SoK: Securing Email—A Stakeholder-Based Analysis

Jeremy Clark[1], Paul C. van Oorschot[2], Scott Ruoti[3], Kent Seamons[4], and Daniel Zappala[4]

[1] Concordia University, Canada
j.clark@concordia.ca
[2] Carleton University, Canada
paulv@scs.carleton.ca
[3] University of Tennessee
ruoti@utk.edu
[4] Brigham Young University
seamons@cs.byu.edu
zappala@cs.byu.edu

Abstract. While email is the most ubiquitous and interoperable form of online communication today, it was not conceived with strong security guarantees, and the ensuing security enhancements are, by contrast, lacking in both ubiquity and interoperability. This situation motivates our research. We begin by identifying a variety of stakeholders who have an interest in the current email system and in efforts to provide secure solutions. We then use the tussle among stakeholders to explain the evolution of fragmented secure email solutions undertaken by industry, academia, and independent developers, and to draw the conclusion that a one-size-fits-all solution is unlikely. We highlight that vulnerable users are not well served by current solutions. We also account for the failure of PGP, and argue secure messaging, well complimentary, is not a fully substitutable technology.

1 Introduction

Email has been called *"probably the most valuable service on the Internet"* [14]. It has evolved over its 50-year history to become a pillar of seamless interoperability—if you know someone's email address, you can send email to them [111] across a diverse range of desktop, mobile, and web client software. As an indication of its near-universal acceptance, an email address is often required to create online accounts and to make online purchases. As of 2020, there were an estimated 4 billion users of email sending over 306 billion email messages per day [117]. Despite its ubiquity, email was not created the security desirable for its ensuing wide deployment.

Work to provide security for email, in various forms, has been ongoing for nearly three decades. Early efforts focused on the confidentiality, authenticity, and integrity of email messages, with efforts to develop PEM leading to work on S/MIME and then, as a reaction, PGP. However, as measured in recent years, email is only sometimes transmitted over an encrypted connection, with limited protection from passive network eavesdropping and active network attacks [35, 43, 98, 67]. Meanwhile, S/MIME has

Stakeholder	Description
Email Service Providers	Organizations that provide email services to industry and the public
Enterprise Organizations	Large organizations in both government and industry
Privacy Enthusiasts	Users with strong privacy preferences who believe email should offer strong protection from corporate or government surveillance
Vulnerable Users	Users who deal with strongly sensitive information that could induce personal safety risks, including journalists, dissidents, whistleblowers, informants, and undercover agents; we also include criminals as part of this stakeholder (due to aligned goals, despite ethical differences)
Secure Mailbox Providers	Organizations that provide secure email services to the public
Typical Users	Users of standard, plaintext email services
Enforcement	National security, intelligence, and law enforcement

Table 1. Stakeholders with an interest in email and secure email.

only seen limited uptake within enterprises and experts are abandoning PGP.[5] Greater attention has focused on spam, malware, and phishing as they became problems for everyday users. While spam filtering by many email providers has significantly improved, extensive email archives are typically stored in plaintext and vulnerable to hacking, and fraud through phishing and spear phishing remain problematic [120]. It is within this context that we set out to systematically understand what went wrong with email security, how email security can theoretically be improved, and how tussling between stakeholders can lead to inaction.

Contributions and Methodology. To better understand the current state of affairs and identify where future research and development efforts should focus, we conduct a stakeholder-based analysis of secure email systems. Our initial deliverable was a framework to evaluate secure email systems (preserved in the full version [25]), allowing us to map out the landscape of solutions and compare how they satisfy a set of security, utility, deployability, and usability properties. Ensuing discussion and review of this framework encouraged us to look specifically at how the actions and interests of a set of stakeholders (Table 1) helps to explain the history of failures and successes in secure email, leading to the current patchwork of partial secure email solutions. Using this new orientation for the paper, we systemize the academic literature on email, relevant IETF standards, industry solutions and software projects. For each, we consider which stakeholder is behind the proposal, determine how it furthers the goals of the stakeholder, and infer how these goals compose with the goals of other stakeholders. This allows us to identify

---

[5] Including Phil Zimmermann [44], the creator of PGP; Moxie Marlinspike [96], who called PGP a "*glorious experiment that has run its course*," and Filippo Valsorda [146], who bemoans the challenges of maintaining long-term PGP keys.

incompatibilities, illustrate how different solutions have evolved to meet their needs, and show which stakeholders are under-served.

We did not follow a fixed methodology for identifying research literature. We (i) examined the proceedings of top ranked security, cryptography, and measurements venues; (ii) expanded the research set by contemplating other work that was cited in the papers we identified; and (iii) relied on our personal experience (which, for some, dates back to the early 1990s) and our acquired knowledge of the literature. Similarly, the stakeholder groups were extracted from the literature through experience and discussion. It is likely that a different set of authors would end up with a somewhat different set of papers and categorizations, but this seems to be true of nearly all SoKs at top security venues.

Rise of Secure Instant Messaging. The relatively low level of adoption of secure email is often contrasted with the wider success of secure messaging applications. WhatsApp and Facebook Messenger have over a billion users, while iMessage, Signal, Telegram, Line, and Viber have millions. The best of these provide forward secrecy and message deniability [17, 113] in addition to end-to-end encryption. Unger et al. [145] have an excellent systematization of secure messaging. Yet, despite some calls to abandon secure email in favor of Signal [146], there are important reasons to not give up on email. Email is an open system, in contrast to messaging's walled gardens, giving it fundamentally different uses, often involving longer messages, archival, search, and attachments. There is no indication email is going away anytime soon. As such, there is still an important need to increase the security and privacy of email-based communication.

2 Preliminaries

A series of protocols are used to send email, transfer it from the sender's email provider to the recipient's provider, and then retrieve it. Figure 1 shows the most basic steps involved, in steps marked (1) through (3). When a user initiates sending an email, their client may use SMTP [82] to submit the message to their organization's mail server (also called a mail transfer agent or MTA [69, 27]). The sender's MTA uses DNS to locate the mail MTA for the recipient's domain, then uses SMTP to transfer the message. Finally, the recipient retrieves the message from their own organization's MTA, possibly using POP or IMAP. If either the sender or receiver is using webmail, then step (1) or step (3) may use HTTPS instead. Note also that the version of SMTP used to submit a message in step (1) is modified from the version of SMTP used to transfer messages [53].

This sequence of events is complicated somewhat by additional features supported by email as shown in step (4). First, a receiving MTA can be configured to forward email for a recipient on to another MTA; *e.g.,* forwarding email from `bob@company.org` to `bob@gmail.com`. This can repeat an arbitrary number of times. Second, a destination email address may correspond to a mailing list server which forwards the email to all subscribers on the list (a potentially large number). This adds numerous other recipient MTAs to the process.

An email message itself consists of two parts: the envelope and the body. The envelope contains SMTP commands that direct MTAs regarding how the message should be
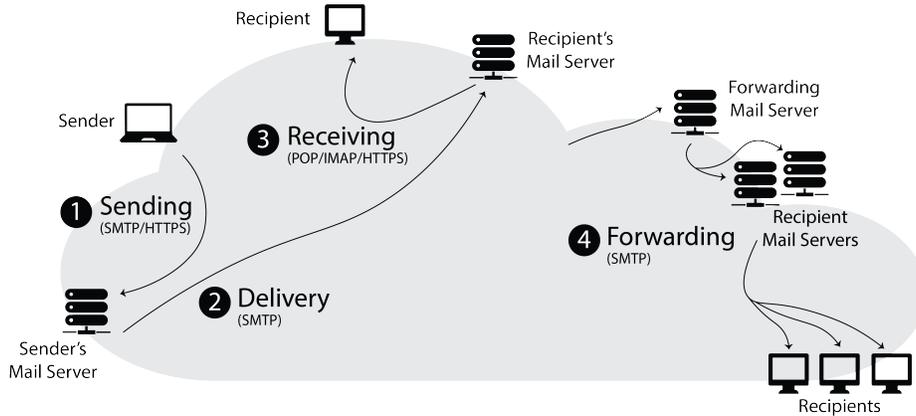
Fig. 1. Overview of email operation and protocols. (1) Sending email generally uses SMTP or HTTPS between a client and its mail server. (2) Delivery of email between mail servers uses SMTP. (3) Receiving email generally uses POP, IMAP, or HTTPS. (4) Any mail server receiving email may forward it to other servers. This happens when a user asks to forward their email to a different account, or when a user sends to a mailing list.

delivered. In particular, the envelope specifies the sender's email address (`MAIL FROM`) and the recipient's email address (`RCPT TO`). The message body has a separate format, including the familiar *From*, *To*, *CC*, and *Subject* header fields. Email clients generally display the sender's email address shown in the *From* header in the body, rather than the one in the SMTP envelope.

Why Email is Insecure. Every aspect of email was initially designed, specified, and developed without foreseeing the need for security protection that would later be recognized given how universal email has become. Security issues persist today despite decades of work to fix them. The original designs of protocols used to send, receive, and deliver email among clients and servers contained no protections for integrity or confidentiality. All messages were transmitted in the clear and could be intercepted and modified by anyone able to act as a man-in-the-middle. The original specifications contain nothing that validates the `MAIL FROM` command or prevents forgery of the *From* header. The ease of forging emails did nothing to inhibit the emergence of unsolicited email. Email never easily faciliated network-level anonymity, message deniability, or untraceability.

3 Stakeholders

The premise of our systematization of knowledge is that understanding the tussles among stakeholders are central to understanding why secure email lacks a universal solution. We identified potential stakeholders through an extensive period of analysis that included reviewing the research literature; reading online posts, discussion threads, and news articles regarding secure email; and by looking at press releases and features provided by secure email tools. We then carefully distilled the set to *key* stakeholders who: (1) reflect

unique preferences, and (2) are important to the history of research and development in this area (see Table 1).

