Why King George III Can Encrypt

Wenley Tong, Sebastian Gold, Samuel Gichohi, Mihai Roman, and Jonathan Frankle

Abstract

We sought to re-examine the conclusions of the classic paper *Why Johnny Can't Encrypt*, which portrayed a usability crisis in security software by documenting the inability of average users to correctly send secure email through Pretty Good Privacy (PGP). While the paper's authors primarily focused on user-interface concerns, we turned our attention to the terminology underlying the protocol. We developed a new set of metaphors with the goal of representing cryptographic actions (sign, encrypt, etc.) rather than primitives (public and private keys). Our objects were chosen such that their real-world analogs would correctly represent the security properties of PGP. Since these metaphors now corresponded to physical actions, we also introduced new forms of documentation that explored narrative techniques for explaining secure email to non-technical users. In quiz-based testing, we found that, while our new metaphors did not dramatically outperform traditional PGP, we were able to convey equivalent levels of understanding with far shorter documentation. Subsequent lab testing confirmed that metaphors with physical analogs and the accompanying briefer instructions greatly eased the process of using secure email. Our results indicate that crafting new metaphors to facilitate these alternative forms of documentation is a fruitful avenue for explaining otherwise challenging security concepts to non-technical users.

1 Introduction

Over the past 15 years, the findings of Whitten and Tygar in the now-classic paper *Why Johnny Can't Encrypt* [6] have become standard wisdom in the privacy and security communities. The paper's conclusions - that existing software made mathematically secure protocols unusable and therefore useless to the average user - devastated the cryptographic triumphs that secure email had promised. *Johnny*'s publication provoked a paradigm shift in the research community, generating renewed emphasis on usability studies and user psychology. The paper demonstrated that mathematical proofs were not sufficient to spur widespread adoption of privacy preserving technologies. Comprehensibility and usability, it asserted, were prerequisites rather than afterthoughts, and existing solutions failed to meet these standards.

The paper's central study challenged non-technical users to perform a series of tasks using Pretty Good Privacy (PGP) for email confidentiality and authentication. Users had to send and receive secure emails, verify signatures, and wade into the challenging territory of public key directories. The study revealed that, for all of its cryptographically guaranteed security, PGP was nearly impenetrable for those without technical backgrounds. In a series of simultaneously hilarious and horrifying anecdotes outlining user failings, the paper summed up two major flaws in the state of PGP at the time of its writing: (1) the then-standard user-interfaces were unwieldy and difficult to understand and (2) the PGP metaphors (private and public keys, signing, etc.) and their visual integration into user-interfaces served only to obscure cryptographic actions.

The research literature since the paper's publication has comprised a combination of echoes of *Johnny*'s pessimism and a rethinking of user-interface design in other systems like web browsers. Few, if any, studies exist examining the second of *Why Johnny Can't Encrypt's* criticisms: the weakness of security metaphors. Those that do, like Whitten's own dissertation [5], focus their energy on improving the visual language describing existing PGP metaphors rather than on honing the metaphors themselves. Further, while *Why Johnny Can't Encrypt* presented a variety of compelling anecdotes, its primary mode of proof was a lab test involving 12 participants, hardly a statistically significant enough number to support the broad conclusions that the paper implies.

In our study, we endeavor to reopen the questions that, in the eyes of the research community over the past 15 years, *Johnny* seems to have closed. Specifically, we reexamine *Johnny*'s conclusions with the statistical rigor that the original paper lacked. Our focus is on whether better physical metaphors and conceptual methods of presenting email security to non-technical users can clarify the process of sending secure email to the point where users can (1) reason about the security properties of the system with cryptographic accuracy and (2) send secure emails easily. While previous studies have attempted to better adapt the visual language of PGP user-interfaces to the existing set of metaphors (public and private keys, signing and verifying, etc.), we investigate ways to improve and replace this underlying terminology. The current PGP metaphors were intended to be understood by researchers rather than

non-technical users; we feel they present an unnecessary obstacle to end users and user-interface designers alike.

To that end, we developed a new set of metaphors that clarify the cryptographic actions necessary to send and receive secure email and provide users with a framework in which to reason about the security properties of behaviors and attacks outside the normal process of secure communication. We also explore new modes of introducing these metaphors and secure email to users, capitalizing on the physical behaviors that our metaphors conjure and associated mental models to give users intuition about the implications of dangerous actions (Section 3). We created a quiz that assesses a user's ability to reason about secure email given a basic overview of security primitives in the language of both the traditional and new PGP metaphors (Section 4). By releasing this study on Mechanical Turk, we were able to rigorously assess our results on a scale well beyond that of *Why Johnny Can't Encrypt*. Finally, in a nod to *Johnny*, we built a user-interface and conducted user tests with both sets of metaphors to gain an understanding of how well our new metaphors functioned in the wild (Section 5). We close with a discussion of the tremendous opportunities for future research and our conclusions (Sections 6 and 7).

Our contributions are two-fold. First, we provide concrete metaphors for PGP and accompanying documentation that have demonstrated success in user testing. Second, and more importantly, we have expanded the design space for communicating complex security protocols to users in a concise and comprehensible manner, showcasing a variety of techniques applicable across any system to improve user understanding of security concepts.

2 Related Work

2.1 PGP Usability

The classic study of PGP usability is Whitten and Tygar's 1999 paper, *Why Johnny Can't Encrypt*, which aimed to explore the usability properties of then-current PGP software [6]. The paper included both a cognitive walkthrough of PGP 5.0 and a 12-person user-test in which subjects were asked to perform the tasks necessary to send and receive secure email. Only a third of the study's subjects succeeded, which the authors attributed to user-interface deficiencies, poor software integration, and a visual language that confused the already opaque cryptographic metaphors.

It seems that, in the wake of the publication of *Why Johnny Can't Encrypt*, researchers unquestioningly accepted the paper's conclusions and almost entirely abandoned further investigation into PGP usability. The only follow-up investigation we were able to find was a small study called *Why Johnny Still Can't Encrypt* that repeated the experiment of the original paper using Outlook Express and an updated version of PGP [3]. As the title suggests, the investigation's results largely upheld the original paper's conclusions, although the researchers attributed most user-failings to specific aspects of the Outlook and PGP interfaces rather than the larger, overarching deficiencies that the original paper had noted.

