Can Johnny Encrypt Now

Kate Ryan
University of California Berkeley
kate.ryan@ischool.berkeley.edu

Max Russell
University of California Berkeley
max_russell@ischool.berkeley.edu

1. Motivation

For each year in the intervening fifty years since email was introduced, it has only grown more popular and pervasive. Today, it is the common medium for internet-connected users to exchange messages. Security was not a design principle for the early versions of email, and as such it has been a bolt-on implementation. For most email users, there are gaps where emails could be read by motivated parties without the account owner's knowledge or consent; transport-layer security has reduced those gaps between email servers and to the clients, but actors at the server level could still read or modify emails in-flight without a user's awareness.

Cypherpunks in the 1990's developed public key cryptography tools to facilitate communication to other parties without any gaps in security. The most popular method uses PGP, or OpenPGP today, where a sender encrypts with a recipient's public key, and signs the message with their own private key. While sounding simple, the process is complicated enough and is fraught with possibilities for dangerous mistakes so adoption has been minute, despite the benefits. Previous work has highlighted the challenges developers need to address for users for their adoption, but more research needs to be done to see if the necessary work has been done to address shortcomings for users.

2. Research Question

This project aims to test whether modern implementations of encryption tools have made significant improvements in usability compared to the systems studied in Whitten and Tygar's landmark paper *Why Johnny Can't Encrypt*. In order for the technology to be considered usable, security tools must adhere to different benchmarks than typical software. Whitten and Tygar defined four priorities for security software in *Why Johnny Can't Encrypt*: users must be aware of the tasks they are needed to complete, users are able to figure out how to complete them, users don't make any dangerous mistakes during their actions, and users are comfortable enough with the interface to continue using it. For today's tools to be considered more usable than in the original work, they must perform better in aggregate across the four priorities.

The hypothesis going into the experiment was that modern tools would perform better across the four priorities than those used in the original experiment.

3. Related Work

Whitten and Tygar's *Why Johnny Can't Encrypt* is often considered the bedrock of security usability research, and provides the basis for the methodology in this experiment. It also outlines the four priorities that security tools must address for users, and shows how the existing tool that was usable by most standards was inadequate for general users. The study explains that conveying a mental model of the system quickly is paramount to users ability to perform tasks, which was absent in PGP 5. That it was possible for users to make dangerous irrevocable mistakes without any knowledge of the event was another major failing and breach of security principles. The paper showed for all security researchers how to approach performing usability testing of security-centric tools.

A long tradition of checking if PGP is usable yet, and follow-up paper by Sheng et al in *Why Johnny Still Can't Encrypt* is an example of this practice. This 2006 paper uses PGP 9 and Outlook Express 6.0, with the experiment methodology also having users creating keys, sending encrypted and signed emails, and decrypting and verifying incoming emails. Being a lab study, Sheng seeks to determine where errors in the security process can occur and if any discernible reasons exist for why the user made the error. Sheng et al determined that while better than before, tools need to more deeply integrate PGP with the email client, provide more prominent feedback and cues, and have interfaces dedicated to encryption-related tasks readily available for users.

A further look at email and PGP took place in 2016 when Routi et al evaluated a purpose-built encrypted email client Mailvelope in *Why Johnny Still, Still Can't Encrypt*. Mailvelope is a browser extension that allows for using PGP with email clients that do not have encryption settings, but found that users were unable to use the security tool. The lab sessions in Routi's study were much shorter than any other previous study at thirty minutes per participant, and required that two individuals were able to complete their encryption tasks in their window. This greatly increases the possibility of failure, as first time users often need a long window to perform their tasks and are helped if the other communication party has encryption set up properly. The study authors stated that integrating tutorials, conveying the public key cryptography mentals model to participants, and attempting to help remote parties through setup instructions in the tool would increase usability for users.