An example of a stakeholder that is not a key stakeholder within our framework would be a company that produces client email software, as these companies tend to reflect the preferences of their customers—customers that are already key stakeholders like *enterprise organizations*, *typical users*, and *privacy enthusiasts*. Another example is government which is multifaceted. Many government departments operate like *enterprise organizations*, while others are captured by *enforcement*. But even within national security, law enforcement and intelligence agents and assets themselves have the preferences of *privacy enthusiasts* or *vulnerable users*. In this section, we align various efforts toward secure email with the appropriate stakeholders and in Section 4 discuss the trade-offs.

3.1 Email Service Providers

An email service (or mailbox) provider [27] is focused on retaining its customers for business and personal use. Providers have adopted several technologies to improve the security of email, including link encryption, domain authentication, and sender authentication. Providers often require access to plaintext so they can scan incoming emails for spam and malware. We review current and planned efforts, the protection they offer, and assessments of their effectiveness.

Link Encryption. Providers have adopted methods for encrypting email while it is in transit between MTAs or between an MTA and a client. Such 'link' encryption is designed to prevent eavesdropping and tampering by third parties that may own untrusted routers along the path that email is being delivered [65], however messages are not protected from inspection or modification at each MTA. While more privacy invasive than end-to-end encryption (encryption between the email sender and recipient), link encryption enables providers to scan for malicious email attachments, classify potential spam or phishing attacks, modify email tracking links, and provide other services.

Mail transferred with SMTP between MTAs is by default plaintext, and an MTA can use the STARTTLS command [65] to negotiate an encrypted channel. However, an active adversary between the MTAs can corrupt or strip STARTTLS, downgrading the connection to plaintext [35]. A recent initiative (currently called MTA-STS [95]) provides a way for an MTA to advertise a strict transport security (STS) policy stating that they always require STARTTLS. The policy is trusted on first use (TOFU) or authenticated using the certificate authority (CA) system. Should DNSSEC become widely deployed, policies can be directly advertised by the MTA in its DNS record [12, 66]. Even with link encryption, SMTP reveals significant metadata about email messages—some proposed mitigations have been drafted [92, 140].

Recall that email client software most often uses IMAP (or the older POP3) to retrieve mail and SMTP to send messages. STARTTLS is supported across each of these protocols [108] and is often required by the mail server. Users of webmail typically access their mail client using HTTPS. Under the link encryption paradigm, end users can ensure encryption to their mail server but have no control over (or even visibility of) the use of encryption for the transport of their emails.

Authentication. Consider the case when Alice receives an email from `bob@gmail.com`. *Domain authentication* indicates that the email was sent by a server authorized to send email from `gmail.com`, while *sender authentication* validates the user account `bob@gmail.com` originated the mail. The final level of authentication is *user authentication*, which occurs when Alice ensures that a human, such as Bob Smith owns the `bob@gmail.com` account. While user authentication is ideal, it taps into a public key infrastructure that email providers have avoided, settling instead for *domain authentication*, which has a long history rooted in identifying spam and filtering malware [81, 41, 85, 5].

Domain Authentication. The primary protocol for domain authentication is DomainKeys Identified Mail (DKIM) [28, 84]. The server originating email for a particular domain will generate a digital signature key pair, advertise the public key in the DNS record for the same domain, and sign all outbound email, with the appropriate validation data added to a header field in the email. A well-positioned adversary can modify a recipient's retrieval of the public key from DNS—DNSSEC can mitigate this threat [6]. DKIM signatures are fragile to any modification to the message body or header fields.

Using the same principle of advertising through DNS records, Sender Policy Framework (SPF) [81] allows a domain to specify which IP addresses are allowed to originate email for their domain, while Domain Message Authentication, Reporting, and Conformance (DMARC) [85] enables specification of which services (DKIM, SPF) they support, along with a policy indicating what action should be taken if authentication fails. DMARC has many additional features around reporting misconfigurations and abuse, but importantly it also requires identifier alignment. For SPF, this means that the domain in the envelope `MAIL FROM` address (which is authenticated with SPF) must match the domain in the *From* header field. For DKIM, this means that the domain used for signing must match the domain in the *From* header field. This links the authentication or signature verification done by SPF and DKIM to the *From* address seen by the user.

Sender Authentication. There is no wide support for sender authentication. Most mailbox providers do authenticate their users [64]. For example, if the sender is using webmail, then she may authenticate by logging into her webmail account. If the sender is using a desktop client, the mail domain can authenticate her with SMTP Authentication, which provides several methods that enable the sender to authenticate with the MTA by a username and password [137, 138, 136]. However, the measures a domain uses to authenticate a sender are not communicated to the recipient of an email message, nor can they be verified by the recipient.

Reducing the Fragility of Authentication. Authenticated Received Chain (ARC) [5, 74] extends email authentication to handle cases when messages are potentially modified when being forwarded, such as by a mailing list. With ARC, authentication checks are accumulated by forwarders in a message header field [83] as well as a signature on the email as received (these header fields are sealed with an additional signature by each forwarder, creating a chain). The protocol is intended for broad use by all email handlers along a transmission path, not just perimeter MTAs, and it is designed to allow handlers to safely extend the chain even if when they are certain they have not modified the message. When all email handlers are trusted by the recipient, ARC enables any modifications to

the message to be attributed, and for DKIM, SPF, and DMRAC results to be validated on the pre-modified message. However, a malicious handler is not prevented from altering messages or removing ARC headers.

Mitigating Email Misuse. Mailbox providers have invested significant effort in spam, phishing, and malware filtering. In the early 2010s, a successful malicious email campaign might see a spammer employ a botnet of 3,000 compromised machines to send 25 billion emails to 10 million addresses [72]. Each aspect of the pipeline—from the compromised machines to the email list to the software tools—might be sold by specialists [90], and the campaign itself is typically run by third-party advertisers earning pay-per-click revenue for directing traffic to a third-party site (*e.g.,* storefronts for unregulated pharmaceuticals constitute over a third of spam) [99].

Spam filtering has evolved from IP address blacklists to highly sophisticated classifiers that examine content, meta-information including origin, user reports, and protocol details such as SMTP header fingerprints [142]. Malware filtering is often performed by comparing email attachments to signatures of known malware. Spammers use a variety of evasion techniques, including sending from the IP addresses of malware-compromised computers [45], spoofing sender addresses, and encoding text as images. An esoteric proposal for spam prevention is requiring the sender to compute a costly function to send an email [36, 9]—an approach that never caught on [87].

Measurement Studies of Adoption and Effectiveness. In 2015–2016, several papers were published [35, 43, 98, 67] that measured the level of adoption and effectiveness of the encryption and domain authentication used by email providers. The general picture they paint is that top email providers encrypt messages with STARTTLS and use SPF and DKIM for authentication, but there is a long tail of organizations that are lagging in deploying these mechanisms. However, even when protection methods within email are deployed, they are often compromised by insecure practices, such as acceptance of: self-signed certificates[6] (when CA-signed certificates were expected), expired certificates, or broken chains, all of which cause the validation of the certificate to fail. Email traffic often uses weak cipher suites, weak cryptographic primitives and parameters, weak keys, or password authentication over unencrypted connections. Of the techniques that rely on DNS, basic attacks such as DNS hijacking, dangling DNS pointers [94], and modifying non-DNSSEC lookups can enable circumvention. Stripping attacks can compromise STARTTLS, with Durumeric et al. [35] illustrating how these attacks caused 20% of inbound Gmail messages to be sent in cleartext for seven countries. Use of SPF is common, but enforcement is limited, and DNS records often are not protected with DNSSEC. There is little use of DKIM, and few servers reject invalid DKIM signatures [43].

As Mayer et al. [98] conclude, *"the global email system provides some protection against passive eavesdropping, limited protection against unprivileged peer message forgery, and no protection against active network-based attacks."*

---

[6] With the advent of free domain certificates with Let's Encrypt, it is possible that more providers are using verifiable certificates since these measurements were conducted in 2015–2016.

3.2 Enterprise Organizations

Enterprises have overlapping interests with email service providers (like reducing email misuse) but often prefer stronger (end-to-end) encryption and authentication, at least within their internal boundaries. Enterprises played a role in developing standards that could meet their needs, starting with PEM [80, 93, 79, 11, 75] and leading to S/MIME [116, 115, 26]. Another issue that is highly relevant to enterprises is mitigating carefully targeted social engineering attacks against its employees, often conducted through email.

End-to-end encryption and authentication. The primary goals of PEM [80, 93, 79, 11, 75] were end-to-end email security with confidentiality, data origin authentication, connectionless integrity (order not preserved), non-repudiation with proof of origin, and transparency to providers and to SMTP. PEM was distinguished by interoperability with non-PEM MTAs, and a hierarchical X.509 public key infrastructure (PKI) with revocation that largely precludes rogue certificate issues haunting later PKI systems. A contributing factor cited [110] in PEM's demise was its slow progress in evolving for Multipurpose Internet Mail Extensions (MIME) [46], the standard for including attachments, multi-part bodies, and non-ASCII character sets. Industry support moved to S/MIME, while privacy advocates favored PGP (see Section 3.3) because it was free from the restrictions imposed by PEM's centralized and hierarchical organization.

S/MIME [115] is a standards suite for securing MIME data with both encryption and digital signatures. It was originally developed during the early 1990s by RSA Data Security, then later adopted by the IETF, resulting in standards in 1999 [116, 115, 26]. S/MIME's Cryptographic Message Syntax (CMS) [68] has origins in PEM and PKCS. S/MIME has wide support on major platforms and products [110, p.60–62]. S/MIME clients use *certificate directories* to look up X.509v3 certificates.[7] S/MIME does not mandate a hierarchy with a single root certificate authority (CA) and any organization can act as an independent, trusted root for its certificates—the most common usage today. Interoperability between organizations is limited or non-existent.