Some research energy did go into examining other secure email systems that emerged in later years. In 2005, a study systematically recreated the original *Johnny* experiment on a far larger scale in order to evaluate the usability of S/MIME, a protocol for secure email similar in nature to PGP [2]. The researchers hypothesized that *Johnny*'s findings were due to PGP's poor certification model rather than UI or metaphor failings. To test this theory, they examined S/MIME in concert with a trust-on-first-use method of key certification called Key Continuity Management (KCM). S/MIME is more tightly integrated with email clients like Outlook and KCM allows clients to automate S/MIME tasks when communicating with known contacts, eliminating an entire class of problems described in *Why Johnny Can't Encrypt*. The usability experiment was run both with and without KCM and included attacks that users had to identify and avert. The team concluded that S/MIME and KCM were vastly more usable than PGP, but that KCM introduced a new category of identity-based attacks.

A further study revisited the conclusions of the S/MIME analysis in 2012, performing a series of cognitive walkthroughs of then-current S/MIME software with various "personas" [1]. The authors determined that S/MIME remained challenging for the average user to grasp due to varying terminology across a fragmented cryptographic software ecosystem. Specifically, certificates are used for so many different functions that management software tends to be general purpose, integrating poorly into secure email workflows.

The most comprehensive study in the area of secure email usability appears to be Whitten's own dissertation, which was completed five years after *Why Johnny Can't Encrypt* [5]. In this paper, Whitten examines several techniques for improving user comprehension of security user-interfaces. She focuses specifically on *staging*, in which users are slowly introduced to progressively more functionality as they are ready for it, and *metaphor tailoring*, in which visual metaphors are crafted in order to leverage existing mental models that correctly mirror security properties. As an example of the latter, Whitten represents public and private keys as fitting together into a single unit resembling a yin and yang. She developed an email client called Lime based on these principles and conducted user studies similar in nature to those in her original paper, determining that staging and tailoring improved the usability of public key encryption to the point where it was broadly comprehensible.

2.2 Metaphor Improvements

We struggled to find instances where researchers specifically targeted improving metaphors as a direct method of making systems more coherent to non-technical users. Many papers propose general principles for designing usable security systems, but none that we could find go through the exercise of constructing new metaphors based upon these ideas. One of the few exceptions is Whitten's dissertation, but her focus is on better visual representations of existing metaphors rather than creating entirely new metaphors.

Our study also attempts to improve the mechanisms for explaining metaphors to users. One of the few papers in this area describes a website dedicated to internet security education via comic strips [4]. Although we did not have the resources to apply this approach in our own study, this tactic represents the logical extension of our use of narratives to present security metaphors in a more concise and accessible manner.

3 Reconsidering PGP Metaphors

3.1 Flaws in Existing Metaphors

PGP makes use of the original RSA metaphors: *public keys* and *private keys*. These two titles, which accurately represent the cryptographic functions of the underlying objects, defy any form of real-world intuition. Are all keys not private? Why would a key be public? How can my public key "lock" (encrypt) a message but not "unlock" (decrypt) it? How do I sign something with a key? And worst of all, how in the world can I verify someone's identity with the same device I use to encrypt messages addressed to them? It is no wonder that we assume - and *Why Johnny Can't Encrypt* asserts - that users without technical backgrounds struggle when forced to reason about PGP's security properties using existing metaphors.

The name "key" originally referred to the secret in symmetric cryptography and was adopted for use in asymmetric cryptography when the former was already well-established. While the real-world semantics of the word "key" apply to the intent and purpose of keys in symmetric protocols like AES, RSA keys are not used in this fashion. This is the disconnect that we attempt to bridge in this study.

3.2 Our Goals

We aim to devise a set of metaphors that fully preserve a user's ability to reason about the security properties of PGP, Tor, HTTPS, or any other system that makes use of asymmetric encryption. These metaphors should be intuitive and readily accessible to a user who lightly skims the documentation before attempting to make use of the system. We hypothesize that, in spite of the pessimism that permeates existing research literature, these two properties can readily coexist.

3.3 Action-Based Metaphors

There are several different levels at which one can aim a set of metaphors for a system like PGP. The lowest is at the level of cryptographic *primitives*: public and private keys. Using metaphors at this level requires imparting a thorough understanding of the underlying mathematical processes; without this knowledge, a user has no basis for intuiting about the metaphors.

At the opposite end of the spectrum are metaphors that focus on the cryptographic *properties* like confidentiality and authentication. Example metaphors might include sending a message on special, unforgeable watermarked paper (authentication) in tamper-proof envelopes (confidentiality). The main weakness of this approach is that it limits a user's ability to reason about any scenario except the proper functioning of the intended application. That is, allowing a user to understand best practices for handling the loss of a private key, creating a new cryptographic identity, or distributing a public key become as convoluted as coming to terms with the primitives themselves. Simply put, property-level primitives mask too much underlying functionality to meet our goals.

We elected to aim our metaphors between these two extremes: at the level of *actions* like encrypt, decrypt, sign, and verify. This level provides the best of both worlds: a low-level picture of the process of sending secure email that is still conducive to intuitive metaphors. We chose to separate the dual functionality of public and private keys (encryption and signing) into distinct metaphorical objects, which allows special-purpose objects to correspond to each action at the cost of obscuring only the very lowest-level mathematics behind asymmetric encryption.

We present the user with four items, a *key*, *lock*, *seal* and *imprint*. The key and lock serve the purposes of encryption: Alice distributes her locks as widely as possible so that others can send her messages that only she can open with her key. Similarly, the seal and imprint handle signing: Alice passes out copies of her imprint so others can verify her as the sender of messages she has stamped with her seal. Collected together, we refer to these four items as a *toolkit*; this abstraction handles the contingency where a user loses her key but not her seal: we insist that the toolkit represents an indivisible unit that must be replaced whenever any element is lost.

One debate we could not resolve ourselves was the choice of terminology for the seal. We also considered referring to the object as a *stamp*. The difficulty of either choice is that a seal or stamp can describe the object that produces a mark, the mark that it leaves, and the action of producing the mark. In early testing, we called the pair of objects the *seal* and *seal imprint*, but we found that avoid-

ing repetition of the word *seal* and instead referring to the latter as the *imprint* was more clear.

3.4 Modes of Presentation

Nearly as important as the metaphors themselves is the method of presenting them to users. Convoluted or sparse documentation can undermine otherwise effective metaphors, and - conversely - clear and concise explanations can clarify potentially confusing topics.

