning about the security provided by them.

References

1. BBC News. Chinese man 'jailed due to Yahoo'. http://news.bbc.co.uk/2/hi/asia-pacific/4695718.stm, February 2006.
2. BitTorrent. http://www.bittorrent.com.
3. K. Butler, W. Enck, J. Plasterr, P. Traynor, and P. McDaniel. Privacy Preserving Web-based Email. Technical report, Technical Report NAS-TR-0009-2005, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, June 2005.
4. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: a distributed anonymous information storage and retrieval system. In *International workshop on Designing privacy enhancing technologies*, pages 46–66, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
5. B. Costales and E. Allman. *Sendmail(2nd ed.)*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1997.
6. R. Dingledine, M. J. Freedman, and D. Molnar. The Free Haven Project: Distributed Anonymous Storage Service. In *International workshop on Designing privacy enhancing technologies*, pages 67–95, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
7. Electronic Frontier Foundation. http://www.eff.org.
8. C. M. Ellison and B. Schneier. Ten Risks of PKI: What You're Not Being Told About Public-Key Infrastructure. *Computer Security Journal*, 16(1):1–7, 1999.
9. D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private Internet connections. *Commun. ACM*, 42(2):39–41, 1999.
10. P. Henry and H. Luo. Off-the-record email system. In *Proceedings of IEEE INFOCOM 2001*, pages 869–877, Anchorage, AK, USA, Apr. 2001.
11. E. Jordan and A. Becker. Princeton officials broke into Yale online admissions decisions. http://www.yaledailynews.com/article.asp?AID=19454, July 25, 2002.
12. S. T. Kent. Internet privacy enhanced mail. *Commun. ACM*, 36(8):48–60, 1993.
13. A. D. R. Marc Waldman and L. F. Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. *Proc. 9th USENIX Security Symposium*, pages 59–72, August 2000.
14. R. L. Mcarthur. Reasonable expectations of privacy. *Ethics and Inf. Tech.*, 3(2):123–128, 2001.
15. L. Palen and P. Dourish. Unpacking "privacy" for a networked world. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 129–136, New York, NY, USA, 2003. ACM Press.
16. D. Peppers and M. Rogers. *The One to One Future: Building Relationships One Customer at a Time*. Doubleday, 1993.
17. B. Ramsdell. S/MIME version 3 message specification. RFC 2633, IETF, June 1999.
18. M. K. Reiter and A. D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
19. Reporters Without Borders. Information supplied by Yahoo! helped journalist Shi Tao get 10 years in prison. http://www.rsf.org/article.php3?id_article=14884, September 2005.

20. R. L. Rivest. Chaffing and Winnowing: Confidentiality without Encryption. *RSA CryptoBytes*, 4(1), Summer 1998.
21. W. Roger. Surfer beware: Advertiser's on your trail, DoubleClick tracks online movements. *USA Today*, page 01.B, 26 Jan. 2000.
22. B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. Mitchell. Stronger Password Authentication Using Browser Extensions. In *Proceedings of the 14th USENIX Security Symposium*, 2005.
23. SAFe-mail.net. SAFe-Mail features. http://www.safe-mail.net/help/SAFeMail-http://www.safe-mail.net/help/SAFeMailFeatures.html Features.html, May 2005.
24. G. Saunders. *Samizdat: Voices of the Soviet Opposition*. Pathfinder Press, Atlanta, GA, USA, 1974.
25. A. Shamir. How to share a secret. *Commun. ACM*, Vol 22:612–613, 1979.
26. SNOW. The SNOW Home Page. http://www.darkside.com.au/snow/.
27. The Anonymizer. http://www.anonymizer.com.
28. P. R. Zimmermann. *The official PGP user's guide*. MIT Press, Cambridge, MA, USA, 1995.