Several works have examined usability deficiencies with S/MIME implementations, noting difficulties knowing which CAs to trust [78], difficulties with certificate management [47], and inconsistency in handling certificates [110, p.60–67]. Automatically creating and distributing signing and encryption keys at account creation is considered good practice [48].

Private Key Escrow. Enterprises often use *private key escrow* in conjunction with S/MIME, which enables the organization to decrypt emails and scan for spam, malware, fraud, and insider trading, as well as archiving messages for regulatory reasons and enabling recovery if a client loses its private key. The suitability of S/MIME's centralized certificate management for enterprises and government has led to large, but siloed, deployments [21]. Some providers simplify S/MIME deployment using *hosted S/MIME* [60], where an enterprise uploads user private keys to an email provider, and the provider automatically uses S/MIME for some emails (*e.g.,* to other users of the same provider). Encryption in this case is only *provider-to-provider* rather than end-to-end.

---

[7] Of note, S/MIME uses a supporting suite of certificate management protocols, including RFC 5280 [26], which defines an IETF subset of X.509v3 certificates.

As an alternative to S/MIME, some enterprise email solutions rely on identity-based encryption (IBE) [135]. IBE uses a trusted server to store a master private key and generate individual private keys for users. The trusted server also advertises a master public key, which clients can use to derive a public key for any email address. Users can validate their ownership of an email address with the IBE server to retrieve their generated private key. IBE simplifies key management for clients but leaves the IBE server with persistent access to each user's private key, and also substantially complicates revocation [16].[8] Ruoti et al. [127, 125] integrated IBE into a webmail system, demonstrating how automating interactions with key management results in successful task completion and positive user feedback.

Transparent email encryption. A distinct approach to making interactions with PKI transparent to users is to layer encryption and signing below client software. Levien et al. [91] places this functionality between the email client software and the MTA, while Wolthusen [153] uses the operating system to intercept all network traffic and then automatically apply email encryption. Currently, several companies (*e.g.,* Symantec) offer automated encryption of emails by intercepting them as they traverse a corporate network.

Spear Phishing. Social engineering may be crafted as a generic attack but is often a targeted attack against specific enterprise employees. The openness of email enables direct contact with targets and an opportunity to mislead the target through the content of the email, a spoofed or look-alike send address, and/or a malicious file attachment [104, 61]. As an illustration, the company RSA was breached through a sophisticated attack that started with a targeted email impersonating an employee and a corrupted spreadsheet attachment [120]. Employee training [20] and email filtering are important counter-measures, however spam filters are typically trained to detect *bulk* email delivery and classifying bespoke spear phishing emails remains a challenge [86].

3.3 Privacy Enthusiasts

Privacy enthusiasts prefer end-to-end encrypted email to avoid government surveillance or commercial use of their data generally. They differ from vulnerable users (see section 3.4) in that there is not an immediate personal safety risk driving their usage of secure email. Privacy enthusiasts have historically favored PGP, which was developed as "public key cryptography for the masses" and "guerrilla cryptography" to counter authorities [158]. The difficulty with PGP has always been finding a suitable alternative to the centralized trust model of S/MIME.

End-to-end encryption and authentication. PGP's history is a fascinating 25-year tale of controversy, architectural zig-zags, name ambiguity, and patent disputes, with changes in

---

[8] Revocation of a compromised private key can be supported by having versions of the key. The result of obtaining an incorrect key version is comparable to obtaining a compromised key. The trust model of IBE is tantamount to a trusted public key server.

algorithms, formats and functionality; commercial vs. non-commercial products; corporate brand ownership; and circumvention of U.S. crypto export controls.[9] The current standard for the PGP message format is OpenPGP [19, 37], a patent-unencumbered variation. Despite evolving formats or encryption algorithms, PGP enthusiasts until recently have largely remained faithful to PGP's distinguishing concepts:

- **PGP key packets and lightweight certificates:** PGP key packets hold bare keys (public or private). Public keys are kept in *lightweight certificates* (*cf.* [158]), which are not signed certificates in the X.509 sense, but instead contain keys and a User ID (username and email address). To help client software determine which keys to trust, PGP also includes *transferable public keys* [19], which include one or more *User ID packets* each followed by zero or more *signature packets*. The latter attest the signing party's belief that the public key belongs to the user denoted by the User ID. Users typically store private keys on their local device, often encrypted with a password, though hardware tokens are also available.
- **PGP's web of trust:** The web of trust (WoT) is a model in which users personally decide whether to trust public keys of other users, which may be acquired through personal exchanges or from public servers, and which may be endorsed by other users they explicitly designate to be *trusted introducers* [157].
- **PGP key packet servers:** Users publish their public key to either closed or publicly accessible key packet servers, which contain a mapping of email address to the public key. Clients query to locate the public key associated with an email address.

Problems with PGP. PGP's design around the web of trust has allowed quick deployment in small groups without bureaucracy or costs of formal Certification Authorities [100], but leads to other significant obstacles:

- **Scalability beyond small groups:** Zimmerman notes [158, p.23] that *"PGP was originally designed to handle small personal keyrings"*. Scaling PGP requires acquiring large numbers of keys, along with a manual trust decision for each key, plus manual management of key storage and the key lifecycle.
- **Design failure to address revocation:** Zimmermann writes [158, p.31], *"If your secret key is ever compromised...you just have to spread the word and hope everyone hears about it"*. PGP does have methods to revoke keys, but distribution of these to others is ad hoc.
- **Usability by non-technical users:** Zimmerman [158, p.31] says *"PGP is for people who prefer to pack their own parachutes"*. There is no system help or recovery if users fail to back up their private key or forget their passphrase. Furthermore, users must understand the nuances of generating and storing keys, trusting public keys, endorsing a public key for other users, and designating others as trusted introducers. The poor usability of PGP has received significant attention [151, 123].

---

[9] PGP was distributed as freeware on the Internet in 1991, leading to an investigation of Zimmermann by the United States Customs Office for allegedly violating U.S. export laws. He published the PGP source code in book form in 1995 [156], and the case was subsequently dropped in 1996 [88].

- **Trust model mismatch:** Zimmerman notes [158, p.25] that *"PGP tends to emphasize [an] organic decentralized non-institutional approach"* reflecting personal social interaction rather than organizational relationships. The PGP web of trust was designed to model social interaction, rather than decision-making processes in governments and large enterprises. It is thus not a one-size-fits-all trust model.

Trust-on-first-use (TOFU). An alternative to PGP's web of trust is to exchange keys in-band and have clients trust them on first use. This has been the subject of several research projects [122, 49, 97]. Since 2016, the developer community has been integrating TOFU into PGP implementations in the MailPile, PEP [15], LEAP [140], and Autocrypt [143] projects. A common critique of TOFU is that users cannot distinguish valid key changes from an attack. Recent work by developers in the PEP and LEAP projects is aiming to address this problem with additional methods to authenticate public encryption keys, such as using a trusted public key server, auditing public key servers, and the fraught procedure of asking the user to compare key fingerprints [70, 31].

Public key servers and logs. Another web of trust alternative—applicable to (and aligned with) S/MIME's trust model—is introducing a trusted public key server. Recent work [8, 126] showed that automated servers have high usability when integrated into a PGP-based email system. Bai et al. [10] found users prefer key servers to manual key exchange, even after being taught about the security limitations of a key server.

A compromise between TOFU and a fully trusted server is to allow key assertions from users but ensuring they are published publicly in untrusted logs, allowing monitors to examine a history of all certificates or key packets that a key server has made available for any entity [130, 101, 13]. This enables detection of rogue keys and server equivocation.

Social Authentication. Another way to disseminate public keys is to associate them with public social media accounts. The Keybase project[10] helps users to post a signed, cryptographic proof to their account, simultaneously demonstrating ownership of a public key and ownership of the account. By aggregating proofs across multiple social media accounts for the same person, a client can establish evidence that ties a public key to an online persona, under the assumption that it is unlikely that a person's social media accounts are all compromised simultaneously. The Confidante email system leverages Keybase for distribution of encryption public keys, with a study finding it was usable for lawyers and journalists [89].

Short-lived keys and forward secrecy. Schneier and Hall [133] explored the use of short-term private keys to minimize the damage resulting from the compromise of a private key. Brown and Laurie [18] discuss timeliness in destroying a short-lived key and how short-lived keys complicate usability by requiring more frequent key dissemination.

3.4 Vulnerable Users

Vulnerable users deal with strongly sensitive information that could induce personal safety risks. Using email from a malware-infected device is a primary concern [22, 62], as well as risks due to the design and common practices of email.

---

[10] https://keybase.io

Pseudonymity. One concern for vulnerable users is the inability to forgo leaking personally identifiable meta-information: *i.e.,* unlink the contents of the email from their true email address, their IP address, and/or the identity of their mail server. Technically inclined vulnerable users generally opt for pseudonymity [58] where more than one email sent from the same pseudonymous account can be established as having the same origin, but no further information is known.

Historically, PEM accommodated anonymous users with *persona certificates*, which could provide assurances of continuity of a pseudonymous user ID but does not prevent network level traceability. Today, *layered encryption* is used in which messages are routed through multiple non-colluding servers, with each server unwrapping a layer of encryption until the message is delivered to its destination, with the same happening for replies in reverse. This idea was championed by the cypherpunk movement [106, 107] and adapted to the email protocol with remailers like mixminion and others [55–57, 29]. Pseudonymity is realized as indistinguishability from a set of plausible candidates—the set of other users at the time of use [33]—which may be small, depending on the system and circumstances.[11]

A simpler approach is to register a webmail account under a pseudonymous email address, optionally using Tor [34] to access the mailbox. Satoshi Nakamoto, the inventor of Bitcoin [105], corresponded over webmail for many months while remaining anonymous.

Traceability, deniability, and ephemerality. Email senders for some time have abused the browser-like features of modern email clients to determine when recipients view an email, when a links are clicked, and (via third-party trackers) what other collected information is known about the recipient [38]. Email service provider interventions can interfere with domain authentication (DKIM).