For both our new metaphors and those in standard PGP, we developed a lengthy, thorough document intended to brief users on everything they might need to know about sending secure email. These guides covered encrypting, decrypting, signing, and verifying in the appropriate terms of the metaphors. In addition, we described several failure and attack scenarios and proper user responses to these conditions. The version with our new metaphors is available in its entirety in Appendix A.1.

We are interested, however, in whether alternative modes of presenting secure email beyond this conventional documentation might enhance user understanding of metaphors. We developed another set of documentation of our new metaphors couched in the terms of a fictional historical narrative. Such an introduction is nearly impossible with existing PGP metaphors, which lack physical analogs. Our new metaphors, however, are quite compatible with the universe of British King George III and his colonial empire.

We hypothesize that this narrative approach has two major benefits over traditional documentation. First, it presents the metaphors in the form of an example that users can emulate. By watching how they work in fictionalized practice, users gain an immediate understanding of how to send and receive secure email. Although Alice and Bob are well-worn instances of this strategy, we believe that the immediacy of a colonial rebellion will better adhere the metaphors in users' minds.

Second, by experiencing the metaphors in the context of a fictional universe with internally consistent rules and behaviors, we conjecture that users are able to reason about scenarios beyond those described in the documentation (i.e., the security properties of losing a key or seal). This fact does place an enormous burden on the designers of this universe, as the rules that we create must tightly align with the actual cryptographic properties of the system. We believe, however, that the resulting user comprehension should be far more robust than if we merely enumerated every possible real-world scenario and dictated the appropriate responses.

Beyond these benefits, we also found that a narrative form was far shorter than the corresponding traditional documentation. This result may seem counterintuitive at first glance, but a narrative approach obviates exhaustive exposition on the security properties of various objects by simply demonstrating their use in practice. A fictional narrative also seems more inviting in comparison to the dry, technical documents that usually introduce users to new software. While it is difficult to measure the extent to which users skim or completely ignore documentation, we reason that making it shorter and more exciting, as the narrative approach does, will ensure that more users actually consult the documentation before attempting to send secure email.

We wrote three versions of this narrative in order to test how far we could stretch the model. The first (Appendix A.2) attempts to capture the same level of detail as the conventional documentation within the fictional context. We relate the motivation that compelled King George to demand secure communication, the process by which he used the system, and the security properties that each of his tools guaranteed. We specifically tailored this passage to cover as much material as the traditional approach did. Our second attempt (Appendix A.3) is a single excerpt describing King George sending and receiving secure letters using his four tools as if pulled directly from a spy novel. The depiction is concise and compelling enough to please the laziest of readers and should push the limits of a user's ability to intuit from a fictional universe. Rather than making everything explicit as we did in the previous two approaches, this passage requires readers to infer the security properties of the system. We adapted an additional version of the narrative approach for use in user-interface based testing (Appendix A.4).

Another possibility we considered but, due to time constraints, could not implement included embedding the fictional narrative in a comic strip. Beyond potential entertainment value and the benefits inherent in replacing large blocks of text with pictures, a comic could introduce the visual language of metaphors that would later appear in an accompanying user-interface.

4 Quiz-Based Testing

In order to efficiently test a large group of people in a short time frame, we used Mechanical Turk to run our quizzes on our subjects. Mechanical Turk workers were required to have an approval rating of at least 99% and have at least 1,000 approved Human Intelligence Tasks (HITS).

In order to control for pre-existing knowledge, we first required each worker to complete a pre-questionnaire that assessed his or her:

- General technical knowledge and expertise in programming and mathematics
- Existing cryptography experience
- Existing knowledge of PGP, GPG, RSA, or public-private key encryption
- Experience, if any, with sending encrypted emails

After completing the pre-questionnaire, subjects were given one of several possible descriptions of how se-

cure email (i.e., PGP) works. After reading the provided description, subjects completed a final questionnaire designed to measure their mastery and understanding of secure email. Subjects were permitted to re-read the description while completing the comprehension assessment.

We wrote a total of five different versions of descriptions from which subjects could learn. The full texts of three are available in Appendix A (two others were close derivatives of other passages).

PGP Full — A thorough, in-depth explanation and walkthrough of PGP described in the terminology of public and private keys.

Metaphor Full — A line-by-line transcription of the PGP Full description using our metaphors of keys, locks, seals, and imprints (Appendix A.1).

Metaphor Long Story — A page-long narrative describing the objects and their properties contextualized by a story about King George III sending letters securely (Appendix A.2).

Metaphor Short Story 1 and 2 — A two-paragraph description that does not explicitly explain any object or its properties. Instead, it depicts a pair of scenes meant to resemble an excerpt from a historical spy novel portraying how King George III uses the four objects. This passage was revised into a second version after we realized many people were confused about the ownership model for locks and keys, giving us: the original (1) with sending and receiving with a single recipient and the modified (2) with sending and receiving with two recipients (Appendix A.3).

4.1 Quiz Design

We designed the comprehension assessment to measure subjects' understanding of the material without having the questions themselves give prompting information about the right answer. For example, the question "Whose public key does the sender use to encrypt a message?" leaks a number of pieces of information: single-ownership of a public key, the fact public keys are used to encrypt messages, and the fact the sender does the encryption with that public key. In an ideal world, all questions would be free-response to give subjects maximum flexibility to make errors and mislead themselves, since users of personal security software will not have others to prompt or correct them. For ease of comparison and quantifiability, however, we made most questions multiple-choice with as wide a variety of plausible responses as possible.

The full text of all the questions along with a description of the information being tested is available in Appendix B.1.

4.2 Results

The raw success rates for each question by each test group are available in Appendix B.2.

We discovered a number of interesting facts as a result of our testing. They are grouped by overarching themes in the sections below.

4.2.1 The Surrounding Model

The first group of challenges we observed centered on conveying all of the nuances of the PGP threat model in their entirety. A surprisingly high percentage of subjects did not internalize the importance of verification in making messages secure, despite the fact that we explicitly noted this condition in our longer form descriptions and were careful to reference it in the narratives. This failing may be a product of the particular terminology we used in both the descriptions and questions: in lay language, the word "secure" conveys confidentiality and integrity but not authenticity. In the technical vernacular, secure communication does entail verifying authenticity, but this meaning appears to be lost on non-technical users. Perhaps introducing these properties individually and devoting a sentence to an explanation of the importance of each term might better clarify the threat model.

