

An Inconvenient Trust: User Attitudes Toward Security and Usability Tradeoffs for Key-Directory Encryption Systems

Wei Bai, Doowon Kim, Moses Namara, Yichen Qian,
Michelle L. Mazurek, and Patrick Gage Kelley*
University of Maryland University of New Mexico*
wbai@umd.edu, doowon@cs.umd.edu, namaramss@gmail.com,
yqian1@umd.edu, mmazurek@cs.umd.edu, and pgk@unm.edu

ABSTRACT

Many critical communications now take place digitally, but recent revelations demonstrate that these communications can often be intercepted. To achieve true message privacy, users need end-to-end message encryption, in which the communications service is not able to decrypt the content. Historically, end-to-end encryption has proven extremely difficult for people to use correctly, but recently tools like Apple’s iMessage and Google’s End-to-End have made it more broadly accessible by using key-directory services. These tools (and others like them) sacrifice some security properties for convenience, which alarms some security experts, but little is known about how average users evaluate these tradeoffs. In a 40-person interview study, we asked participants to complete encryption tasks using both a traditional key-exchange model and a key-directory-based registration model. We then described the security properties of each and asked participants for their opinions. We found that participants understood the two models well and made coherent assessments about when different tradeoffs might be appropriate. Overall, our participants found the security of the registration model to be “good enough” for many everyday purposes.

ACM Classification Keywords

K.4.1 Public Policy Issues: Privacy; H.1.2 User/Machine Systems: Human factors; H.4.3 Communications Applications: Electronic Mail

Author Keywords

Email; Encryption; Human Factors; Key Exchange; Privacy; Usability

INTRODUCTION

As important communications become primarily digital, privacy becomes an increasingly critical concern. Users of communication services (e.g., email and chat) risk breaches of confidentiality due to attacks on the service from outsiders or rogue employees, or even government subpoenas. The only way to truly assure confidentiality is to use encryption so that the communication service has no access to the content. Despite considerable evidence of and front-page reporting about content breaches [3, 4, 8, 17, 20], encryption has generally not been widely adopted for person-to-person communications such as email and chat [16].

Researchers have given considerable thought to the reasons for this lack of adoption. More than 15 years of research have identified major usability problems with encryption tools,

ranging from poorly designed user interfaces to the fundamental challenges of safe and scalable key distribution [35, 28, 13, 26].

Recently, however, progress toward better usability and thus wider adoption has been made. Apple applied seamless end-to-end encryption to its iMessage and FaceTime services [21, 2]. By centrally distributing public keys, Apple ensured the encryption is transparent to users, bringing end-to-end encryption to millions of iPhone, iPad, and Mac users. This design, however, leaves open the possibility that Apple itself could carry out a man-in-the-middle attack to break its users’ privacy, for example at the request of law enforcement authorities [7, 34]. Google and Yahoo! are currently implementing similar approaches, with an added monitoring protocol that allows users and third parties to audit the key directory for consistency and transparency [30, 24, 15]. Some privacy experts have suggested that given this potential man-in-the-middle attack, these services should not be recommended to end users. As just one example, one security researcher suggests that “iMessage remains perhaps the best usable covert communication channel available today if your adversary can’t compromise Apple. ... If one desires confidentiality, I think the only role for iMessage is instructing someone how to use Signal¹” [34].

In a sense, the issue comes down to whether the benefit from many more people adopting encrypted communications is outweighed by the reduced security inherent in the central key distribution model. To our knowledge, however, no one has asked average users for their opinions. To understand how non-expert users feel about this tradeoff, we undertook a 40-person lab study. We introduced participants to two encryption models: an *exchange* model in which participants manually exchange keys (analogous to traditional PGP) and a *registration* model in which participants sign up with a central service that distributes keys (analogous to iMessage). For each model, we asked them to complete several encrypted communication tasks; we also gave them a short, high-level explanation of each model’s security properties. (We varied the order of presentation to account for biases.) We then asked participants to comment on the security and usability of each model, as well as their overall opinion of the tradeoffs involved. The experiment was designed, insofar as possible, to avoid comparisons based on user-interface design and focus instead on the underlying properties of each model.