Deniability considers a case where the recipient wants to authenticate the sender, but the sender does not want the evidence to be convincing to anyone else. Cryptographers have suggested new signature types [23, 73, 119] to provide deniability, but these typically require trusted third parties and/or a robust PKI and have near-zero deployment.

Once sent, a sender loses control over an email and the extent to which its contents will be archived. In order to automate a shorter retention period, emails might contain a link to the message body which is deposited with and automatically deleted by a trusted service provider or a distributed network [52, 152].

3.5 Secure Mailbox Providers

A secure mailbox provider offers end-to-end encryption and authentication between users of their service. Providers like ProtonMail [114], Hushmail [71], and Tutanota [144] have millions of users combined. Users' private keys are password-protected client-side and then stored with the provider, preventing provider access (assuming the password is strong [42]) while allowing cross-device access. However, providers are trusted in other regards: inter-user encryption and authentication is generally blackbox and not

---

[11] To illustrate, a student emailed a bomb threat to Harvard's administration via webmail accessed over Tor [34]. The suspect was found to be the only individual accessing Tor on Harvard's network at the time the email was sent—while strictly circumstantial, the suspect confessed [59].

independently verifiable,[12] and the model relies on client-side scripting where malicious (first or third-party) scripts would compromise security. Additional methods are needed to provide code signing and privilege separation for JavaScript in the browser [103, 147]. Generally, email sent to outside users are encrypted client-side with a one-time use passphrase, deposited in message repository with an access link sent as the original email (the passphrase is communicated between the sender and recipient out-of-band).

A second approach is to use a browser extension to overlay signed and encrypted email on an existing mailbox provider. Initiatives here include automating PGP key management tasks (Mailvelope and FlowCrypt), providing automated S/MIME-based encryption and signing (Fossa Guard), encryption with a symmetric key held by the service (Virtru), or encryption using a password shared out of band (SecureGmail). Google developed E2EMail to integrate OpenPGP with Gmail in Chrome but the project has been inactive for several years.

3.6 Typical Users

Some work has examined the question of why most people do not use encrypted email. Renaud et al. [118] found support for four reasons for non-adoption—lack of concern, misconceptions about threats, not perceiving a significant threat, and not knowing how to protect themselves. An earlier survey of 400+ respondents by Garfinkel et al. [48] found that half indicated they didn't use encrypted email because they didn't know how, while the rest indicated they didn't think it was necessary, didn't care, or thought the effort would be wasted. Other work reports that users are unsure about when they would need secure email [124] and are skeptical that any system can secure their information [128, 30]. It is not clear that users want to use digital signatures or encryption for daily, non-sensitive messages [40, 51]. Overall, work in this area demonstrates that usability is not the only obstacle to adoption, and that users don't perceive significant risk with email, lack knowledge about effective ways to mitigate risk, and don't have self-confidence about their ability to effectively use secure systems.

The usable security and privacy community is increasingly utilizing new approaches to address broader questions of adoption of security and privacy practices. Users are often rational when making decisions about whether to follow security advice; Herley [63] makes the case that users sometimes understand risks better than security experts, that worst-case harm is not the same as actual harm, and that user effort is not free. Sasse [132] has likewise warned against scaring or bullying people into doing the "right" thing. As a result, effort is being made to understand users' mental models [150, 39, 77, 155] when they interact with secure software and using risk communication techniques to better understand adoption or non-adoption of secure software [141, 154], among other methods.

3.7 Enforcement

We broaden the term enforcement to encompass police and law enforcement agencies, as well as national security and intelligence services. Law enforcement prioritizes access to

---

[12] Fingerprint comparison is common with secure messaging applications, but the feature is often ignored by users [134].

plaintext communications, either through broad surveillance or exceptional access such as with a warrant. This need for access to plaintext communications has led to calls for so-called encryption back doors, leading to regular debates on whether this is desirable or feasible. This debate originally surfaced in the U.S. in the 1990s concerning email and has been rekindled regularly, now with greater emphasis on instant messaging which has seen better success than email at deploying end-to-end encryption to regular users. Proponents cite fears that widespread use of end-to-end encryption will enable criminals and terrorists to "go dark" and evade law enforcement. In response, privacy advocates decry growing mass surveillance, point to a history of abuses of wiretapping [32], and suggest that market forces will ensure there is plenty of unencrypted data for use by law enforcement regardless [50].

A 2015 paper from Abelson et al. [2] highlights risks of regulatory requirements in this area, reiterating many issues discussed in their earlier 1997 report [1]. Identified risks include reversing progress made in deploying forward secrecy, leading to weaker privacy guarantees when keys are compromised; substantial increases to system complexity, making systems more likely to contain exploitable flaws; and the concentration of value for targeted attacks. Their report also highlights jurisdictional issues that create significant complexity in a global Internet. More broadly, whenever service providers have access to keys that can decrypt customer email, this allows plaintext to be revealed due to incompetent or untrustworthy service providers, by disillusioned employees, by government subpoena, or by regulatory coercion.

4 Stakeholder Priorities

In the previous section, we aligned past efforts in securing email with their appropriate stakeholders. In Table 2, we establish 17 priorities that are important to at least one stakeholder. These priorities are a result of extensive discussion among the authors using our literature review and current practices as evidence for our ratings.

For each stakeholder, a given priority can be a high, low, or a non-priority. In some cases, we rate a stakeholder as highly valuing partial support of a property. We also identify several cases where a stakeholder has a high priority that the property is *not* met, meaning it is antithetical to their goals. We lightly clustered the stakeholders into three groups. Enforcement has unique priorities for the targets of their investigation; priorities are to backdoor completely confidential and anonymous communication. The second cluster generally prioritizes utility and deployability, while the third prefers security. We accept that the reader may disagree with some rankings but believe the framework enables a useful discussion of tradeoffs that are often otherwise glossed over.

We call particular attention to instances where a stakeholder strongly opposes a property (marked ☠). One might think that no stakeholder would be opposed to increase security, utility, deployability, or usability. However, enforcement prefers a system where exceptional access is granted (S4), as do enterprises, because analyzing plaintext is essential to their operation. (One could argue that enforcement prefers when most traffic is not encrypted at all.) Enforcement likewise prioritizes attribution and thus opposes sender pseudonymity (S7). Vulnerable users are opposed to server-side content processing

Stakeholder	S1	S2	S3	S4	S5	S6	S7	S8	T1	T2	T3	T4	D1	D2	D3	U1	U2
	Security								Utility				Deploy.			Usab.	
Enforcement		☠		☠													
Email Service Providers	▦							■	⋮	■	■		⚔	■			
Typical Users	▦			⚔					■				■				
Enterprise Organizations	▦		☠		■				■		⋮			■	■		
Secure Mailbox Providers	■		▦	■	■			■		⋮						■	■
Privacy Enthusiasts	■		⚔	■	⋮	⋮		■	⋮		⚔	⋮				■	■
Vulnerable Users	■	■	■	■	■	■					☠	☠					⋮

Column key:
S1: Protection from eavesdropping
S2: Protection from tampering and injection
S3: Private keys only accessible to user
S4: Prevents exceptional access
S5: Responsive public key revocation
S6: Provides a public key audit trail
S7: Supports sender pseudonymity
S8: Easy to detect phishing
T1: Supports user choice of email providers
T2: Supports user choice of identity provider
T3: Supports server-side content processing
T4: Provides persistent access to email
D1: No client software modifications needed
D2: No email server modifications needed
D3: No infrastructure modifications needed
U1: Effortless same system encryption key discovery
U2: Effortless encryption/signing key validation

Legend: ■ high priority for full support · ▦ high priority for partial support · ⋮ low priority · □ a non-priority or not applicable · ⚔ there is disagreement within the stakeholder group about the priority of this property · ☠ high priority for no support

Table 2. Stakeholder priorities.

(T3) and systems that provide persistent access (T4) since they cannot trust their safety to others.

There are several cases where we found disagreement within a stakeholder group regarding the priority of a given property (marked ⚔). An example is preventing exceptional access to email (S4)—typical email users are divided between those who advocate for government surveillance of email and who are willing to accept government access to email on presentation of a warrant, and those who strongly prefer end-to-end encryption that would prevent exceptional access. Likewise, privacy enthusiasts are split on whether there is a high priority on ensuring that private keys are accessible only to users (S3), with a minority placing a high priority on this property but others accepting password-protected cloud storage of a private key. Privacy enthusiasts are also split on whether persistent access to email is a high priority (T4), along similar lines. Finally,

while many email service providers place a high priority on not being required to deploy new email-related servers to support a given technology (D2), this is likely not a high priority for larger providers. For example, large providers have shown a willingness to adopt best practices such as STARTTLS and DKIM more rapidly.

In several cases, stakeholders have a high priority for partial support of a property but do not want it fully (or universally) supported (marked ▦). All stakeholders, aside from enforcement, prefer that emails are protected from eavesdropping by third parties (S1). However certain stakeholders want read capabilities for some email. For example, an enterprise may want to run automated services on their employees' plaintext emails—for security, compliance or other reasons—but do not want the emails accessible in plaintext by anyone outside of the enterprise, or even anyone within the enterprise that is not a party to the email. Similarly, enterprises and service providers may want the ability to modify email messages (S2) to protect their users (remove malware or insert a phishing warning) without disrupting message authentication. Users may want this protection as well.

As a final example of partial support, secure mailbox providers offer users the ability to control their own signing and encryption keys (S3) but balance this with some usability features. For example, storing password-protected decryption keys in the cloud allows users to check their email from new devices without transferring their keys, while it limits the provider's access to their users' decryption keys. This is in contrast to a (normal) email service provider that, if it supported encryption and signatures at all, would give customers the additional usability feature of backing up their private decryption keys, enabling key recovery and the ability to read past encrypted emails. Note that private keys for signing do not require backup as users can generate new ones, although the old public signature keys should be maintained for verification of past emails (or revoked if the signing key is stolen as opposed to lost).