Across all test groups, approximately 40% of subjects erroneously thought using secure messages protected them from message forwarding by the recipient. Initially, we were unsure how subjects came to this conclusion since, even in the physical world, the recipient of a letter can copy and distribute it. We conjecture that the question might have been interpreted as the recipient forwarding the box with the letter still locked inside. If the documentation and narratives had made references to a double-agent, subjects might have better understood the implicit trust in the recipients of secure email.

Subjects given a narrative completely misunderstood the trust model upon which PGP is built: a reliable, but insecure transmission channel with trusted endpoints. Over 60% believed at least one of either your ISP or Wifi network had to be secure in order to send secure messages. Subjects given one of the two technical descriptions performed much better, with under 20% making the same error.

4.2.2 The Ownership of Metaphorical Objects

Another major misunderstanding involved the ownership model of the objects. When asked which items to use when sending a message, many subjects thought recipients held copies of a key and locks were kept secret. Given trusted recipients, this supposition makes some sense, but structuring a secure messaging network in this fashion should seem less secure even using the physical analogs of the metaphors. We attempted to clarify this misconception

Answer Choice	True?	PGP	Metaphor	Long	Short 1	Short 2
$n =$		40	40	20	20	20
(Select None)	N	0.0	0.025	0.0	0.0	0.0
Future Send	N	0.325	0.20	0.36	0.44	0.36
Past Send	N	0.65	0.30	0.44	0.56	0.44
Future Receive	Y	0.70	0.675	0.80	0.64	0.76
Past Receive	Y	0.975	0.875	0.64	0.76	0.92
Impersonate You	Y/N*	0.90	0.225	0.20	0.28	0.24
Impersonate Others	N	0.30	0.125	0.08	0.20	0.20

Included Own Key as requirement to Send					
Answer Choice	True?	Metaphor	Long	Short 1	Short 2
$n =$		11	16	17	12
(Select None)	N	0.0	0.0	0.0	0
Future Send	N	0.27	0.25	0.59	0.42
Past Send	N	0.11	0.38	0.65	0.50
Future Receive	Y	0.55	0.81	0.76	0.66
Past Receive	Y	0.82	0.63	0.82	0.92
Impersonate You	Y/N*	0.36	0.25	0.29	0.33
Impersonate Others	N	0.27	0.06	0.18	0.25

Table 1: Summary of answers selected as true by subjects when asked, "If someone else acquires your [key/private key], what will they be able to do? Check all that apply." *Note that our metaphor model separates the signing operation from the decrypting operation, the two functions performed by the underlying private key.

in the second narrative, but without much success. We might have been able to better test this understanding with a re-wording of the question to clarify our interest only in the objects the sender, not the recipient, needs. That said, when we asked subjects to explain the process of sending a secure message themselves, most still used the objects in the correct manner even if they had incorrectly identified whose objects to use previously. The most common mistake when providing steps to secure a message was forgetting to sign/seal the message. It is therefore likely that the wording of the question, not user understanding, was to blame for low accuracy on this topic.

4.2.3 Replacing Lost Objects

While the three long-form descriptions had over 90% success recognizing the need to make a new toollkit when a key/seal or private key is lost, the first and second short narratives had 72% and 56% success in this area. This is most likely due to a lack of explicitness of irrecoverability in the narratives. This also explains why subjects did not include exchange of public objects as a necessary step when a party loses his or her toolkit. These two actions were presumed to have been performed before the events of the short narratives, meaning they were not directly referenced in the passages. We might consider adding a sentence or two to address this shortcoming.

In the short narratives, a common response to a friend losing his or her toolkit was to send a new one, revealing a mental model of a single creator of toolkits who distributes them to the people with whom he or she wishes to correspond. The ability for anyone to make a toolkit is a property that was markedly absent from the short descriptions, and, especially in the longer narrative, it makes sense to envision a central creator of toolkits. This flaw, again, could likely be corrected with minor revisions to the narrative itself.

An interesting misconception was the idea that each person manufactures boxes with locks built into them, distributes those, and holds onto the corresponding keys. From a security standpoint in our fictional world, this is equivalent to, but not the same as, having generic boxes with people making and distributing locks.

4.2.4 Understanding Thefts and Copies

The proportion of subjects who correctly identified all the implications of a key copy was surprising low. Table 1 shows the aggregation of answers selected for each of the five descriptions. Generally, most people understood the affirmative cases: that a copy of your key enables another person to read messages sent to you, both in the past and in the future. We can attribute confusion about reading sent messages (both past and future) to a misunderstanding of the ownership model, but the section of Table 1 that summarizes results for subjects who said the sender's key is required to send a secure message shows that the error rates are not different enough to support this hypothesis. Since the errors present a more paranoid view of the world than is necessary, this is probably acceptable from a security standpoint since it will drive users to be even more cautious than is necessary.

4.3 Summary of Quiz Testing

Overall, our quiz testing showed that our metaphors did not significantly outperform standard PGP metaphors in terms of quickly bringing correct intuitions about the protocol. There are simply too many elements in the assumptions and setup that are not intuitive and still need explanation. However, we were able to get a comparable level of learning between a full technical explanation of the standard PGP terminology and our metaphors, even with just a short example of usage.

Across all our test groups, there were a number of key concepts that were often confusing and not fully comprehended by subjects. The first is the overall threat model: having a reliable but insecure transmission channel and trustworthy endpoints. The second is the ownership model of the metaphorical objects: using the recipient's lock when sending so that there is only one key per set of locks. Both are apparently not the first assumption

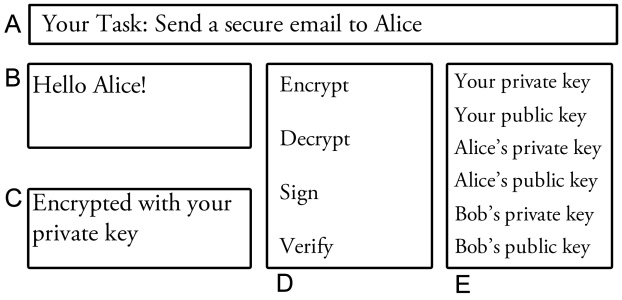

Figure 1: A sketch of the user-interface that we implemented for our lab testing. In this example, PGP metaphors are shown. (A) A simple task for the subject to complete. (B) The original message that the subject must somehow transform into the desired result. (C) A log of all actions the subject has performed so far. (D) Actions for the subject to apply. (E) Objects for the subject to use with an action.

made by the majority of subjects, but are critical to fully understanding the protocol.