¹An encryption tool: <https://whispersystems.org/>

We found that participants understood the two models fairly well and expressed nuanced insights into the tradeoffs between them. As predicted, participants found the registration system considerably more convenient than the exchange system. More interestingly, while the exchange system was considered more secure overall, the difference was slight: both general trust that large email providers would not risk their reputations by cheating and reasonable concerns about participants' own ability to implement the exchange model correctly mitigated this difference. Separately, we asked some of our participants to evaluate the auditing model proposed in CONIKS [23], which is similar to that planned by Google and Yahoo!, and we found that it provides a small but meaningful additional degree of confidence in the system's privacy. Overall, our results suggest that users find the registration model sufficient for the majority of everyday communications they engage in. We therefore argue that in many cases, the marginal benefit of broadly adopting such a model outweighs the risks. Rather than spreading alarm about these risks, we recommend that policymakers and designers present tradeoffs clearly and encourage adoption of usable but imperfect security for many casual scenarios.

BACKGROUND AND RELATED WORK

We briefly discuss the history of public-key-encrypted email systems and encryption usability studies.

A brief history of encrypted email

Diffie and Hellman proposed public-key cryptography in 1976, suggesting that a public directory would allow anyone to send private messages to anyone else; in 1978, the RSA algorithm made the idea practical [9]. In 1991, John Zimmerman developed PGP, which supported sending public-key encrypted email. In the second version, to alleviate the key verification problem, he proposed a "web of confidence" (later known as web of trust) for establishing key authenticity [36]. In a web of trust, users can sign each others' keys to endorse their authenticity, and can choose to accept keys that come with signatures from "trusted introducers." Despite this, key verification has remained problematic for many years.

In 1999, RFC 2633 defined Secure/Multipurpose Internet Mail Extensions (S/MIME), which takes a centralized approach to key distribution: all users have public-key certificates signed by a certification authority (CA), which are distributed along with any signed emails sent by that user [25]. S/MIME allowed straightforward integration of encryption to email clients like Microsoft Outlook and Netscape Communicator and was adopted by some corporate organizations with the capability to manage keys hierarchically, but was not adopted broadly by consumers.

More recently, several researchers and companies have explored ways to split the difference between completely decentralized and completely centralized key management. Gutmann proposed applying *key continuity management*, in which keys are trusted on first use but key changes are detected, to email [18]. In Apple's iMessage, private keys are generated on users' devices and the corresponding public keys are uploaded to Apple's proprietary directory service. To

send a message to a user with multiple devices, the message is encrypted once for each device [21, 1]. In *certificate transparency*, publicly auditable append-only logs can be used to determine whether rogue certificates have been signed using a stolen CA key [22]. Ryan extended this approach for end-to-end email encryption [27]. CONIKS extends certificate transparency to allow users to efficiently monitor their own key entries and to support privacy in the key directory [23]. Google and Yahoo! are adopting a variation of certificate transparency for their end-to-end encryption extension [15, 24]. Each of these approaches trades off a different amount of security for convenience.

Other researchers have considered alternatives to standard public-key encryption that are designed to be more usable. Fahl et al. proposed Confidentiality as a Service (CaaS) [10], which operates on a registration model mostly transparent to users. This approach uses symmetric cryptography and splits trust between the communications provider and the CaaS provider. Neither individually can read private messages, but if the two collude they can.