4. Methodology

Recruitment for the experiment was conducted by soliciting volunteers from friends and family. No major special considerations were given to selecting a diverse group of participants across age, gender, or career demographics. As email is used by everyone today, the ability for every person to use encryption tools is worth determining. Twelve participants were solicited to replicate the experiment in *Why Johnny Can't Encrypt*. Before beginning the experiment, participants were given an Informed Consent Document with details explaining what the study was about, the tasks in the lab, and risks of participation. Upon completion, a debrief was provided with contact information for feedback or questions, and information on what the data would be used for.

Measuring the ability for tools to address the security usability priorities involves determining the ability of participants to complete their tasks without error.

To provide an identical, controlled lab environment, a server was built that provided a virtual workspace preconfigured with the necessary tools to complete the task list. Kasm, an open source workspace streaming platform was selected to provide Docker-based Debian 12 “Bookworm” desktops with the Xfce4 desktop environment. Participants were able to access the Kasm server on the public internet from their own machines but perform the tasks on the virtual desktop environments. Each workspace was configured to automatically open Thunderbird 128, the email client participants were asked to use, along with a PDF with the task list for ease of reference. Thunderbird was configured before each participant’s session with individual access credentials to a unique account on an email server specially set up for the experiment. Participants were informed that all of the tasks were able to be and should be completed in the Thunderbird client, and no other applications were required for lab completion.

A typical email server was deployed on the internet using Postfix for SMTP and Dovecot for IMAP, both of which were used by participants to send emails internal to the lab domain, and receive emails from the keys.openpgp.org key server. The OpenPGP keyserver is configured as the only one usable in the Thunderbird client and was not a choice made for the experiment. Each email account had typical folders found on most email accounts corresponding to Inbox, Sent, Drafts, etc.

A special recipient email address and corresponding published public key, referred to throughout as Kryten, was set up for participants to encrypt emails to and decrypt emails from. A special Docker container was developed with access credentials to Kryten’s account, and monitored the inbox for emails from the lab participant email accounts. Upon receipt of a plaintext email, the daemon would respond in plaintext that the email was not encrypted and to try again. If an email was received that was properly encrypted to the public key for Kryten, the daemon checked the keyserver for published public keys several times over a short period. If no correctly formatted keys were published on the server, Kryten responded in plaintext that there were no valid keys published. Finally, if there were valid keys published and the email was encrypted with OpenPGP, the daemon responded with an encrypted and signed message telling the participant that the message looked good and thanked them.

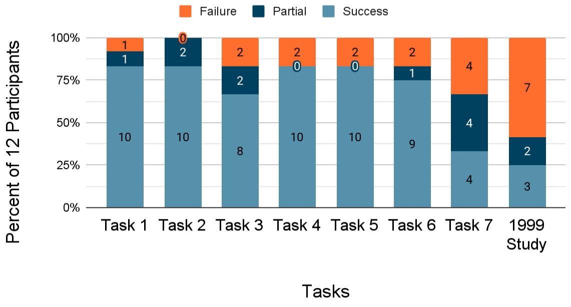
Seven tasks were outlined for participants to complete in their ninety minute lab window, with the instructions to complete them at the participant’s own speed and to the best of their ability. Each task was described in very few words and described the high level mechanisms required to send and receive encrypted email messages, but left the steps for each task up to the participant to determine and execute. The ability for participants to correctly discover the correct procedures was being measured to determine how usable Thunderbird is with OpenPGP encryption.

Finally, after completing the tasks or the allotted time was up, participants were asked several follow up questions. The questions included how familiar participants were with email encryption tools before the experiment and their confidence in their performance in completing their task list, on a Likert scale of 1-5. Additionally, a further free response question was asked to allow participants to highlight any challenges they perceived and how improvements could be made to address them.