We believe our metaphors will also be easier for people to remember once learned, but future research with repeat assessment is necessary to test this hypothesis.

5 User-Interface Based Testing

5.1 User-Interface Design

We designed a basic user-interface in order to test our metaphors and documentation approaches in a more realistic setting. Our goal was to examine the efficacy of the metaphors themselves, not the cosmetic design of the user-interface. To this end, we created a very simple user-interface that could easily be used interchangeably with original PGP terms or our own new metaphors.

A sketch of the user-interface is visible in Figure 1. We had a space on the screen to show the message being manipulated and a log of all actions taken. In the middle, there was a column of actions (encrypt, sign, etc). On the right, there was a column of objects (Alice's public key/lock/imprint, My private key/key/seal, etc.). We showed all objects that existed in our small world of Alice, Bob, and the current user - even those to which the user would not have access (e.g., Bob's private key). The correct terminology was substituted depending on whether we wished to use PGP metaphors or our own.

The goal of this exercise was to assess user comprehension in person and in a real-world setting. We allowed subjects to make mistakes that would not even be possible in real life (i.e., using Alice's private key) to avoid leaking answers to subjects. Rather than simply checking for correctness, we sought to gather insight into the thought process of our test subjects. Facilitating user misunderstandings and mistakes was therefore essential for us to analyze our metaphors with a critical eye. We could far better understand the weaknesses of our metaphors and documentation by observing this behavior in person as opposed to inferring from multiple-choice quizzes.

5.2 Testing Methodology

We wrote a script by which interviews should be performed for user-interface testing for consistency. We had two versions of the user-interface. One used our metaphors while the other supplied the PGP terminology (public and private keys). The steps involved in user-interface testing were:

1. Read, or have the subject read, one of the instructions. This could be the formal instructions, the in-depth narrative about King George III, or the abbreviated narrative about King George III. Note that we prepared a version of the abbreviated narrative specifically for this task (Section A.4).
2. Have the subject perform tasks on the user-interface. The correctness of his or her actions was judged manually by the interviewer. The tasks involved sending or receiving a secure email. Subjects were then verbally asked about the significance of potential markers of attack, like a missing or invalid imprint on a message. Lastly, they were interviewed about the significance of losing different parts of their toolkit as well as what actions they would have to take as a result.
3. Show the subjects the other two forms of documentation and ask for feedback on the methods of presenting instructions.

Interviewers would answer questions asked by the subjects, but would not give away the solutions. The goal was to simulate the ability to search for general explanations about the protocol but not specific answers to particular tasks. In some cases, small hints were given if the interviewer deemed these helpful for the process of the study. These were noted and included in the results.

We used this regimen to test a user-interface emphasizing the aspects of secure email that we aimed to improve with our metaphors and documentation. Some aspects that were not tested, such as key management, were intentionally ignored because they pose a significant research problem beyond the scope of this paper.

5.3 Results

We interviewed a range of subjects varying broadly in age and technical experience. Preference for methods of instruction seemed to vary based on personal styles of learning, but people seemed to generally favor the new metaphors. Some users even referred back to our new metaphors in order to clarify their understanding of the interface with the PGP terminology. A general comfort

with technology or an ability to follow precise directions seemed to be required for success.

Some particularly interesting results came from two elderly subjects. Neither of them self-identified as technologically savvy, but both seemed intelligent and willing to learn. Both were given the short narrative as a method of instruction. One of them had some trouble, but still managed some tasks correctly with a little coaxing. When attempting to send an email, she struggled to understand the user-interface workflow: she thought that by clicking "sign" and selecting Alice's private key, she was signing the letter and locking it such that only Alice could open it. We partially attribute this misunderstanding to the subject, who explained that she is barely able to use any standard email web portal or desktop client successfully without significant help and practice.

When receiving an email, she first tried to use Alice's key, but after the interviewer repeated the actions taken by General Gage, she immediately switched to her own key. When verifying, she again tried to use Alice's key, but understood the mistake when it was explained. After seeing the length of the longer instruction sets, she proclaimed herself a bad subject and gave up entirely. She did note that she had more success with less help in our experiment than with the commercial Yahoo! mail web portal.

The other elderly subject was far more successful. Before we began, he explained that he does not use technology heavily, but is very good at following directions to perform tasks in general. He was also given the abbreviated narrative as an introduction, but without the disclaimer that all parts of the toolkit are linked together. His biggest mistake was that he did not fully read the task instructions and first tried to send an email to Bob instead of Alice. After this was pointed out, he performed all of the tasks correctly, although with some hesitation. When locking, the interviewer simply reminded him that he wanted a "padlock" (and pointed out that the intended recipient was Alice instead of Bob), after which he successfully locked the message. When signing, he first asked if it was Alice's "stamper" that he wanted, and then answered his own question correctly.

When receiving a letter, he confidently and correctly performed both tasks. When asked about the implications of a missing imprint on a letter, he responded "It's not from Alice," which was almost correct. He then thoroughly read the other two versions of the instructions, declaring that the long narrative was "too long," but that the standard instructions were "clear." He also pointed out that the standard instructions clarify the linking of all parts in the toolkit, which is absent from the shorter narrative.

A third subject, who also professed to have limited technological literacy, struggled but said she did about as well as she usually does with Thunderbird or the Gmail portal. She performed all of the steps correctly with a little bit of encouragement, but was stymied by the verification process. She failed to understand the concept of verifying the imprint with another object, because as far as she was concerned she would have everyone's imprint memorized. She did, however, recognize the nuance that if the imprint were wrong, it meant that Alice did not send it (because the owner of the imprint did), but that if it was simply missing, Alice may still have sent it. This is likely because she nearly forgot to sign the message she sent, so she realized that Alice might do the same.

Finally, on the technologically-literate end of the spectrum, a college student with some computer science background did very well. She had no formal RSA training, but her father had explained the concept of asymmetric key encryption to her when she was young. She read the formal instructions quite quickly. She then performed every task correctly. When sending an email, she almost tried to lock the box with her own lock, but corrected herself and instead used Alice's lock. She immediately understood the nuance of an incorrect or missing imprint (signature) and explained it fully. When told she lost her key, she immediately knew to make a new toolkit and share her public key with all of her friends (she said this in the RSA language). When she admitted to having some previous RSA exposure from her father many years ago, she pointed out that our metaphors were very helpful in understanding how RSA works. She thought the short story was an effective and concise way of relaying the necessary information. She did, however, believe that the long narrative was too much length for too little gain, while the formal instructions at least provided every aspect in complete detail.