The usability of encrypted email

In 1999, Whitten and Tygar published the now-seminal *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0* [35]. This paper evaluated the interface for PGP 5.0 and found that most users (two-thirds) were unable to successfully sign and encrypt an email in the 90 minute session. This led to a series of follow-on papers: evaluating PGP 9 (key certification is still a problem) [28], S/MIME and Outlook integration (KCM seems promising) [13], Facebook encryption (using CaaS) [11], and several others (e.g., [26, 31]). These studies largely ask users to do tasks they are unfamiliar with and focus on success rates (key pairs generation and collection, sending and decrypting messages, etc.). They provide valuable insight into how effectively novices can learn a particular system, how specific user interface design choices impact users, and where the difficulties lie. However, users are rarely presented with multiple potential encryption infrastructure models, or asked to consider the underlying security and convenience tradeoffs of the systems they are evaluating.

Tong et al. re-evaluated the test of *Johnny* with a different set of terms and documentation, including using a lock-and-key metaphor for public and private keys [32]. In preliminary results, they found that the metaphors aided understanding. We adopt the lock metaphor in our study, as detailed below.

Finally, there are social and cultural norms that also lead to aversion to encryption. Often users believe that they have no reason to encrypt their email because they have "nothing to hide," or because they cannot imagine anyone being interested in the messages they are sending [29]. In an interview study at an unnamed non-violent, direct-action organization (which one might expect to be more interested and aware of the benefits of encryption) Gaw et al. found that employees believed "routine use of encryption [was] paranoid [behavior]" [14]. In this work, we do not directly address social norms regarding encryption, but several participants did discuss paranoia and suggested using different systems to accommodate different levels of privacy concern.

METHODOLOGY

We used a within-subjects lab study to examine participants’ concerns and preferences regarding the usability and security of end-to-end email encryption. Each participant was introduced to two general models for key management, *exchange* and *registration*. For both models, we described a public key as a *public lock*. This approach, inspired by Tong et al., avoids overloading the term “key” and was used to provide a more intuitive understanding of how public-key pairs operate [32].

In the exchange model, similar to traditional PGP, participants generate a key pair and then distribute the public locks to people they want to communicate with. We offered participants several methods for exchanging locks: the same email account they would use for encrypted communication, a secondary email account, posting the public lock on Facebook or sending via Facebook Messages, or using a simulated “key server” to upload their lock to a public directory. (These options were presented to each participant in a random order.) Simulated correspondents (played during the study by a researcher) sent back their own public locks via the same mechanism the participant chose, or via the mechanism the participant requested.

In the registration model, participants again generate a key pair. In this case, they “register” their public lock with a simulated key directory service; correspondents’ locks were pre-installed to simulate automatically retrieving them from the directory. Participants were thus able to send and receive encrypted email from all simulated correspondents immediately upon creating and registering their own keys. In iMessage, the key generation step itself is completely transparent to users, who may never realize a key was created; we chose instead to make key generation explicit to help users understand the process.

Within each model, participants were asked to complete a series of simulated tasks, such as exchanging encrypted emails in a role-playing scenario (see details below); they were also introduced to a brief, non-technical review of the security properties of each model. Participants were asked to give their opinions about each model immediately after completing the tasks and security learning for that model. We also conducted an exit interview regarding the overall usability and security of each model, whether participants would use it themselves or recommend it to others, and in what circumstances it might or might not be appropriate.

We chose a within-subjects study because we were primarily interested in how participants would understand and value the tradeoffs among the options. As shown in Table 1, we varied the order of activities to account for ordering effects. Participants were assigned round-robin to one of these four possible orders of activities.

Encryption tasks

The set of encryption-related tasks for each model is shown in Table 2. In both models, participants were asked to generate a key pair locally. In the exchange model, participants then exchanged public locks with simulated friend Alice, including both sending Alice their lock and importing

First activity	Second	Third	Fourth
ET (Exchange, Tasks)	ES	RT	RS
ES (Exchange, Security learning)	ET	RS	RT
RT (Registration, Tasks)	RS	ET	ES
RS (Registration, Security learning)	RT	ES	ET

Table 1: The order of activities varied across participants. Each participant worked with either the Exchange (E) or the Registration (R) model first. Within each model, participants either completed the encryption Tasks (T) first or learned about Security properties (S) first. Throughout the paper, participants are labeled by first activity; e.g., participant RT3 completed encryption tasks for the registration model first.