5. Results

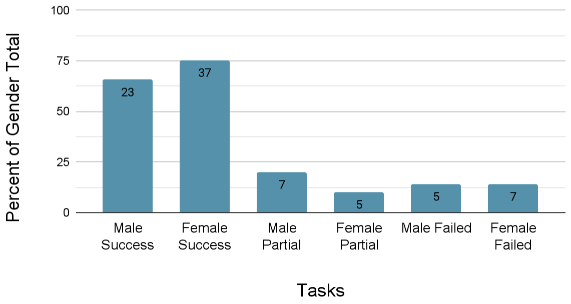
Overall, participants demonstrated a moderate level of success when using Thunderbird and OpenPGP for email encryption tasks. Across all users, tasks such as creating encryption keys (Task 1), publishing public keys (Task 2), and sending encrypted emails (Task 5) had the highest rates of successful completion, with 10 out of 12 participants completing these fully. However, Task 7, which involved exporting and backing up private keys, proved to be the most challenging (only 4 participants completed it successfully, while another 4 completed it partially and 4 failed entirely). Most users were able to complete at least five of the seven tasks with little to no assistance, indicating a notable improvement in usability compared to the results observed in prior studies like *Why Johnny Can't Encrypt*.

Fig 1: Study Result Comparison



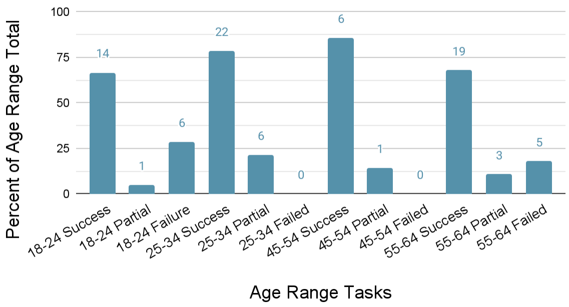
When breaking down performance by task, we observed that Tasks 3 (retrieving public keys) and 6 (decrypting and verifying messages) had mixed results. Task 3 saw 8 successes, 2 partial completions, and 2 outright failures, while Task 6 resulted in 9 successes, 1 partial, and 2 failures. Interestingly, despite being conceptually complex, Task 4 (encrypting and signing messages) maintained a relatively high success rate (10 successes), suggesting that Thunderbird may offer better guidance or more intuitive interactions for message-level encryption than for key management tasks like exporting or publishing keys.

Fig 2: Gender Results



In terms of demographic trends, participants in the 18–24 and 25–34 age ranges generally performed more consistently, with female participants in these groups completing nearly all tasks successfully. For instance, lab-user1 (F, 25–34) and lab-user5 (F, 18–24) completed all tasks, including Task 7. In contrast, participants aged 55–64 experienced more difficulties; lab-user3 (M, 55–64) failed five of the seven tasks, highlighting potential age-related usability barriers. Male participants overall showed more variance in success, with several partial and failed attempts across tasks. These findings suggest that while the tools may be more usable than in the past, there may still be a learning curve for older users or those less familiar with modern email client interfaces.

Fig 3: Age Results



6. Discussion

The results of this experiment support the idea that Thunderbird with OpenPGP has made usability improvements compared to earlier encryption tools. Most participants were able to complete core tasks like generating keys, encrypting messages, and sending email without major issues. These improvements suggest that Thunderbird's interface, default configurations, and integration with OpenPGP help reduce some of the friction that past users experienced in earlier systems.

Tasks like exporting and backing up private keys, however, remained challenging for many participants. Only a third completed this final step successfully, with the rest either partially completing it or missing it entirely. Since key backup is critical for maintaining long-term access and control, this indicates that key management continues to be a weak point in encryption usability. The lack of in-tool guidance or prompts for these less intuitive steps may contribute to their lower success rates and highlights an area for future improvement.

One surprising pattern was that participants who reported having the least technical knowledge or experience with encryption tended to perform better overall. We think this may be because they followed the prompts more literally and were less likely to second-guess what the tool was asking them to do. In contrast, participants with more technical backgrounds sometimes overthought the tasks, which led to avoidable mistakes. This finding suggests that overly technical users might bring assumptions that conflict with the tool's intended flow, while less experienced users may actually benefit from a simpler, step-by-step interface.