5.4 Discussion

These results shed a positive light on the use of our alternative metaphors. In some cases, the subjects used them directly in the user-interface. In others, they used the new metaphors to understand the concepts of the tasks they were trying to accomplish and translated the RSA language as such, mentally converting "private key" to "key" or "seal" appropriately. The abbreviated narrative was preferred, primarily due to its brevity, while the formal instructions were appreciated for being clear and complete.

Some subjects did point out that the user-interface was somewhat clunky, but the metaphors helped them understand how to use it properly, a criticism we describe in more detail in Section 6. This condition suggests that our metaphors were largely able to compensate for a weak user-interface, demonstrating that more effective metaphors can share the burden that most researchers of security usability seem to have placed on carefully crafted user-interfaces alone.

These interviews validate our hypothesis that users did not necessarily need to understand the underlying mathematical properties of public-key encryption to be able to

reason about the security properties of secure email. By framing our metaphors at the level of actions, we were successfully able to preserve a high-level picture of the process of using secure email while giving non-technical users the tools necessary to understand behaviors outside the norm, including various kinds of attacks.

6 Future Work

We feel that our investigation has barely scratched the surface of the possibilities for user-oriented metaphor design. As we discussed in our related work (Section 2), very little research has gone into strategies for developing better metaphors and methods of presenting them to users. As such, there are a variety of fruitful directions for expanding on our work.

Other Modes of Presentation We have examined only three of many possible ways of presenting security metaphors to non-technical users. Although our tests have already yielded interesting insights, the design space for metaphor introductions is vast. We could certainly delve deeper into the variations of our narrative approach, experimenting with other characters, scenarios and styles to hone a document that gives users a brief but readable and thorough introduction to our chosen terminology.

Alternatively, we have considered many other ways of explaining metaphors to users. Although we were limited by time and artistic ability in this experiment, we see substantial promise in using a comic strip format. This medium is inherently more inviting than a large block of text and can use pictures to clarify potentially imprecise language (especially prevalent with the many meanings of the word "seal"). In addition, a comic strip is a natural tool for tying the visual language of a user-interface to the conceptual ideas of metaphors.

We do acknowledge, however, that not all users are the same. Our user-interface testing drove home the notion that multiple kinds of learners encountered different pain-points. This topic borders on educational theory and psychology more than computer science, but seems like a critical area of research for those interested in security usability. We are certainly not experts in these fields, but we would be interested in exploring how subjects might make use of being provided with multiple forms of documentation or the ability to choose one of many introductions. Users might consult a comic strip for a quick introduction but reference thorough documentation for more specific questions, or could select sources better suited to individual learning styles. We are curious whether such choices would be an asset to usability or an unnecessary hoop for users to jump through.

Deeper User Testing Our Mechanical Turk-based quiz testing represents a first step that provides a rough picture of usability, but certainly has room for further study. It is difficult to verify the identities of participants or the quality of responses over this medium; we would ideally prefer to engage in large-scale in-person testing to make these features more readily apparent. This format also permits us to interview subjects and gain a much deeper picture of user-understanding than a multiple-choice quiz allows. More interestingly, this methodology enables us to engage with subjects more than once, testing recall and the ability to retain information about the metaphors on a long-term basis. Finally, we could more readily seek out demographics that are otherwise unlikely to be available on Mechanical Turk, like the elderly subjects we interviewed in our user-interface testing.

Fully-Featured User-Interface The user-interface with which we performed our lab testing was a mock-up of a possible email client that might include encryption features. This interface was not fully-featured in its own right, and, due to time constraints, only printed a log of user actions rather than performing RSA operations. Our lab testing therefore has expansive room to improve in both authenticity and real-world applicability. We would like to adapt our metaphors as a plug-in to a traditional email client to facilitate full-scale user testing. Since our study is focused on conceptual rather than visual motifs, it should be quite simple to exchange PGP metaphors for our own without drastically altering the interface.

Expanded Lab Testing We did not have the resources to perform the sort of large-scale replication of Whitten and Tygar's original study as done in *Johnny 2* [2], but this method of investigation would be of enormous value for further study of our metaphors and modes of presentation. With a fully-featured user-interface, we could adapt *Johnny 2*'s methodology to evaluate many combinations of metaphors and documentation with attention to both statistical and qualitative factors. This would also give us a direct basis for comparison against the results of *Why Johnny Can't Encrypt* and *Johnny 2*.

Other Applications Although PGP provides a useful toy case in which to experiment with metaphors and documentation, it has lost relevance in the decade and a half since *Why Johnny Can't Encrypt* was first published. More modern security and privacy-enhancing systems like Tor, OTR, or HTTPS are far more relevant media for exploring these techniques, even if they present many of the same conceptual challenges as PGP. We believe that the lessons of this paper (if not the metaphors themselves) are general enough to apply to other systems, but separate investigations are necessary to validate this hypothesis.

7 Conclusions

We developed alternatives to asymmetric encryption metaphors (public and private keys, etc.) that work on

the level of PGP actions rather than cryptographic primitives. We hypothesized that these metaphors would be easier for users to reason about and understand, facilitating better comprehension of the security properties and proper usage of secure email. To accompany this new lexicon, we experimented with forms of documentation that leveraged the real-world analogs of our metaphors - a task impossible with public and private keys.

In multiple-choice quiz-based testing, we found that, although our metaphors did not dramatically outperform those from standard PGP, we were able to achieve an equivalent level of understanding with far less rigorous documentation. Users were able to intuit many of the the security properties of secure email via a shorter fictional narrative describing how a historical figure used the physical analogs of the metaphors. Since these new forms of documentation are nearly impossible to compose using traditional PGP terminology, we feel this is a key use case for security metaphors with real-world analogs. Shorter documentation, we surmise, is more likely to be read in its entirety, increasing the likelihood that users correctly utilize security software and fully understand the implications of their actions.

Further lab testing confirmed our conclusions: using metaphors with real-world analogs and supplying concise documentation dramatically enhanced the overall user-experience. Results from both rounds of testing demonstrate that, while our metaphors could certainly be improved, our strategies are a sound and effective way to make intimidating security systems comprehensible and welcoming to non-technical users. Since many modern-day privacy and security-preserving programs require knowledge of the underlying technology for proper usage, such user-friendliness is essential. By crafting metaphors that resonate with end-users rather than recycling terms coined by security researchers, we have opened up a world of new ways to explain security to non-technical users.