the lock received in return. In the registration model, participants registered with a simulated central service and had their public lock automatically “uploaded” and others’ locks automatically “imported.” After the locks were exchanged or the participant registered, participants composed and sent an encrypted email to Alice. A researcher, posing as Alice, sent an encrypted response. As a slightly more complex task, participants were asked to send an encrypted email to a group of two recipients. This task was designed to get participants to consider how the two models scale. Finally, we asked participants to consider how they would handle several other situations, including communicating with larger groups of people and various possible errors related to losing or publicizing one’s own private key or losing other users’ public locks. The possible errors were specific to each model and are shown in Table 2. In the interest of simplicity, we did not include any email signing (or signature verification) tasks.

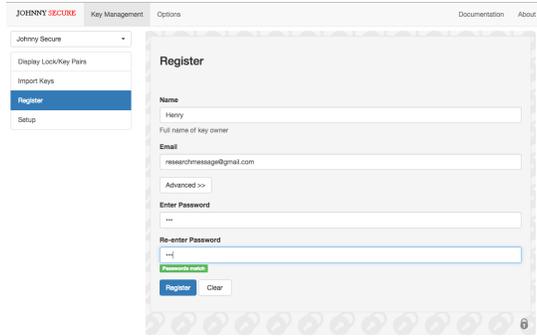
Encryption tasks were completed using a Gmail account created especially for the study and a Chrome browser extension based on Mailvelope.² We modified Mailvelope to remove its branding, change the labels to match our lock/key metaphor, and reduce the interface to include only those features relevant to the study tasks. Figure 1, right shows a screenshot of sending encrypted email with our extension. As in Mailvelope, users of our extension compose an email and then use an “Encrypt” button to select recipients. Upon receiving encrypted email, users are prompted to enter their password to decrypt it (with the option to save the password and avoid future prompting).

We created two versions of our extension, one for exchange and one for registration, taking care to make them as similar as possible. The only two visible differences were (1) changing the “Generate lock/key pair” menu item and subsequent screen (exchange model, Figure 1, left) to read “Register” (registration model) and (2) a lock import screen (Figure 1, center) that was only relevant in the exchange model.

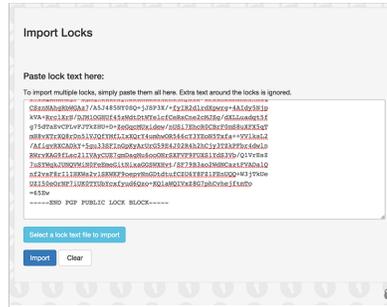
We also provided participants with detailed instructions to help them use the Chrome extension. By simplifying the interface, keeping it consistent, and providing detailed instructions, we hoped participants’ reactions would better reflect

²<https://www.mailvelope.com/>

1. Register/Generate



2. Import



3. Compose

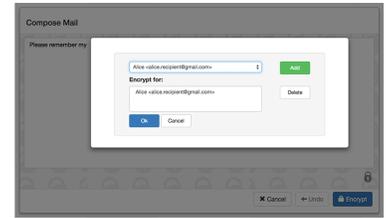


Figure 1: To use our extension, participants first generated (or registered) a key pair. Participants using the exchange model then needed to import recipients' locks. Finally, when composing encrypted emails, they clicked the Encrypt button (shown in the lower right of Step 3) to bring up a modal dialog to select recipients.

Task #	Exchange Model	Registration Model
1	Generate public lock/private key pair	Register public lock/private key pair
2	Exchange public locks with Alice	N/A
3	Send encrypted email to Alice	Send encrypted email to Alice
4	Decrypt received email from Alice	Decrypt received email from Alice
5	Exchange public locks with Bob and Carl	N/A
6	Send encrypted email to Bob and Carl	Send encrypted email to Bob and Carl
7	Decrypt received email from Bob and Carl	Decrypt received email from Bob and Carl
8	Imagine sending encrypted email to 10 people.	Imagine sending encrypted email to 10 people.
9	Consider misconfigurations: a. Lose Alice's public lock b. Lose own private key c. Publicize own private key	Consider misconfigurations: N/A b. Lose own private key c. Publicize own private key

Table 2: The encryption-related tasks completed by participants. The tasks differed slightly in the two models.

the inherent properties of each model rather than idiosyncrasies of a particular interface.

