Editors: Heather Richter Lipford, richter@uncc.edu | Jessica Staddon, jessica.staddon@gmail.com

Johny's Journey Toward Usable Secure Email

Scott Ruoti | University of Tennessee
Kent Seamons | Brigham Young University

Since the publication of "Why Johnny Can't Encrypt,"[10] there has been interest in creating usable, secure email that is adoptable by the general public. In this article, we summarize research from the usable-security community on this topic, identify open problems, and call for more research on usable key management.

Security Challenges

Email has existed for nearly 50 years, and it is doubtful that its inventors recognized how widespread its use would become. Statista estimates that there are currently 3.8 billion email users, and that number is expected to grow to 4.3 billion by 2022. Email usage is estimated at 281 billion messages/day, with an estimated 333 billion messages/day by 2022. In short, email has become an essential part of our modern world.

Like many early Internet systems, email was not designed to protect against network attackers or compromised/untrustworthy servers. Figure 1 shows how plaintext email has many vulnerabilities that an adversary can exploit: ① unsecured links: an attacker reads, modifies, or drops traffic during email transmission; ② message forgeries: an attacker injects messages claiming to be from any user; ③ malicious content: an attacker sends email with malicious content (e.g., malware, phishing, spam); and ④ untrusted servers: the mail providers and intermediary servers can read and/or modify email.

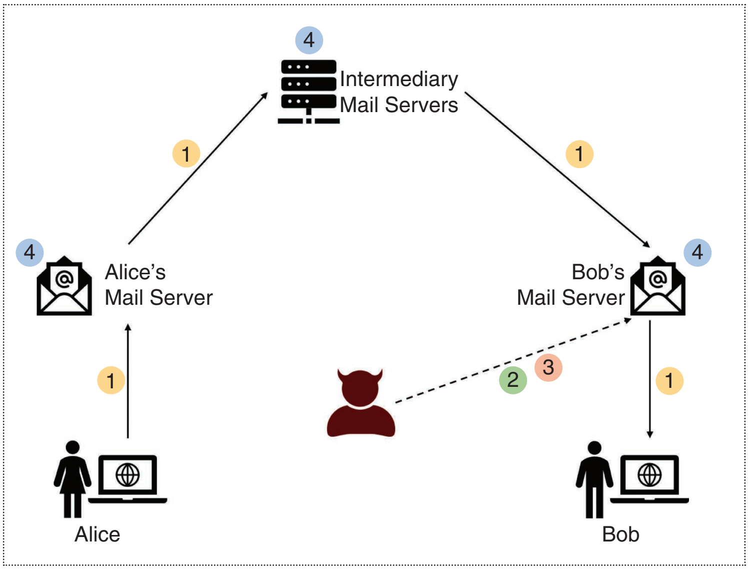


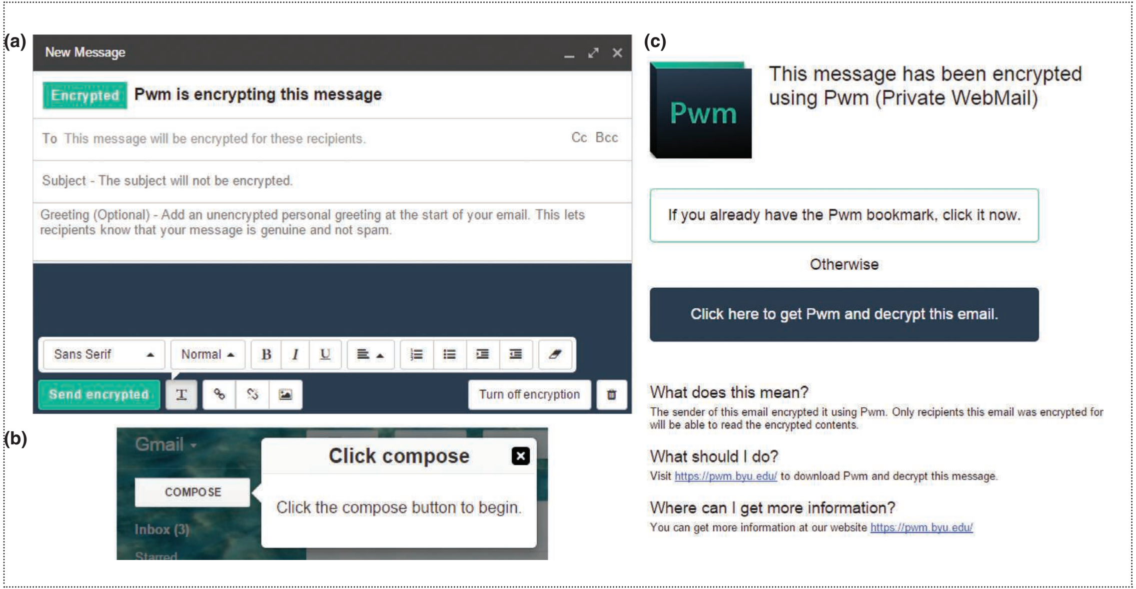
Figure 1. The vulnerabilities of email include ① unsecured links,② message forgeries, ③ malicious content, and ④ untrusted servers.