Additional usability trends emerged around the handling of public keys and interaction with the keyserver. Participants who struggled with exporting their keys often also had difficulty manually managing their keychain, particularly when it came to sending or retrieving public keys. This highlights a broader challenge with decentralized key management systems. The keyserver used in this experiment had relatively low maintenance, which may have contributed to some failures in key retrieval and weakened the reliability of the overall encryption process. That said, Thunderbird included helpful cues (most notably, a yellow bar that appeared at the bottom of the email client when a recipient’s public key was missing or not properly acquired). This prompt offered users a chance to “resolve” the issue with one click. Almost all users who saw this bar used it successfully, showing that good usability features like real-time guidance and clear error resolution options can make a significant difference. Expanding these types of in-context support features could further improve user success with encryption tools.

Age also seemed to play a role in task success. Younger participants, especially those in the 18–34 range, completed more tasks correctly, while the one participant in the 55–64 age group had more difficulty. These results line up with previous research showing that digital interface familiarity and age can influence success with security tools. They also reinforce the importance of designing for a range of users, including those who may not have grown up using this type of technology.

7. Limitations and Future Work

Soliciting volunteers from friends and family naturally hinders the accurate depiction of a more general population. The external validity of the experiment may be less representative than if recruitment was conducted with people from other areas. In future work, the same lab could be provided to participants from Amazon’s Mechanical Turk for example by embedding the Kasm desktop into survey pages. For similar levels of observation as was obtained during the over-the-shoulder lab experiment, more instrumenting of the lab would be required; Kasm supports recording the desktop and saving to a storage medium, which could serve as the instrumentation.

Participants were provided with a task list of what to do to achieve encryption; they were only meant to determine how to achieve those tasks. A better representation of real world conditions would be to provide users with a scenario where they would be encouraged to configure encryption, forcing users to use the tools at hand to create their mental model of the security system. The ability for a program to convey the tasks to users is a metric in the usability priorities, and a task list outlining the steps reduces the external validity of the experiment. With tutorials on how to configure OpenPGP readily available online, real world users may instead reach for that option to determine what to do once they know they need to begin using encryption. Participants of this experiment did lean on the task list to perform actions, and as such it would be prudent to perform further testing with multiple groups to determine to what level a task list influenced actions, as well as what users feel most comfortable with.

8. Conclusion

The results of this study suggest that while encryption tools like Thunderbird and OpenPGP have become more usable than the systems studied in *Why Johnny Can’t Encrypt*, there are still notable barriers to full adoption. Participants were generally able to complete tasks related to key generation, encryption, and sending messages with few issues, but struggled more with key management, particularly when asked to export and back up their private keys. These findings indicate that while usability has improved, certain areas remain unintuitive or under-supported.

One particularly interesting trend was that participants who self-identified as having low technical experience often performed better than those with more technical backgrounds. We believe this may be due to the fact that they approached the tool without preconceived notions or expectations and were more likely to follow interface prompts at face value. This finding reinforces the idea that encryption tools must be designed not just for technical users, but for anyone who may need to communicate securely.

While the participant pool was small and drawn from a limited demographic, the experiment shows encouraging signs that modern encryption tools are moving in the right direction. Future research should explore these findings in broader populations and focus on improving support for tasks that still cause friction, particularly around key management and long-term security behaviors.

9. References

Alma Whitten and J.D. Tygar, Why Johnny Can't Encrypt: A Usability Case Study of PGP 5.0. Proceedings of the 8th USENIX Security Symposium, August 1999.

Sheng, S., Broderick, L., Koranda, C. A., & Hyland, J. J. (2006). Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software. https://cups.cs.cmu.edu/soups/2006/posters/sheng-poster_abstract.pdf

Ruoti, S., Andersen, J., Zappala, D., & Seamons, K. (2016, January 13). Why Johnny Still, Still Can't Encrypt: Evaluating the Usability of a Modern PGP Client. https://arxiv.org/pdf/1510.08555

10. Acknowledgements

Special thanks goes to the myriad similar experiments that inspired this work, and CYBER 215 for the contextual discussions that framed the research results.