Description of security properties

We provided participants with short, non-technical descriptions of possible attacks on each model.

Exchange model

For the exchange model, we described a man-in-the-middle attack in which the attacker could intercept or replace keys during the exchange process: "For example, when you try to get the public lock from Dave, the attacker secretly switches the public lock to his own. You think you have Dave's public lock, but in fact you have the attacker's. ... As a result, the attacker can read your email. The attacker will then use Dave's public lock and send the encrypted email to Dave, so that neither you nor Dave realize the email has been read." We also showed participants the illustration in Figure 2.

We decided not to include an option for key signing in our exchange model both because we thought it would add unnecessary complexity to our explanations and because it does not change the underlying requirement to trust some keys that are manually exchanged.

Registration model

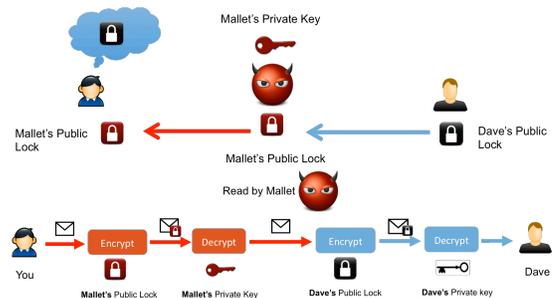


Figure 2: Possible attacks on the exchange model

For the registration model, we primarily described a man-in-the-middle attack enabled by the key directory service: "When you try to send encrypted emails to Dave, you think the database will return Dave's public lock to you. But in fact, it returns the attacker's lock, so the attacker can read your email. Therefore, you need to trust the email provider in this system." We showed participants the illustration in Figure 3.

In addition, we described two variations on the basic key directory approach: the Confidentiality as a Service (CaaS) variation [10, 11], and an auditing model similar to the

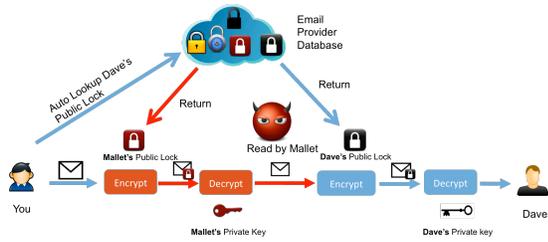


Figure 3: Possible attacks on the registration model

one proposed by Google and CONIKS [15, 23]. Because these approaches are not currently in wide use the way the iMessage-analogous system is, they were treated as secondary options. The auditing model was added (to the end of the interview, to maintain consistency with earlier interviews) during recruiting, and was therefore presented only to 12 participants.

The security of the CaaS variation was described as follows: “There is a third-party service (not the email provider) as an intermediary. In this version, neither the third-party service nor your email provider can read your email themselves. However, if your email provider and the third-party service collaborate, they can both read your email. Therefore, you need to trust that the two services are not collaborating.”

We described the auditing variation as follows: “The email provider stores all users’ public locks, just like [the primary registration model]. But there are other parties (auditors) who audit the email provider, to ensure it is giving out correct public locks. These auditors may include other email providers, public interest groups, and software on your devices. If the email provider gives you a public lock that doesn’t belong to the recipient, or gives someone else the wrong public lock for you, these auditors will notify you. You (or someone else) may use the wrong lock temporarily (for an hour or a day) before you are notified. In this model, you don’t need to trust your email provider, but you need to trust the auditors and/or the software on your device. Because there are several auditors, even if one auditor does not alert you another one probably will.”