Technologies addressing the first three vulnerabilities are already deployed by major email providers.[4] First, Transport Layer Security secures the communication links as email is transmitted. Second, DomainKeys Identified Mail allows mail servers to sign email to protect against forged messages. Third, machine learning identifies and filters malicious content. Unfortunately, there is a long tail of mail providers who have either not deployed these technologies or misconfigured them, leaving email vulnerable.[4]

Even if correctly adopted, these technologies do not address the fourth vulnerability: untrusted servers. The sender's and recipient's mail providers—along with intermediary mail transfer agents between their providers—can see and potentially modify email sent by users. Even if the servers themselves are trusted, server compromise or government surveillance is still a problem.

End-to-end, encryption-based secure email systems address this fourth vulnerability—as well as the first and second vulnerabilities—by encrypting/signing email at the sender's side and only decrypting/verifying it once it reaches the recipient. The two most popular implementations of end-to-end encryption are Pretty Good Privacy (PGP) and the Secure

Digital Object Identifier 10.1109/MSEC.2019.2933683
Date of current version: 29 October 2019



Pwm is encrypting this message

To This message will be encrypted for these recipients. Cc Bcc

Subject - The subject will not be encrypted.

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

Click compose

Click the compose button to begin.

This message has been encrypted using Pwm (Private WebMail)

If you already have the Pwm bookmark, click it now.

Otherwise

What does this mean?

The sender of this email encrypted it using Pwm. Only recipients this email was encrypted for will be able to read the encrypted contents.

What should I do?

Visit https://pwm.byu.edu/ to download Pwm and decrypt this message.

Where can I get more information?

You can get more information at our website https://pwm.byu.edu/

Figure 2. The interface for Private Webmail (Pwm) 2.0,[8] a modern usable secure email system. (a) The placeholder text that acts as an inline tutorial instructing users about how secure email works. (b) An inline, context-sensitive tutorial helping users send encrypted email for the first time. (c) The body of the encrypted email providing plaintext instructions to streamline onboarding.

Multipurpose Internet Mail Extensions (S/MIME) standard.

For many years, the community thought that these solutions were sufficient. Whitten and Tygar shattered this view with their seminal article "Why Johnny Can't Encrypt,"[10] revealing significant usability issues in the leading PGP client of the time. This article helped establish the fact that usability and security are intertwined and cannot be easily separated.

Although usability is not the only issue impeding the adoption of secure email,[4] it remains an important challenge. In this article, we summarize usable secure email research, including increasing the usability of secure email interfaces, user perceptions toward secure email use, and easy-to-use key management. For each topic, we also identify open research questions.

Making Secure Email Interfaces Usable

Recent research has produced several highly usable secure email interfaces (e.g., Atwater et al.,[2] Garfinkel and Miller,[5] and Lerner et al.[6]). Figure 2 shows the interface for Private Webmail 2.0, a system we created and improved through a series of studies (e.g., Ruoti et al.[7,8]) over the last eight years. This research shows that the following are essential properties for usable secure email interfaces.