A Guides for Quiz-Based Testing

A.1 Traditional Documentation

So you want to use secure email? Great! But what does it mean to send email securely? With secure email, we want to ensure two things: (1) Nobody can read a message except you and the recipient and (2) The message was actually sent by the person who claims to be the sender.

Getting started: To use secure email, you need to create a *toolkit* with everything you'll need. A toolkit contains four items: a bag of *locks*, a *key* that opens those locks, a *seal*, and an *imprint* of that seal. We'll explain what these items are and how to use them later on. The important thing to remember for now is that all of the items in the toolkit go together - if something breaks or gets lost, we need to get an entirely new toolkit. If you make a toolkit, everything in it is unique to that toolkit: no one else has the same locks, key, or seal. Thankfully, toolkits are free to create and can be generated out of thin air as necessary.

Sending Emails: We can think of an unsecured email as a normal letter. Anyone can open the envelope and read the contents, or, worse, send a letter pretending to be you. To stop this from happening, we place all of our letters in indestructible metal boxes. Next, we'll explain how to make sure that nobody can open letters addressed to someone except the recipient.

Sending Secure Emails: To send a secure email, you must obtain one of the recipient's locks and lock the box containing the message shut. In our world, these locks can't be picked or cut, so once you lock the box the message is protected. Once you do this, only a person possessing the corresponding key can open the box and read the message.

Opening Secure Emails: Once you receive a message, you simply use your key to open the lock on the box (remember, the sender put one of your locks on the box). You can now read the message as normal.

Proving your identity: Couldn't someone claim to be me when they send a message? Each toolkit also contains a seal - think an old-fashioned wax-pressing seal from the 1800s. To prove your identity, you imprint your seal on the box that contains a message. In our world, seals are unforgeable, meaning that a seal is definitive proof of your identity.

Verifying someone's identity: Someone can check that you sent a message by checking your seal imprint against the seal on the box. If they match, the message had to come from you.

Don't share your key or seal: As you have probably noticed, you can use this toolkit to communicate securely only so long as you don't share your key or seal. If someone else has your key, they can open emails that are addressed to you. If they have your seal, they can pretend to be you. If you share your key or seal, you should immediately let everyone know that the corresponding lock and imprint are no longer valid; afterward, you should create a new toolkit.

Don't lose your toolkit: The things that you use in your toolkit (i.e., the key and seal) are attached to the toolkit and if lost, are all lost together. If you lose your key, you won't be able to read any mail that was addressed to you using locks for that key. If you lose your seal, you'll be unable to prove your identity. If you lose your imprint, nobody will be able to check your identity. If you lose your locks, nobody will be able to send you secure email. If you lose any of these items, you'll need to create a new toolkit and let everyone know to stop using your old lock and imprint.

Share your lock and imprint as widely as possible: Nobody can send you emails or check your identity without them, so you should try to share this information with as many people as possible.

The mathematical reality: Do remember that these explanations are just metaphors for the underlying mathematics. In reality, your toolkit is two numbers: a key/seal (a "private key" in cryptography terms) and a lock/seal-imprint (a "public key").

A.2 Narrative Documentation

In the 1760s, British King George III ruled over a vast empire stretching across the entire world. As such, he needed a secure way to communicate royal orders to his colonial viceroys - one in which only a specific viceroy could read a letter and, further, in which that viceroy could always be sure it came from the King.

After locking himself away in his study for months, the chief royal scientist came up with an ingenious plan. Every viceroy would have his own special *key*; King George would have all of the corresponding *locks*. To send a letter, King George would place it in an impregnable *metal box* and lock it using the lock corresponding to his intended recipient. Since only this recipient had the matching key, nobody else would be able to open the box.

But couldn't anyone send a letter pretending to be King George? This was the royal scientist's act of genius: King George would have an unforgeable royal *seal*, whose *imprint* would be known to viceroys the world around. Any letter with the royal stamp had to come from King George.

The King was hesitant at first, but the system worked so well that he asked why his viceroys could not do the same when responding to his letters. The royal scientist obliged, making *toolkits* consisting of a key, a seal, locks, and imprints for every royal official. Since the scientist had to make these toolkits in batches for efficiency, anyone who lost a key or seal had to replace his entire toolkit.

Although it is left out of many textbooks, King George III's global system of secure communication was an essential element of the British Empire's success. In fact, all secure email systems today are based upon precisely the same principles that connected King George to his viceroys more than two centuries ago.

A.3 Abbreviated Narrative

King George III set aside his quill, having completed secret orders to put down the rebellion. It was imperative that they remain secure, visible only to Generals Gage and Howe. The King opened a cabinet in the wall behind him, revealing hundreds of *locks* each labelled with the name of a British General. Selecting one with "Gage" engraved on the side, the King placed his orders for General Gage in an impregnable *metal box* and secured it shut with the lock. Since only General Gage possessed the corresponding *key*, the King knew that the orders were secure from prying eyes. After doing the same for General Howe, King George marked the boxes with his royal *seal*, whose *imprint* was known throughout the world. Anyone who received the message could now be sure it came from the King.

Several weeks later, two metal boxes arrived on the King's desk, one bearing the unforgeable imprint of General Gage's seal and the other of General Howe's. Both boxes were bound shut with locks engraved with "His Majesty King George III" on their sides. The King unlocked the boxes with his personal key, revealing two identical documents: "It is done."

A.4 Narrative for Lab Testing

King George III wants to send a secure letter to General Gage. First, he writes the letter Then, he puts it in an indestructible metal box. He locks the box with a padlock to which only General Gage has the key (not even the King has it!). Next, he stamps the box with his (King George's) seal, leaving his imprint on it. Nobody else has the royal seal, and the imprint is unforgeable.

General Gage receives it. Opening it is easy. He simply uses his key to open the lock. Then, he checks the imprint against a copy of the King's imprint he has on file. It matches, so he knows it came from the King. He knows these are official orders, so he reads the letter.

Notes: This communication works both ways. General Gage can also send reports to the King reciprocally. If any piece of a toolkit is lost (key or seal), then the entire toolkit (key, seal, locks, and imprint records) needs to be remade. Once lost, keys and seals cannot be recovered.