Participant feedback

Participants were asked questions after completing tasks for each model and at the end of the process. After completing tasks and learning about security for each model, participants were asked for their agreement (on a five-point Likert scale) with the following statements:

- The task was difficult (for each task).
- The task was cumbersome (for each task).
- The system effectively protected my privacy.

The first two questions were repeated for each task in Table 2. Before answering, participants were reminded that difficult tasks would require intellectual effort or skill, while cumbersome tasks would be tedious or time-consuming. After each Likert question, we asked participants to briefly explain their answer choice (free response).

After completing all tasks and learning about all security models, participants were asked several summative questions, including:

- Willingness to use each system, on a five-point Likert scale, and why.
- Willingness to recommend each system, on a five-point Likert scale, and why.
- What the participant liked and disliked about each system.

Recruitment

We recruited participants 18 or older who were familiar with Gmail and Chrome and who send and receive email at least 3 times per week. We placed flyers around our university campus and the surrounding area, advertised via email listservs for the university, and advertised on web platforms like Craigslist. All interviews were conducted in person at our university campus; interviews were video recorded with the explicit consent of participants. Participants were paid \$20 for a one-hour study and were reimbursed for parking if necessary. Our study protocol was approved by the university’s Institutional Review Board.

Data analysis

We used statistical analysis to investigate participants’ responses to the exchange and registration models. To account for our within-subjects design, we used the standard technique of including random effects to group responses from each participant together. We used a cumulative-link (logit) mixed regression model (notated CLMM), which fits ordinal dependent variables like the Likert scores we analyzed [19]. We included three covariates: whether the participant performed tasks or learned about security first, whether the model she was evaluating was seen first or second, and the model type itself (exchange or registration). This approach allows us to disentangle the ordering effects from the main effects we are interested in. For each encryption model, we tested statistical models with and without the obvious potential interaction of model type and model order, selecting the model with the lower Akaike information criterion (AIC) [5].

Qualitative data was independently coded by two researchers using textual microanalysis. After several iterative rounds of developing a coding scheme, the researchers each independently coded the full set of participant responses, with multiple codes allowed per response. The researchers originally agreed on more than 94% of the codes, then discussed the instances of disagreement until consensus was reached. Some frequencies of these finalized qualitative codes are reported to provide context.

Limitations

Our methodology has several limitations. Our lab study participants had only limited exposure to the different encryption models, and their opinions might change after working with the models for a longer period. Participants also only imagined their responses to misconfigurations, rather than actually handling them. Nonetheless, we argue that first impressions like the ones we collected influence whether people will try any tool for long enough to develop more-informed opinions.

It is well known that study participants may rate tools they examine more favorably (acquiescence bias) [33], which may explain the high rate of participants reporting they wanted to use or recommend each model. Because we are primarily interested in comparing results between models, we believe this has limited impact on our overall results; however, the absolute ratings should be interpreted as a ceiling at best.

In order to provide participants with any understanding of the security properties of each model, we had to prime them with descriptions of possible attacks. While this priming was unavoidable, we endeavored to keep the security descriptions as neutral as possible so that priming would affect both models approximately equally.

To avoid overwhelming participants, we evaluated a limited subset of possible encryption models and possible tasks; in particular, we left out key signing as well as any email signing or signature verification tasks. We did this because we believe signing to be the most difficult aspect of cryptography for non-experts to understand (see e.g., [35]), but including it might have provided a broader spectrum of user opinions. Our registration model, unlike for example iMessage, was not completely invisible to participants. We believe it was necessary to give participants something to do other than just sending a normal email, in order to help them think through the tradeoffs involved. While presumably using a fully transparent variation would only have increased the convenience gap between the two models, prior work indicates that taking any steps at all increases feelings of security [26]. This may have contributed to the small observed security gap between the two models, but we argue that a version with intervention required would lead to underestimations of security. Because we added the auditing model late, we were not able to get as much feedback about it or to compare it quantitatively to the other models we examined, but the qualitative data we collected does provide interesting insights. Future work can examine all these alternatives in more detail.