- *Tight integration*: Users want secure email systems that enhance their existing email clients and fit within their existing workflows.[2] This integration includes both visual and functional integration—that is, it looks like a part of the client application and has similar functionality, respectively (see Figure 2). Although visual integration is important, care must be taken to ensure that users can still clearly distinguish between emails protected with end-to-end encryption and those that are not.[8] Although most users prefer integrated solutions, a small but consistent portion prefer standalone clients,[2,6] believing that handling secure email in a separate client makes it obvious when encryption is in use.
- *Inline, context-sensitive tutorials*: Tutorials are essential in helping users understand how to use secure email properly.[8] For users to pay attention and use these tutorials, it is important that the tutorial be shown inline with the secure email system.[8] Additionally, the system should provide context-sensitive tutorials, walking first-time users through the process of sending and receiving secure email [Figure 2(a) and (b)].
- *Streamlined onboarding*: Encrypted email should be designed to help recipients understand what they have received and what actions they need to take next[8] [see Figure 2(c)]. If the secure email system requires recipients to first generate a key pair, the system should automatically send an email explaining what the recipient needs to do.[2] Additionally, the system should save a draft of the sender's message to send automatically after the recipient generates and makes the public key available.
- *Understandable and trustworthy design*: It is vital for interfaces to help users understand how secure email is protecting them—for example, telling them whether the subject line is encrypted (it usually is not).

Increased understanding allows users to make informed decisions and avoid mistakes.[3,8] Additionally, system operation needs to conform to user expectations; otherwise, users reject the system. For example, we found that if encryption happens too quickly, users assume that their messages were not actually encrypted and, thus, do not trust the tool.[8]
- *Easy-to-use key management*: Users struggle with managing their keys. Automation of key generation, uploading, and discovery significantly improve the user experience.[2,3,5]

Studies show that systems applying (most) of these principles are perceived as highly usable, result in a low mistake rate, and help novice users begin sending encrypted email without expert assistance.[2,3,6–8]

Research Directions: Longitudinal Studies

Current research always uses short-term studies. There is a need to study secure email systems over longer periods of time—that is, months or years. Longitudinal studies could confirm whether the described design principles are sufficient for usable secure email or whether more improvements need to be made to support long-term usage.

Longitudinal studies could also be used to identify additional obstacles to the adoption of secure email. To date, researchers have relied on users' self-reported recollections to explain why users have not adopted secure communication tools.[1] Through the application of longitudinal, ethnographic studies, it might be possible for researchers to understand these obstacles better and identify new approaches for overcoming them.

Perceptions of the Necessity of Secure Email

There are obvious use cases for secure email, such as whistleblowing and protecting communication between dissidents. In other cases, the need is less clear. Although there is an argument that everyone should use secure email to prevent users who rely on its protections from being singled out, this argument finds little traction outside of academic circles.

For ordinary users, research has found mixed feelings regarding the necessity of secure email. In our research, secure email study participants consistently stated that they want the ability to encrypt their email but that they would use encryption only rarely.[7] When asked to explain, participants indicated that they only rarely need to transmit sensitive information and that they can leverage other communication tools when the need arises. The work of Abu-Salma et al.[1] supports this viewpoint, showing that users do not sufficiently understand how encryption works or how it protects them, limiting the urgency they feel in adopting secure email. Still, participants in our studies identified situations for which they desire secure email, such as submitting documents for a home loan, communicating with a doctor, and applying for a job. Interestingly, all of these cases involve short-term transactions.

Although the need for secure email in their personal lives is rare, users recognize its importance in business contexts. For example, Lerner et al.[6] found significant interest from journalists and lawyers, with both groups noting that it was often important to protect the confidentiality of their messages. Unfortunately, the operational constraints faced in business can also make it difficult to adopt end-to-end encryption—for example, lawyers often need to support email discovery, but many current secure email systems inhibit this.

Research Directions: Support Infrequent Usage

Although the community has succeeded in designing secure email systems that successfully onboard new users, it is unclear how well those systems support rare usage. Areas needing additional research include the following.

- *Protecting users from forgetting to enable encryption*: In our research, participants would sometimes forget to enable encryption until after they sent their email messages.[7] In our short-term studies, this was a learning experience and served to inoculate users from making the same mistake again. Still, with rare usage, we believe that this problem could recur frequently after periods of nonuse.
- *Handling lost keying material*: With rare usage, users are likely to lose their private keys—for example, forgetting the password for the private key or losing the device it was stored on—requiring them to generate and share new key pairs without revoking the original key pair. If systems allow this behavior, users could become habituated to ignoring key changes, putting them at risk for man-in-the-middle attacks. If systems disallow this behavior or make it difficult, the long-term usability of the systems is significantly impacted. Research is needed to understand how to handle this situation best.
- *Reacquaint users with secure email*: Users do not innately understand how secure email works[1] and need instruction on how to use secure email.[7] With infrequent use, users may forget how to use secure email over time. Research is needed to determine the design principles necessary to reacquaint users with correct secure email usage after periods of inactivity.