Table 2 illustrates the reality that there are significant disagreements between stakeholders in the secure email space and that no single solution will satisfy them all. The strongest disagreements happen in columns where at least one stakeholder fully supports a property (marked ▉) while another strongly opposes it it (marked ☠). The four high conflict properties are exceptional access (S4), sender pseudonymity (S7), server-side content-processing (T3), and persistent access (T4).

The conflict between enforcement and other stakeholders over exceptional access (S4) and sender pseudonymity (S7) is well-known in both secure email and other technical domains: web browsing, network traffic, server IP addresses and locations, and payment systems. We emphasize again that the enforcement stakeholder category captures enforcement's preferences for the targets of their investigations and actions, while the agents themselves are better aligned with privacy enthusiasts, and agents could use (or create) vulnerable users through their investigations.

High conflict also exists over server-side content-processing (T3) for spam, malware filtering, classification, or automatic replies; and persistent access (T4) which indicates that the user can recover their access and archive after losing their authentication credentials. This conflict illustrates an important result: some of the most fundamental disagreements occur over the utility properties of a secure email system. Email service providers, typical users, and enterprise organizations all place a high value on content

processing and persistent access. Yet, these are mostly low priorities for the other stakeholders and, in some cases, antithetical to the principles held by vulnerable users who prioritize exclusive access to their email with no backdoors. Even if it means managing a secret value that only they know, they accept the risk of key loss being permanent.

The tussles among stakeholders help explain the history of how this space has evolved. The needs of typical users are largely met by email service providers; these two stakeholders disagree mainly on deployment properties that affect only the service provider (D2, D3), along with a tussle over exceptional access (S4). Privacy enthusiasts have a demonstrated history of highly valuing end-to-end encryption (hence the development of PGP and person-to-person key exchange), but it is not a priority for email service providers and typical users, and this explains why it is not pursued more broadly. The needs of some enterprise organizations to deploy secure email explains why they often adopt S/MIME based products. They need encryption within the organization, plus escrow of private keys and content processing. They also have the IT budget to provide a seamless user experience.

Privacy enthusiasts overlap significantly with enterprise organizations, but disagreements on private key storage (S3), server-side content processing (T3) and persistent access (T4) make finding common ground difficult. Privacy enthusiasts also overlap with vulnerable users but vulnerable users will tolerate poor usability and a lack of features to maximize security. To our knowledge, no major commercial provider currently meets the needs of vulnerable users.

Most email service providers prioritize opportunistic encryption with TLS. Secure email providers have emerged, with priorities that mostly match those of privacy enthusiasts, some of whom may previously have used PGP-based services. Some privacy enthusiasts would prefer the private key is only accessible to themselves (S3), but due to the loss of grass-roots support for PGP, the only apparent feasible alternative is password-protected keys used in secure webmail. The services offered by secure email providers have supported vastly more users of secure email than PGP ever did. However their business model naturally means some deployment properties cannot be met, hence requiring users to use new email software.

5 Further Discussion

After extensively reviewing the history of email, academic literature, and discussing stakeholder priorities, we highlight several critical points in understanding the state of secure email today.

A one-size-fits-all solution is unlikely. It is clear from Table 2 that stakeholders have conflicting priorities and that the needs of different stakeholders dictate diverging solutions. As such, it is unlikely that any single secure email system will be suitable for all users and their divergent use cases. Furthermore, no single party controls the email ecosystem, and widespread deployment of secure email needs cooperation of numerous stakeholders. No one stakeholder has the capability to build (or the ability to demand) a secure email system that provides seamless interoperability for the billions of email users and supports email's many diverse uses. This means that even in the best case, with different solutions

being adopted by different parties, there will almost surely be interoperability challenges that act as natural roadblocks and will require significant investment to overcome, if this is even possible.

The PGP web of trust remains unsuccessful after 25 years. The web of trust that is central to the original design of PGP—including manual key exchange and trusted introducers—has largely failed. Its use is generally limited to isolated, small communities. Its appeal is that it allows quick, interoperable deployment in small groups without bureaucracy or costs of formal Certification Authorities, but in practice the downside is poor usability and lack of responsive revocation. Arguably, the resulting product indecision and non-interoperability has negatively impacted the deployment of secure email in general.

Incremental improvement is still possible. Most email users trust their mailbox providers with plaintext email. While link encryption and domain authentication are available, vulnerabilities to active attacks and a lack of adoption leave email in transit subject to eavesdropping and message forgery. Providers could create an interoperable hosted S/MIME standard to automate provider-to-provider confidentiality and integrity, while still working within the threat model of a trusted mailbox provider. Unlike end-to-end encryption, server-based search, content-filtering, and persistent/portable mailbox access would be supported. Easy-to-deploy tools are needed to ensure the solution is not a barrier to entry for small providers.

Secure messaging is only a partial answer. Messaging protocols are walled gardens, allowing proprietary protocols that are interactive and supported by central servers. This enables automated encryption for users, including automatic key exchange via a trusted key server and automatic end-to-end encryption of messages [145]. Using a trusted key server means that users may be unaware of the security and usability tradeoffs they are making. Users of secure messaging applications are typically only warned to check the encryption keys if they change, and numerous studies have shown that these applications fail to help users understand how to do this successfully [3, 134, 149]. Security experts recommend encrypting all messages, however some applications make encryption optional, resulting in many users failing to turn encryption on [148].

Further, email's open nature gives it fundamentally different uses than messaging, including easily communicating with strangers, sending long, content-rich messages, permanently archiving messages, searching past conversations, and attaching files. While email's additional features are part of the reason ubiquitous end-to-end encryption is so elusive, they are also why email is likely to continue to be a primary form of communication on the Internet for years to come.

Vulnerable users are not well served. Aside from vulnerable users, every stakeholder represents a class of user that has their needs met by at least one system available today. Typical users are served by current offerings from email service providers. Enterprises (and their employees) are served by corporate S/MIME, which provides a combination of security, utility, and usability that matches their priorities. Deployment cost are likely what hinders its broader adoption among enterprises. Privacy enthusiasts are served by secure webmail services, with their stronger emphasis on end-to-end encryption and

good usability, while sacrificing utility to meet these priorities. In contrast, there is no system that clearly serves vulnerable users well. PGP is perhaps the best option, given its use by investigative journalists [121], but it does not meet all the security priorities of vulnerable users. No system except for remailers provides sender pseudonymity, and these do not typically meet other security properties important to vulnerable users. The small size and desire for anonymity among members of this stakeholder group (journalists, dissidents, whistleblowers, survivors of violence, informants, under-cover agents, and even criminals) does not lend itself to commercial solutions, and volunteer organizations in this area have historically struggled.

6   Research and Development Directions

Improving the security of email is important to us. In this section, we briefly outline several avenues for future research and development.

Interoperability. Interoperability among secure email systems is a complex topic. Email evolved into an open system decades ago, allowing anybody to email anyone else. Thus, a justifiable user expectation is that secure email should likewise be open. However, we are far from achieving this today with secure mailbox providers (recall Section 3.5), since the primary secure systems in use are walled gardens, as either online services and/or dedicated software clients. Using standardized cryptographic suites is a small step but systems should also allow key (and key server) discovery between services (*e.g.,* ProtonMail-esque mailboxes to enterprise S/MIME certificate directories).

Interoperability introduces challenging issues around privacy, spam, and trust. Enterprises and providers are unwilling to expose the public keys of their users to outside queries. Encrypted spam, and other kinds of malicious email, can evade standard content filtering techniques that work on plaintext. Different systems operate under different trust models. While the web has built a system based on global trust, this requires only one-way trust of the web server, whereas secure email involves two-way trust between individuals and organizations. Simply adopting the web's CA trust model would be unlikely to yield a workable system, given the challenges that remain still largely unsolved with this model [24]. Technically a system based on a CA alternative (*e.g.,* trust-on-first-use) could interoperate with a different system (*e.g.,* certificate directory) but typical users are unlikely to comprehend the difference in trust even if communicated to them, and the entire system could end up with weakest link security. Even if formats and protocols were universally agreed upon, it is not clear whether interoperability is always desired or meaningful. Finally, opening any system to interoperability means users will need help deciding which organizations or providers to trust to provide correct public keys. We argue it is both infeasible and unnecessary to expect that every individual or organization can be globally trusted by the others.

We advise future work on a much more limited goal of establishing trust among communicating parties when they need it. Any individual user or organization has a relatively small set of other users or organizations that it needs to trust. Developing infrastructure and protocols with this end in mind would appear to be necessary to leverage any gains made in technical interoperability.

Content inspection on encrypted email. Another major problem for secure email is coping with spam and malware. Even if interoperability was a solved problem, authentication of an email sender is not the same as authorization to send email [14], and building a system that provides the former but not the latter simply means users will get authenticated spam and phishing emails. End-to-end encryption systems without sufficient spam prevention for users are impractical, since both email providers and users lack an incentive to use such a system.

One possibility is to try to work around this problem. A secure email client could accept encrypted email only from regular or accepted contacts; rejecting encrypted email from unapproved senders could serve as a viable substitute for spam and malware filtering. Spam and malware could still be propagated by compromising accounts and spreading it to others who have approved those users, but the attack surface would be significantly limited. However, email providers are not likely to embrace such a system since it arguably offers less spam and malware protection for users than current practice.

A better alternative might be to build secure email systems that allow for server-side content processing even when private keys are only accessible to users. One possibility is to develop improved methods for processing on data that is encrypted [139, 54, 76]. Alternatively, clients could send encrypted email and a decryption key to a trusted cloud computing environment [131, 112], perhaps based on trusted execution platforms where the email could be decrypted and filtered for malware and spam. Likewise, a trusted computing environment could be used for storing and searching archives. Another possibility is to move email storage to edge devices owned by an end-user where content processing can be performed, with encrypted backup in the cloud to provide fault tolerance and portability.