B Quiz

B.1 Questions and Information Sought

1. *What does it mean for an email to be secure?*
   Free-response. Security depends on both confidentiality and integrity of the message.
2. *Which of the following does secure email protect you from?*
   Secure messaging protects against eavesdroppers and forgers.
   - Someone looking over your shoulder as you send and receive emails
   - Someone impersonating your friend in emails to you
   - The NSA trying to read your emails in transit
   - Your friend forwarding to others the contents of messages from you
   - Someone stealing your friend's key and reading emails you sent to that friend before the theft

3. *Who do you have to rely on to ensure your email stays secure?*
   The sender requires cooperation from the receiver to keep messages confidential.
   - Yourself
   - Your Internet Service Provider (e.g., Comcast, Verizon)
   - Your Wifi Network
   - People who send you emails
   - People who receive emails that you send
4. *If you, Alice, want to send a secure letter to Bob, what objects do you need?*
   Who uses whose objects when sending and receiving a secure message.
   - A steel box
   - One of your (Alice's) locks
   - One of Bob's locks
   - Your key
   - Bob's key
   - Your seal
   - Bob's seal
5. *When sending a secure letter to Bob, which object(s) will guarantee only Bob can read the letter?*
   Which cryptographic object guarantees confidentiality? How do you use those objects?
   - A steel box
   - One of your (Alice's) locks
   - One of Bob's locks
   - Your key
   - Bob's key
   - Your seal
   - Bob's seal
6. *You, Alice, receive a letter in a box from Bob. How do you know or check that it actually came from him?*
   A valid seal shows who the sender must be.
   - I don't know
   - It will have one of my locks on it and only Bob has my locks
   - It will have one of Bob's locks on it and only Bob has his locks
   - It will have an imprint of my seal on it and only Bob has my seal
   - It will have an imprint of Bob's seal on it and only Bob has his seal
   - Other
7. *Suppose you receive a box from Bob that doesn't have the thing(s) required. What can or can't you assume about the identity of the sender of the letter?*
   Absence of the seal is not a guarantee of forgery; it does warrant caution.
   - The sender is still Bob
   - The sender could be Bob, but we don't know
   - The sender is definitely not Bob
8. *Suppose you lose your toolkit. If you want to keep sending and receiving secure messages, what do you need to do?*
   Losing your toolkit requires replacement and informing future senders and recipients of the change.
   - Make a new toolkit
   - Get new locks from all your friends
   - Send your new locks to all your friends
   - Get new seal imprints from all your friends
   - Send your new seal imprint to all your friends
   - Other
9. *If you lose your toolkit, can you still read your old messages?*
   A lost toolkit means you cannot read old encrypted messages.
   - Yes
   - Maybe
   - No
10. *Suppose you lose your toolkit. A friend of yours hasn't heard the news and sends you a secured message. Can you be sure that the message came from her?*
    Even when you lose your toolkit, you can validate messages.
    - Yes
    - Maybe
    - No
11. *Suppose you lose your toolkit. A friend of yours hasn't heard the news and sends you a secure message. Can you still read the message?*
    A lost toolkit means you cannot read new messages encrypted to your old key.
    - Yes
    - Maybe
    - No
12. *Your friend tells you he lost his toolkit. What do you need to do before you can send him a message securely?*
    Free-response. A friend's lost toolkit means you must wait to receive his or her new credentials before messages can be sent between you two.
13. *If someone else acquires your key/private key, what will they be able to do?*
    If someone else gets a copy of your key/private key, they can read messages sent to you in the past or in the future. With the private key, they can also impersonate you.
    - Impersonate you in messages to others
    - Impersonate others in emails to you
    - Read old emails sent to you
    - Read old emails sent to others by you
    - Read future emails sent to you
    - Read future emails sent to others by you
14. *If someone else acquires your lock/public key, what will they be able to do?*
    If someone else gets a copy of your lock/public key,

they can't do anything malicious they couldn't do otherwise.
    - Impersonate you in messages to others
    - Impersonate others in emails to you
    - Read old emails sent to you
    - Read old emails sent to others by you
    - Read future emails sent to you
    - Read future emails sent to others by you
15. (Metaphor only) *If someone else acquires your seal, what will they be able to do?*
    If someone gets a copy of your seal, they can impersonate you when sending messages.
    - Impersonate you in messages to others
    - Impersonate others in emails to you
    - Read old emails sent to you
    - Read old emails sent to others by you
    - Read future emails sent to you
    - Read future emails sent to others by you

B.2 Summary of Results

Question $n =$	PGP 40	Metaphor 40	Long 20	Short 1 20	Short 2 20
1	0.65	0.55	0.28	0.28	0.12
2	0.40	0.30	0.44	0.36	0.20
3	0.325	0.25	0.24	0.0	0.0
4	0.225	0.325	0.28	0.16	0.24
5	0.425	0.20	0.04	0.04	0.08
5.1	0.275	0.35	0.56	0.76	0.64
6	0.575	0.55	0.44	0.56	0.44
7	0.80	0.70	0.84	0.96	0.88
8	0.375	0.225	0.12	0.08	0.16
9	0.80	0.825	0.32	0.48	0.52
10	0.50	0.35	0.40	0.08	0.36
11	0.85	0.85	0.84	0.72	0.60
12	0.60	0.575	0.36	0.08	0.28
13	0.0	0.0	0.0	0.0	0.0
14	0.425	0.275	0.04	0.16	0.20
15	0.0	0.725	0.80	0.68	0.68

Table 2: Summary of the success rates of subjects given the description they were provided. Rates are broken down by question. Unless noted otherwise, success rates are for a group of size 40 for the PGP and Metaphor columns, and 20 for the rest.

References

[1] A. Fry, S. Chiasson, and A. Somayaji. Not sealed but delivered: The (un) usability of s/mime today. In *Annual Symposium on Information Assurance and Secure Knowledge Management (ASIA'12), Albany, NY*, 2012.

[2] S. L. Garfinkel and R. C. Miller. Johnny 2: a user test of key continuity management with s/mime and outlook express. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 13–24. ACM, 2005.

[3] S. Sheng, L. Broderick, C. A. Koranda, and J. J. Hyland. Why johnny still can't encrypt: evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security*, 2006.

[4] S. Srikwan and M. Jakobsson. Using cartoons to teach internet security. *Cryptologia*, 32(2):137–154, 2008.

[5] A. Whitten. *Making security usable*. PhD thesis, Carnegie Mellon University, 2004.

[6] A. Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium*, volume 99, page 16. McGraw-Hill, 1999.