Research is also needed to better understand how to tailor secure email for specific business organizations and applications. Examples include 1) using secure email to enable communications that are compliant with the Health Insurance Portability and Accountability Act among hospitals, caregivers, and patients or 2) allowing secure email to support government-mandated discoverability requirements.

Easy-to-Use Key Management

Key management has long proven to be a significant obstacle in the usability of secure email.[10] Modern secure email research addresses this challenge by hiding the details of key management from users.

- The user's key pair is automatically generated[2,3,6] or downloaded[8] during installation.
- The user's public key is automatically uploaded to the appropriate key server.[2,3,6]
- A recipient's public key is automatically downloaded from the appropriate key server.[2,3,6,8]

Although hiding key-management details has obvious usability benefits, there are downsides to this approach. Most importantly, it inhibits the ability for users to understand how the system works, leading to a lack of trust and an increased risk of mistakes.[3,9] This can even lead users assuming that the system is providing them greater security then it is, leading them to send sensitive information that they should not—and would not if they better understood the security properties.

A clear example of this problem is secure chat clients (e.g., Signal, WhatsApp). These systems take transparent key management to its logical conclusion, providing a system that, to most users, is indistinguishable from an unencrypted session. Although these systems provide protection against passive attacks, they are vulnerable to man-in-the-middle attacks if users do not properly verify their contacts' public keys.

Unfortunately, in the interest of usability (and transparent key management), the need to perform this verification has been obscured from users.[9] Additionally, even if users do understand the need for verification, the process of manually verifying keys has significant usability issues, making it difficult—if not impossible—for many users to complete this security-critical step.[9] Thus, although automating key management may increase usability, it can impact real-world security.

Research Directions: Usable Key Management

Although the automatic generation and dissemination of public keys increase usability, most key-management-related usability issues remain unaddressed. As attention turns toward secure email's long-term usability and adoption, it is critical to address the usability of all phases of the key-management lifecycle.

- *Key verification*: Users need to be able to verify that the cryptographic keys used to sign and encrypt email correlate with the individuals they think they do. If this real-world binding were easy to establish, it could potentially provide significant protection against phishing. The work on key continuity by Garfinkel and Miller[5] is a step in the right direction, but much more needs to be done.
- *Key revocation*: Users need the ability to easily revoke keys that they lose or believe are compromised. These revocation mechanisms must not only permit the user to roll over to a new key but also must help the user's contacts verify and begin using the new key. Systems also have to help users differentiate between legitimate key loss and a malicious adversary claiming key loss to begin intercepting secure email messages.
- *User key management*: Users already struggle to keep track of a single key pair, and this problem is worse if they need to manage more than one key. Although in some cases the management of multiple keys can be hidden from users, in other cases, users must understand and, to some extent, self-manage their keys (e.g., keys for multiple email addresses or an email alias). Research is needed to build interfaces supporting this user-driven management of personal key pairs.
- *Key backup*: Users are prone to losing private keys, but they are not willing to lose access to their encrypted email. Although private key escrow provides a solution to this problem, most security experts agree that key escrow systems are undesirable. Additional work is needed to discover how to provide usable private-key backup and recovery.
- *Cross-device synchronization*: Users access email from multiple devices, and secure email needs to support this use case. Future research needs to identify the most secure and usable methods of synchronizing private keys across multiple devices and helping users track the storage of their private keys.
- *User understanding*: Users make better choices when they understand how their systems function.[1,3] Future work should explore how to help users build correct mental models, with an emphasis on helping users build mental models organically as they use secure email.[8]
- *Key server interoperability*: Current usable secure email research assumes that all users rely on a universal key server. This approach does not match the decentralized nature of email, nor does it help us understand the usability of secure email proposals that rely on decentralized key directories. More research needs to be done to understand and improve the usability of decentralized key servers.
- *Advanced features*: Users want more features in their secure email, and research is needed to understand how key-management schemes can support these features. Examples include delegating the ability to send/read a portion of a user's encrypted email (e.g., for a secretary

or a caregiver) and the ability to revoke access to a secure email message after a short period of time.

Conclusion: A Call to Action

Usable secure email has come a long way since Whitten and Tygar's original study,[10] with the pace of improvements recently intensifying. The research from the usable-security community has succeeded in creating secure email systems that are easy for first-time users, help users avoid mistakes, and have high perceived usability. Still, these systems have been tested only in short-term scenarios in the lab, and future research needs to explore the long-term usability and adoptability of secure email systems.