Auditing identity providers. Providing an auditable certificate directory or key server enables a system to provide a public key audit trail, responsive public key revocation, and effortless public key verification. However, additional work is needed to ensure such a system can meet its goals. For example, consider auditing systems like Certificate Transparency and CONIKS [130, 101, 13]. When it is a user's personal public key that is audited in such a system, the system must also then provide a usable method for users to monitor the public keys being advertised. In the case that a client's system notices that an unauthorized key is advertised for them, the system needs a method for the user to whistleblow and have the offending key revoked. Additionally, if the user's own identity provider has equivocated, then the user needs a method for being informed of this in a trustworthy manner and then being guided on choosing a new identity provider. If the identity provider is also their email provider, then they will also need to choose a new email provider. These auditing systems are promising and would benefit from further development and study to the point where we can be confident that it will be easy for users to accomplish these tasks.

Increasing trust. Recent work has shown that even with the proliferation of secure messaging applications, there is still a gap in how users perceive the effectiveness of security technology [4, 30]. Users overestimate the capabilities of attackers and underestimate the strength of encryption technology, resulting in a lack of trust in applications that claim to protect their privacy. It is debatable whether this lack of trust is misplaced—the best

cryptography cannot protect against errors in implementations or breaches that expose data that is stored unencrypted. Users have a healthy skepticism of general software and technology when they pay attention to highly publicized security failures. This is further complicated by 'snake-oil' security and encryption tools that do not offer concrete benefits. Nevertheless, users are better off using encryption if they are going to communicate sensitive data online. Thus, user lack of trust in encryption is a major obstacle to overcome.

Trust is a longstanding challenge in computing [7]. Secure messaging is only secure if you trust WhatsApp, for example, to exchange keys properly, or if you know enough to verify exchanged keys manually, or if you trust your messaging partners not to reveal the content of your messages. Yet the biggest success to date in getting users to adopt secure communication—the use of secure messaging applications—is not due to users choosing security or privacy but because users migrate to applications with large user bases and convenient functionality, which happen to use end-to-end encryption [4]. It is not clear how email can follow the same path. Getting users to adopt secure email services may require gains in user understanding of risks and trust in solutions that mitigate those risks. The field of risk communication which has been used successfully for many years in public health, may offer a path toward helping users understand and cope with online security risks [109, 154].

Removing private key management barriers. There are numerous open questions regarding how typical (non-enterprise) users [129] will manage the full key life cycle, which includes private key storage, expiration, backup, and recovery [102, §13.7]. These questions are complicated by issues such as whether to use separate keys for encrypting email during transmission, as opposed to those for long-term storage [21]. Storing keys in trusted hardware where they cannot be exfiltrated solves some storage issues, but also requires users to create backup hardware keys and revoke keys stored in lost or stolen devices. It is worth noting that major browsers and operating systems now support synchronizing passwords across user devices (under a user account with the provider), and one part of solving key management problems may involve using similar techniques to synchronize private keys.

Addressing archive vulnerability. One of the consequences of high-profile phishing attacks in recent years has been the digital theft of the extensive information stored in long-term email archives of various individuals, companies, and organizations. It is ironic that the most active areas of research into securing email are largely orthogonal to the email security issues reported in the news. While data leaks might be categorized as a general data security issue, the way email products and architectures are designed (*e.g.,* emails archived by default, mail servers accessible by password) are inculpatory factors. Research on technical solutions, revised social norms about email retention, and other approaches could be helpful in addressing this issue.

7 Concluding Remarks

Deployment and adoption of end-to-end encrypted email continue to face many technical challenges, particularly related to key management. Our analysis indicates that conflicting

interests among stakeholders explains the fragmented nature of existing secure email solutions and the lack of widespread adoption. This suggests it is time to acknowledge that a one-size-fits-all (*i.e.,* for all target scenarios, environments, and user classes) solution or architecture will not emerge. In particular, we find the strongest conflicts among stakeholders over exceptional access, sender pseudonymity, server-side content-processing, and persistent access (T4). In each case, at least one stakeholder strongly prioritizes of these properties while another strongly opposes it.

In this light, a significant barrier to progress is opposition to any new product or service that does not meet one stakeholder's particular needs, though it works well for others. A path forward is to acknowledge the need for alternate approaches and support advancement of alternatives in parallel. Divided communities and differing visions can lead to paralysis if we insist on a single solution, but it can also be a strength if we agree that multiple solutions can co-exist.

Full Version. In the full version of this paper [25], we provide a detailed evaluation framework for secure email systems. Using the same properties as our stakeholder analysis, we evaluate existing secure email systems. The definition of each property is given, along with an explanation of how a given secure email system is rated to have full support, partial support, or no support in terms of meeting this property. This analysis shows how different secure email systems line up with the needs of each stakeholder. Highlighting the properties that are important to a stakeholder reveals which solutions serve them well or poorly.

References

1. Abelson, H., Anderson, R.J., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B.: The risks of key recovery, key escrow, and trusted third-party encryption. World Wide Web Journal **2**(3) (1997)
2. Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Green, M., Landau, S., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B., Specter, M., Weitzner, D.J.: Keys under doormats: Mandating insecurity by requiring government access to all data and communications. Journal of Cybersecurity **1**(1) (2015)
3. Abu-Salma, R., Krol, K., Parkin, S., Koh, V., Kwan, K., Mahboob, J., Traboulsi, Z., Sasse, M.A.: The security blanket of the chat world: An analytic evaluation and a user study of Telegram. In: European Workshop on Usable Security (EuroUSEC 2017). Internet Society (2017)
4. Abu-Salma, R., Sasse, M.A., Bonneau, J., Danilova, A., Naiakshina, A., Smith, M.: Obstacles to the adoption of secure communication tools. In: IEEE S&P (2017)

5. Andersen, K., Long, B., Blank, S., Kucherawy, M.: Authenticated Received Chain (ARC) protocol. RFC 8617, IETF (Jul 2019), `https://datatracker.ietf.org/doc/html/draft-ietf-dmarc-arc-protocol-09`, work in progress

6. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: DNS security introduction and requirements. RFC 4033 (March 2005)

7. Association, C.R.: Four grand challenges in trustworthy computing (2003)

8. Atwater, E., Bocovich, C., Hengartner, U., Lank, E., Goldberg, I.: Leading Johnny to water: Designing for usability and trust. In: SOUPS (2015)

9. Back, A.: Hashcash-a denial of service counter-measure. Tech. rep., hashcash.org (2002), `ftp://sunsite.icm.edu.pl/site/replay.old/programs/hashcash/hashcash.pdf`

10. Bai, W., Namara, M., Qian, Y., Kelley, P.G., Mazurek, M.L., Kim, D.: An inconvenient trust: User attitudes toward security and usability tradeoffs for key-directory encryption systems. In: SOUPS (2016)

11. Balenson, D.: Privacy enhancement for Internet electronic mail: Part III: Algorithms, modes, and identifiers. RFC 1423 (February 1993)

12. Barnes, R.L.: DANE: Taking TLS authentication to the next level using DNSSEC. IETF Journal **7**(2) (2011)

13. Basin, D., Cremers, C., Kim, T.H.J., Perrig, A., Sasse, R., Szalachowski, P.: ARPKI: Attack resilient public-key infrastructure. In: CCS (2014)

14. Bellovin, S.M.: A look back at "security problems in the TCP/IP protocol suite". In: ACSAC (2004)

15. Birk, V., Marques, H., Shelburn, Koechli, S.: pretty Easy privacy (pEp): Privacy by default. Internet-Draft draft-birk-pep-03, IETF (Mar 2019), `https://datatracker.ietf.org/doc/html/draft-birk-pep-03`, work in progress

16. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: CCS (2008)

17. Borisov, N., Goldberg, I., Brewer, E.: Off-the-record communication, or, why not to use PGP. In: WPES (2004)

18. Brown, I., Laurie, B.: Security against compelled disclosure. In: ACSAC (2000)

19. Callas, J., Donnerhacke, L., Finney, H., Shaw, D., Thayer, R.: OpenPGP message format. RFC 4880 (November 2007)

20. Caputo, D.D., Pfleeger, S.L., Freeman, J.D., Johnson, M.E.: Going spear phishing: Exploring embedded training and awareness. IEEE S&P Magazine **12**(1) (2014)

21. Chandramouli, R., Garfinkel, S.L., Nightingale, S.J., Rose, S.W.: Trustworthy email. Special Publication (NIST SP) 800-177 (2016)

22. Chatterjee, R., Doerfler, P., Orgad, H., Havron, S., Palmer, J., Freed, D., Levy, K., Dell, N., McCoy, D., Ristenpart, T.: The spyware used in intimate partner violence. In: 2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA. pp. 441–458. IEEE Computer Society (2018). https://doi.org/10.1109/SP.2018.00061, `https://doi.org/10.1109/SP.2018.00061`

23. Chaum, D.: Designated confirmer signatures. In: EUROCRYPT (1995). https://doi.org/10.1007/BFb0053427

24. Clark, J., van Oorschot, P.: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In: IEEE S&P (2013)

25. Clark, J., van Oorschot, P.C., Ruoti, S., Seamons, K.E., Zappala, D.: Securing email. CoRR **abs/1804.07706** (2018), `http://arxiv.org/abs/1804.07706`

26. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. RFC 5280 (May 2008)