Appendices

INFORMED CONSENT DOCUMENT

Study Title: Can Johnny Encrypt Now?
Researchers: Max Russell & Kate Ryan
Institution: University of California, Berkeley
Contact Info: max_russell@ischool.berkeley.edu & kate.ryan@ischool.berkeley.edu
What's This Study About?
Thank you for considering participating in this study! This research is all about testing how user-friendly modern encryption tools are compared to the ones studied in Why Johnny Can't Encrypt. We want to see if today's tools (Thunderbird and OpenPGP) are easier to use and help people avoid common mistakes.
What You'll Be Doing:
If you decide to participate, you'll be asked to complete a series of tasks using an email client (Thunderbird) and an encryption tool (OpenPGP). These tasks include:

- Creating encryption keys
- Sending and publishing public keys
- Receiving and verifying public keys
- Encrypting and digitally signing messages
- Sending and receiving encrypted messages
- Decrypting messages and verifying signatures
- Exporting and backing up private keys

You'll be given a scenario where you're acting as a campaign coordinator, using encrypted email to update your team. You'll be working on a lab computer (a virtual desktop hosted on AWS), and we'll be observing how you interact with the system. The whole session will take about 90 minutes.
Any Risks?
The biggest risk here is potential frustration if the software is tricky to use. You are not being graded, and there are no wrong answers. If at any point you want to stop, you can.

What's In It for You?
There's no direct benefit for participating, but your feedback will help improve the usability of encryption tools, which could make online security easier for everyone.
Keeping Your Info Safe:
We're not collecting any personal data, and all findings collected will be anonymized. Your responses and interactions with the software will only be used for research purposes.
Totally Voluntary:

Your participation is completely up to you. If you start and decide you don't want to continue, you can stop at any time.

Questions?

If you have any questions, feel free to reach out to Max or Kate at the emails listed at the beginning of this document. By signing below, you're saying you understand what this study is about and that you agree to all the terms listed above.

Participant Name (Printed): ________________________________

Participant Signature: ______________________________________

Date: ________________

Researcher Signature: ______________________________________

Date: ________________

Thanks so much for your help!

Task List

During this experiment, you as the participant will be expected to perform the following tasks in the Thunderbird email client:

i. Create Keys
ii. Publish Public Keys
iii. Get Public Keys
iv. Encrypt Message/Digitally Sign
v. Send an Email Message to kryten@cybercoop.xyz
vi. Decrypt Message/Verify Keys
vii. Export and Backup Private Keys

Debrief Handout

STUDY DEBRIEF DOCUMENT

Study Title: Can Johnny Encrypt Now?

Researchers: Max Russell & Kate Ryan

Institution: University of California, Berkeley

Contact Info: max_russell@ischool.berkeley.edu & kate.ryan@ischool.berkeley.edu

Thank You!

First off, thank you so much for participating in this study! Your time and effort are really appreciated, and your feedback will help us better understand the usability of modern encryption tools.

What Happens Next?

We'll be analyzing the data from all participants to identify trends (things like how many people were able to complete each task successfully, where common mistakes happened, and whether users felt confident using the tools). The findings will help inform how encryption tools can be improved in the future.

We'd Love Your Feedback!

Please complete our brief survey (it takes no longer than 5 minutes). Your input can provide extra context for our analysis, and we greatly appreciate your feedback!

SCAN FOR SURVEY

Questions?

If you have any questions about this study or would like to know more about the results once they're
available, feel free to reach out to Max or Kate at the emails listed at the beginning of this document.

Thank you again for being part of this study!

Debrief Survey

This survey aims to gather insights into participants' experiences while taking part in this study. We seek to understand their confidence in their performance and any challenges or feedback they may have.

1. First Name, Last Name
2. How knowledgeable would you consider yourself about email encryption tools before completing this study? (1 = Not knowledgeable at all, 5 = Very knowledgeable)
3. On a scale of 1-5 (1 = Not Confident, 5 = Very Confident), how confident are you that you successfully completed the tasks provided to you during this exercise?
4. What challenges, if any, did you encounter while attempting to complete the assigned tasks, and how do you think the process could be improved?