More importantly, the area with the most open research problems is usable key management. Many, if not most, of the obstacles to designing usable secure email directly arise from the challenge of providing usable key management. Although some parts of key management can be made transparent to users, this transparency does not come without risks, nor does it address all of the challenges for usable key management.

We believe usable key management represents a grand challenge for the usability–security community and, indeed, the security community at large. A lack of usable key management impedes not only the development and adoption of usable and secure email but also a significant number of other systems. For example, many blockchain systems require users to manage and store a wide array of key pairs. If this is difficult, there is a risk of losing billions of dollars. Similarly, proposals for key-based access control abound, but for these systems to be practical, users must manage, back up, and synchronize many key pairs. As these examples demonstrate, usable key management is an important issue that affects a wide range of security and privacy problems and is a topic that needs focused attention from the community. ■

Acknowledgments

We thank Daniel Zappala and the students in the research lab at Brigham Young University who contributed to our research on usable secure email. Kent Seamons is supported in part by the National Science Foundation grant CNS-1816929.

References

1. R. Abu-Salma, M. Angela Sasse, J. Bonneau, A. Danilova, A. Naiakshina, and M. Smith, "Obstacles to the adoption of secure communication tools," in *Proc. 2017 IEEE Symp. Security and Privacy*, 2017, pp. 137–153. doi: 10.1109/SP.2017.65.
2. E. Atwater, C. Bocovich, U. Hengartner, E. Lank, and I. Goldberg, "Leading Johnny to water: Designing for usability and trust," in *Proc. 2015 ACM Symp. Usable Privacy and Security*, 2015, pp. 69–88.
3. W. Bai, M. Namara, Y. Qian, P. Gage Kelley, M. L. Mazurek, and D. Kim, "An inconvenient trust: User attitudes toward security and usability tradeoffs for key-directory encryption systems," in *Proc. 2016 USENIX Symp. Usable Privacy and Security*, 2016, pp. 113–130.
4. J. Clark, P. C. van Oorschot, S. Ruoti, K. E. Seamons, and D. Zappala, Securing email. 2018. [Online]. Available: https://arxiv.org/abs/1804.07706.
5. S. L. Garfinkel and R. C. Miller. "Johnny 2: A user test of key continuity management with S/MIME and outlook express," in *Proc. 2005 ACM Symp. Usable Privacy and Security*, 2005, pp. 13–24. doi: 10.1145/1073001.1073003.
6. A. Lerner, E. Zeng, and F. Roesner, "Confidante: Usable encrypted email: A case study with lawyers and journalists," in *Proc. 2017 IEEE European Symp. Security and Privacy*, 2017, pp. 385–400. doi: 10.1109/EuroSP.2017.41.
7. S. Ruoti et al., "We're on the same page: A usability study of secure email using pairs of novice users," in *Proc. 2016 CHI Conf. Human Factors in Computing Systems*, 2016, pp. 4298–4308. doi: 10.1145/2858036.2858400.
8. S. Ruoti, J. Andersen, T. Hendershot, D. Zappala, and K. Seamons, "Private Webmail 2.0: Simple and easy-to-use secure email," in *Proc. 2016 ACM Symp. User Interface Software and Technology*, 2016, pp. 461–472. doi: 10.1145/2984511.2984580.
9. E. Vaziripour et al., "Is that you, Alice? A usability study of the authentication ceremony of secure messaging applications," in *Proc. 2017 USENIX Symp. Usable Privacy and Security*, 2017, pp. 29–47.
10. A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: A usability evaluation of PGP 5.0," in *Proc. 1999 USENIX Security Symp.*, vol. 348, 1999, pp. 169–184.

Scott Ruoti is an assistant professor in the Electrical Computer and Science Department at the University of Tennessee, Knoxville. His research interests include exploring how to increase the security of password-based authentication and two-factor authentication, using blockchain technology to secure non cryptocurrency systems, and how to help developers create secure software. Ruoti received a Ph.D. in computer science from Brigham Young University. Contact him at ruoti@utk.edu.

Kent Seamons is a professor in the Computer Science Department at Brigham Young University, where he directs the Internet Security Research Lab. His research interests include usable privacy and security, authentication, and key management. Seamons received a Ph.D. in computer science from the University of Illinois. He is an associate editor of *IEEE Transactions on Dependable and Secure Computing*. He is a Member of the IEEE, ACM, and USENIX. Contact him at seamons@cs.byu.edu.