27. Crocker, D.: Internet mail architecture. RFC 5598, IETF (2009)

28. Crocker, D., Hallam-Baker, P., Hansen, T.: DomainKeys Identified Mail (DKIM) service overview. RFC 5585 (July 2009)
29. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: design of a type iii anonymous remailer protocol. In: 2003 Symposium on Security and Privacy, 2003. pp. 2–15 (2003)
30. Dechand, S., Naiakshina, A., Danilova, A., Smith, M.: In encryption we don't trust: The effect of end-to-end encryption to the masses on user perception. In: EuroS&P 2019 (2019)
31. Dechand, S., Schürmann, D., IBR, T., Busse, K., Acar, Y., Fahl, S., Smith, M.: An empirical study of textual key-fingerprint representations. In: USENIX Security (2016)
32. Diffie, W., Landau, S.: Privacy on the Line: The Politics of Wiretapping and Encryption (2/e). The MIT Press (2007), second edition 2007 (472 pages), first edition 1998 (352 pages)
33. Dingledine, R., Mathewson, N.: Anonymity loves company: Usability and the network effect. In: WEIS (2006)
34. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: USENIX Security (2004)
35. Durumeric, Z., Adrian, D., Mirian, A., Kasten, J., Bursztein, E., Lidzborski, N., Thomas, K., Eranti, V., Bailey, M., Halderman, J.A.: Neither snow nor rain nor MITM...: An empirical analysis of email delivery security. In: IMC (2015)
36. Dwork, C., Naor, M.: Pricing via processing or combatting junk mail. In: CRYPTO (1992)
37. Elkins, M., Torto, D.D., Levien, R., Roessler, T.: MIME security with OpenPGP. RFC 3156 (August 2001)
38. Englehardt, S., Han, J., Narayanan, A.: I never signed up for this: Privacy implications of email tracking. PETS (2018)
39. Fagan, M., Khan, M.M.H.: Why do they do what they do?: A study of what motivates users to (not) follow computer security advice. In: SOUPS (2016)
40. Farrell, S.: Why don't we encrypt our email? IEEE Internet Computing **13**(1) (2009)
41. Fenton, J.: Analysis of threats motivating DomainKeys Identified Mail (DKIM). RFC 4686 (September 2006)
42. Florêncio, D., Herley, C., van Oorschot, P.C.: An administrator's guide to Internet password research. In: USENIX LISA (2014)
43. Foster, I.D., Larson, J., Masich, M., Snoeren, A.C., Savage, S., Levchenko, K.: Security by any other name: On the effectiveness of provider based email security. In: CCS (2015)
44. Franceschi-Bicchierai, L.: Even the inventor of PGP doesn't use PGP. motherboard.vice.com (September 2015), `https://motherboard.vice.com/en_us/article/vvbw9a/even-the-inventor-of-pgp-doesnt-use-pgp`
45. Franklin, J., Perrig, A., Paxson, V., Savage, S.: An inquiry into the nature and causes of the wealth of Internet miscreants. In: CCS (2007)
46. Freed, N., Borenstein, N.S.: Multipurpose Internet Mail Extensions (MIME) Part one: Format of Internet message bodies. RFC 2045 (November 1996)
47. Fry, A., Chiasson, S., Somayaji, A.: Not sealed but delivered: The (un) usability of S/MIME today. In: ASIA (2012)
48. Garfinkel, S.L., Margrave, D., Schiller, J.I., Nordlander, E., Miller, R.C.: How to make secure email easier to use. In: CHI (2005)
49. Garfinkel, S.L., Miller, R.C.: Johnny 2: A user test of key continuity management with S/MIME and Outlook Express. In: SOUPS (2005)
50. Gasser, U., Gertner, N., Goldsmith, J.L., Landau, S., Nye, J.S., O'Brien, D., Olsen, M.G., Renan, D., Sanchez, J., Schneider, B., et al.: Don't panic: Making progress on the "going dark" debate. Berkman Center for Internet & Society at Harvard Law School (2016)
51. Gaw, S., Felten, E.W., Fernandez-Kelly, P.: Secrecy, flagging, and paranoia: Adoption criteria in encrypted email. In: CHI (2006)
52. Geambasu, R., Kohno, T., Levy, A.A., Levy, H.M.: Vanish: Increasing data privacy with self-destructing data. In: USENIX Security Symposium (2009)

53. Gellens, R., Klensin, J.: Message submission for mail. RFC 6409 (November 2011)
54. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009)
55. Goldberg, I., Wagner, D., Brewer, E.: Privacy-enhancing technologies for the Internet. In: IEEE COMPCON. Digest of Papers (Feb 1997). https://doi.org/10.1109/CMPCON.1997.584680
56. Goldberg, I.: Privacy-enhancing technologies for the Internet, II: Five years later. In: PETS (2003)
57. Goldberg, I.: Privacy enhancing technologies for the Internet III: Ten years later. In: Acquisti, A., Gritzalis, S., Lambrinoudakis, C., De Capitani di Vimercati, S. (eds.) Digital Privacy: Theory, Technologies and Practices. Auerbach Press (2007)
58. Goldberg, I.A.: A Pseudonymous Communications Infrastructure for the Internet. Ph.D. thesis, UC Berkeley (2000)
59. Goodin, D.: Use of Tor helped FBI ID suspect in bomb hoax case. Ars Technica (December 2013)
60. Google: Hosted S/MIME by Google provides enhanced security for Gmail in the enterprise (2019), `https://security.googleblog.com/2017/02/hosted-smime-by-google-provides.html`
61. Hadnagy, C.: Social Engineering: The art of human hacking. Wiley (2010)
62. Havron, S., Freed, D., Chatterjee, R., McCoy, D., Dell, N., Ristenpart, T.: Clinical computer security for victims of intimate partner violence. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 105–122. USENIX Association, Santa Clara, CA (Aug 2019), `https://www.usenix.org/conference/usenixsecurity19/presentation/havron`
63. Herley, C.: So long, and no thanks for the externalities: The rational rejection of security advice by users. In: NSPW (2009)
64. Hoffman, P.: Allowing relaying in SMTP: A series of surveys. Internet Mail Consortium Report **16** (2002)
65. Hoffman, P.E.: SMTP service extension for secure SMTP over Transport Layer Security. RFC 3207 (February 2002)
66. Hoffman, P.E., Schlyter, J.: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698 (Aug 2012). https://doi.org/10.17487/RFC6698, `https://rfc-editor.org/rfc/rfc6698.txt`
67. Holz, R., Amann, J., Mehani, O., Wachs, M., Kaafar, M.A.: TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication. In: NDSS (2016)
68. Housley, R.: Cryptographic Message Syntax (CMS). RFC 5652 (September 2009)
69. Houttuin, J.: A tutorial on gatewaying between x.400 and internet mail. RFC 1506, IETF (2016)
70. Hsiao, H.C., Lin, Y.H., Studer, A., Studer, C., Wang, K.H., Kikuchi, H., Perrig, A., Sun, H.M., Yang, B.Y.: A study of user-friendly hash comparison schemes. In: ACSAC (2009)
71. Hushmail (2019), `https://www.hushmail.com/`
72. Iedemska, J., Stringhini, G., Kemmerer, R., Kruegel, C., Vigna, G.: The tricks of the trade: What makes spam campaigns successful? In: SPW (2014)
73. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: EUROCRYPT (1996)
74. Jones, S.M., Rae-Grant, J., Adams, J.T., Andersen, K.: Recommended Usage of the Authenticated Received Chain (ARC). Internet-draft, IETF (May 2020)
75. Kaliski, B.: Privacy enhancement for Internet electronic mail: Part IV: Key certification and related services. RFC 1424 (February 1993)
76. Kamara, S.: Encrypted search. XRDS **21**(3), 30–34 (Mar 2015). https://doi.org/10.1145/2730908, `http://doi.acm.org/10.1145/2730908`

77. Kang, R., Dabbish, L., Fruchter, N., Kiesler, S.: "My data just goes everywhere:" User mental models of the Internet and implications for privacy and security. In: SOUPS (2015)

78. Kapadia, A.: A case (study) for usability in secure email communication. IEEE S&P Magazine **5**(2) (2007)

79. Kent, S.: Privacy enhancement for Internet electronic mail: Part II: Certificate-based key management. RFC 1422 (February 1993)

80. Kent, S.T.: Internet privacy enhanced mail. CACM **36**(8) (1993)

81. Kitterman, D.S.: Sender Policy Framework (SPF) for authorizing use of domains in email, version 1. RFC 7208 (April 2014)

82. Klensin, J.C.: Simple Mail Transfer Protocol. RFC 5321 (October 2008)

83. Kucherawy, M.: Simple Mail Transfer Protocol. RFC 8601, IETF (May 2019)

84. Kucherawy, M., Crocker, D., Hansen, T.: DomainKeys Identified Mail (DKIM) signatures. RFC 6376 (September 2011)

85. Kucherawy, M., Zwicky, E.: Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489 (March 2015)

86. Laszka, A., Vorobeychik, Y., Koutsoukos, X.D.: Optimal personalized filtering against spear-phishing attacks. In: AAAI (2015)

87. Laurie, B., Clayton, R.: Proof-of-work proves not to work; version 0.2. In: WEIS (2004)

88. Lauzon, E.: The Philip Zimmermann investigation: The start of the fall of export restrictions on encryption software under first amendment free speech issues. Syracuse L. Rev. **48**, 1307 (1998)

89. Lerner, A., Zeng, E., Roesner, F.: Confidante: Usable encrypted email: A case study with lawyers and journalists. In: IEEE EuroS&P (2017)

90. Levchenko, K., Pitsillidis, A., Chachra, N., Enright, B., Félegyházi, M., Grier, C., Halvorson, T., Kanich, C., Kreibich, C., Liu, H., McCoy, D., Weaver, N., Paxson, V., Voelker, G.M., Savage, S.: Click trajectories: End-to-end analysis of the spam value chain. In: IEEE S&P (2011)

91. Levien, R., McCarthy, L., Blaze, M.: Transparent Internet e-mail security. In: NDSS (1996)

92. Levison, L.: Dark Internet Mail Environment architecture and specifications (March 2015), `https://darkmail.info/downloads/dark-internet-mail-environment-march-2015.pdf`

93. Linn, J.: Privacy enhancement for Internet electronic mail: Part I: Message encryption and authentication procedures. RFC 1421 (February 1993)

94. Liu, D., Hao, S., Wang, H.: All your DNS records point to us: Understanding the security threats of dangling DNS records. In: CCS (2016). https://doi.org/10.1145/2976749.2978387

95. Margolis, D., Risher, M., Lidzborski, N., Chuang, W., Long, D., Ramakrishnan, B., Brotman, A., Jones, J., Martin, F., Umbach, K., Laber, M.: SMTP MTA Strict Transport Security. RFC 8461, IETF (2018)

96. Marlinspike, M.: GPG and me. moxie.org (February 2015), `https://moxie.org/blog/gpg-and-me/`

97. Masone, C., Smith, S.W.: Abuse: PKI for real-world email trust. In: EuroPKI. Springer (2009)

98. Mayer, W., Zauner, A., Schmiedecker, M., Huber, M.: No need for black chambers: Testing TLS in the e-mail ecosystem at large. In: IEEE ARES (2016)

99. McCoy, D., Pitsillidis, A., Grant, J., Weaver, N., Kreibich, C., Krebs, B., Voelker, G., Savage, S., Levchenko, K.: PharmaLeaks: Understanding the business of online pharmaceutical affiliate programs. In: USENIX Security Symposium (2012)

100. McGregor, S.E., Watkins, E.A., Al-Ameen, M.N., Caine, K., Roesner, F.: When the weakest link is strong: Secure collaboration in the case of the Panama papers. In: USENIX Security Symposium (2017)

101. Melara, M.S., Blankstein, A., Bonneau, J., Felten, E.W., Freedman, M.J.: CONIKS: Bringing key transparency to end users. In: USENIX Security Symposium (2015)
102. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of applied cryptography. CRC press (1996)
103. Meyerovich, L., Livshits, B.: ConScript: Specifying and enforcing fine-grained security policies for JavaScript in the browser. In: IEEE S&P (2010)
104. Mitnick, K.D., Simon, W.L.: The art of deception: Controlling the human element of security. John Wiley & Sons (2011)
105. Nakamoto, S.: Bitcoin: A peer-to-peer electionic cash system. Unpublished (2008), `https://bitcoin.org/bitcoin.pdf`
106. Narayanan, A.: What happened to the crypto dream?, Part 1. IEEE S&P Magazine **11** (2013). https://doi.org/doi.ieeecomputersociety.org/10.1109/MSP.2013.45
107. Narayanan, A.: What happened to the crypto dream?, Part 2. IEEE S&P Magazine **11** (2013). https://doi.org/doi.ieeecomputersociety.org/10.1109/MSP.2013.75
108. Newman, C.: Using TLS with IMAP, POP3 and ACAP. RFC 2595 (June 1999)
109. Nurse, J.R., Creese, S., Goldsmith, M., Lamberts, K.: Trustworthy and effective communication of cybersecurity risks: A review. In: Workshop on Socio-Technical Aspects in Security and Trust (STAST 2011). IEEE (2011)
110. Orman, H.: Encrypted Email: The History and Technology of Message Privacy. Springer (2015)
111. Partridge, C.: The technical development of Internet email. IEEE Annals of the History of Computing **30**(2) (2008)
112. Pasquier, T.F.M., Singh, J., Eyers, D., Bacon, J.: Camflow: Managed data-sharing for cloud services. IEEE Transactions on Cloud Computing **5**(3), 472–484 (2017)
113. Perrin, T., Marlinspike, M.: Double ratchet algorithm, revision 1. signal.org (2016)
114. Protonmail (2019), `https://protonmail.com/`
115. Ramsdell, B., Turner, S.: Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.2 message specification. RFC 5751 (January 2010)
116. Ramsdell, B.C.: S/MIME version 3 message specification. RFC 2633 (June 1999)
117. The Radicati Group: Email statistics report, 2020–2024 (2019)
118. Renaud, K., Volkamer, M., Renkema-Padmos, A.: Why doesn't Jane protect her privacy? In: PETS (2014)
119. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: ASIACRYPT (2001)
120. Rivner, U.: Anatomy of an attack. RSA blog (1 April 2011), `http://web.archive.org/web/20110413224418/http://blogs.rsa.com:80/rivner/anatomy-of-an-attack/`
121. Romera, P., Gallego, C.S.: How ICIJ deals with massive data leaks like the Panama Papers and Paradise Papers (3 July 2018), `https://www.icij.org/blog/2018/07/how-icij-deals-with-massive-data-leaks-like-the-panama-papers-and-paradise-papers/`
122. Roth, V., Straub, T., Richter, K.: Security and usability engineering with particular attention to electronic mail. International Journal of Human-Computer Studies **63**(1) (2005)
123. Ruoti, S., Andersen, J., Dickinson, L., Heidbrink, S., Monson, T., O'Neill, M., Reese, K., Spendlove, B., Vaziripour, E., Wu, J., Zappala, D., Seamons, K.: A usability study of four secure email tools using paired participants. ACM Transactions on Privacy and Security **22**(2) (April 2019)
124. Ruoti, S., Andersen, J., Heidbrink, S., O'Neill, M., Vaziripour, E., Wu, J., Zappala, D., Seamons, K.: "We're on the same page": A usability study of secure email using pairs of novice users. In: CHI (2016)
125. Ruoti, S., Andersen, J., Hendershot, T., Zappala, D., Seamons, K.: Private webmail 2.0: Simple and easy-to-use secure email. In: UIST (2016)

126. Ruoti, S., Andersen, J., Monson, T., Zappala, D., Seamons, K.: A comparative usability study of key management in secure email. In: SOUPS (2018)
127. Ruoti, S., Kim, N., Burgon, B., Van Der Horst, T., Seamons, K.: Confused Johnny: When automatic encryption leads to confusion and mistakes. In: SOUPS (2013)
128. Ruoti, S., Monson, T., Wu, J., Zappala, D., Seamons, K.: Weighing context and trade-offs: How suburban adults selected their online security posture. In: SOUPS (2017)
129. Ruoti, S., Seamons, K.: Johnny's journey toward usable secure email. IEEE Security & Privacy **17**(6), 72–76 (2019)
130. Ryan, M.D.: Enhanced certificate transparency and end-to-end encrypted mail. In: NDSS (2014)
131. Santos, N., Gummadi, K.P., Rodrigues, R.: Towards trusted cloud computing. HotCloud **9**(9), 3 (2009)
132. Sasse, A.: Scaring and bullying people into security won't work. IEEE S&P Magazine **13**(3) (2015)
133. Schneier, B., Hall, C.: An improved e-mail security protocol. In: ACSAC (1997)
134. Schröder, S., Huber, M., Wind, D., Rottermanner, C.: When SIGNAL hits the fan: On the usability and security of state-of-the-art secure mobile messaging. In: EuroUSEC (2016)
135. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Crypto (1984)
136. Siemborski, R., Gulbrandsen, A.: IMAP extension for Simple Authentication and Security Layer (SASL) initial client response. RFC 4959 (September 2007)
137. Siemborski, R., Melnikov, A.: SMTP service extension for authentication. RFC 4954 (July 2007)
138. Siemborski, R., Menon-Sen, A.: The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) authentication mechanism. RFC 5034 (July 2007)
139. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: IEEE S&P (2000)
140. Sparrow, E., Halpin, H., Kaneko, K., Pollan, R.: LEAP: A next-generation client VPN and encrypted email provider. In: CANS (2016)
141. Stewart, G., Lacey, D.: Death by a thousand facts: Criticising the technocratic approach to information security awareness. Information Management & Computer Security **20**(1) (2012)
142. Stringhini, G., Egele, M., Zarras, A., Holz, T., Kruegel, C., Vigna, G.: B@bel: Leveraging email delivery for spam mitigation. In: USENIX Security Symposium (2012)
143. Team, A.: Autocrypt level 1 specification, release 1.1.0 (April 2019)
144. Tutanota (2019), `https://tutanota.com/`
145. Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., Smith, M.: SoK: Secure messaging. In: IEEE S&P (2015)
146. Valsorda, F.: Op-ed: I'm throwing in the towel in PGP, and I work in security. Ars Technica (December 2016)
147. Van Acker, S., De Ryck, P., Desmet, L., Piessens, F., Joosen, W.: WebJail: Least-privilege integration of third-party components in web mashups. In: ACSAC (2011)
148. Vaziripour, E., Wu, J., Farahbakhsh, R., Seamons, K., O'Neill, M., Zappala, D.: A survey of the privacy preferences and practices of iranian users of telegram. In: Workshop on Usable Security (USEC) (2018)
149. Vaziripour, E., Wu, J., O'Neill, M., Clinton, R., Whitehead, J., Heidbrink, S., Seamons, K., Zappala, D.: Is that you, Alice? A usability study of the authentication ceremony of secure messaging applications. In: SOUPS (2017)
150. Wash, R.: Folk models of home computer security. In: SOUPS (2010)
151. Whitten, A., Tygar, J.D.: Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In: USENIX Security Symposium (1999)

152. Wolchok, S., Hoffman, O.S., Henninger, N., Felten, E.W., Haldermann, J.A., Rossback, C.J., Waters, B., Witchel, E.: Defeating Vanish with low-cost sybil attacks against large DHTs. In: NDSS (2010)
153. Wolthusen, S.D.: A distributed multipurpose mail guard. In: IAW (2003)
154. Wu, J., Gatrell, C., Howard, D., Tyler, J., Vaziripour, E., Seamons, K., Zappala, D.: "Something isn't secure, but I'm not sure how that translates into a problem": Promoting autonomy by designing for understanding in Signal. In: SOUPS (2019)
155. Wu, J., Zappala, D.: When is a tree really a truck? exploring mental models of encryption. In: SOUPS (2018)
156. Zimmermann, P.: PGP source code and internals. MIT Press (1995)
157. Zimmermann, P.: PGP marks 10th anniversary (5 June 2001)
158. Zimmermann, P.R.: The official PGP user's guide. MIT press (1995)