As with many lab studies, our participants do not perfectly reflect the general population, which may limit the generalizability of our results.

PARTICIPANTS

A total of 66 people completed our pre-screening survey. We interviewed the first 41 who qualified and scheduled appointments. One participant was excluded for failing to understand or respond coherently to any directions or questions.

Demographics for the 40 participants we consider are shown in Table 3. Among them, 62.5% were male and 80% are between the ages of 18-34, which is somewhat maller and younger than the general American population. Almost 90% of participants reported “primarily” growing up in the United States, South Asia, or East Asia. Half of participants reported jobs or majors in computing, math, or engineering.

Despite this high rate of technical participants, most had little experience with computer security. We measured security expertise using a slightly adapted version of the scale developed by Camp et al [6]. Higher scores indicate security expertise;

ID	Gender	Age	Occupation	Security Expertise	Where grew up
ET1	F	25-34	Professional	0	United States
ET2	F	45-54	Education	0.5	United States
ET3	M	21-24	Education	1.5	United States
ET4	M	25-34	Education	2	Middle East
ET5	M	21-24	Computers/math	1	South Asia
ET6	M	25-34	Engineering	2	East Asia
ET7	M	45-54	Life Sciences	2	United States
ET8*	M	18-21	Engineering	0.5	East Asia
ET9*	F	21-24	Computers/math	1	South Asia
ET10*	F	35-44	Computers/math	2	United States
ES1	M	35-44	Engineering	0	United States
ES2	M	21-24	Sales	0.5	United States
ES3	F	25-34	Health Care	0.5	United States
ES4	M	21-24	Computers/math	4	South Asia
ES5	M	21-24	Computers/math	1	East Asia
ES6	M	25-34	Computers/math	1.5	South Asia
ES7	F	21-24	Education	0.5	United States
ES8*	M	25-34	Engineering	0.5	East Asia
ES9*	F	21-24	Engineering	1	South Asia
ES10*	M	25-34	Engineering	1	United States
RT1	M	25-34	Computers/math	3	East Asia
RT2	F	25-34	Sales	0.5	United States
RT3	M	21-24	Engineering	2.5	South Asia
RT4	F	21-24	Engineering	1.5	United States
RT5	M	21-24	Business	2	East Asia
RT6	F	25-34	Professional	1.5	United States
RT7	F	25-34	Health Care	0	United States
RT8*	F	18-20	Sales	0.5	United States
RT9*	M	18-20	Education	0.5	United States
RT10*	M	25-34	Engineering	2	Middle East
RS1	M	21-24	Professional	1	East Asia
RS2	M	25-34	Life Sciences	1.5	Middle East
RS3	M	21-24	Computers/math	0	Africa
RS4	M	21-24	Computers/math	0.5	South Asia
RS5	M	25-34	Life Sciences	2	Middle East
RS6	M	25-34	Professional	0.5	United States
RS7	F	25-34	Health Care	0	United States
RS8*	F	45-54	Sales	0	United States
RS9*	F	25-34	Engineering	1.5	East Asia
RS10*	M	21-24	Engineering	1	United States

Table 3: Participant Demographics. The columns show: participant identifiers (coded by activity order), gender, age, occupation, security expertise, and place where the participant grew up. The * indicates participants who were exposed to the auditing model.

the maximum score is 5.5 and the minimum score is zero. Only two of our participants scored 3 or higher.

A Kruskal-Wallis omnibus test found no significant differences among our four conditions in age, gender, country of origin, or security expertise ($p > 0.05$).

RESULTS AND ANALYSIS

We present participants’ reactions to the convenience and security of each model, followed by a discussion of their overall preferences among the models.

Registration is less cumbersome

Unsurprisingly, our participants found the registration system considerably more convenient; interestingly, however, this was true for tediousness but not difficulty. Figure 4 and Tables

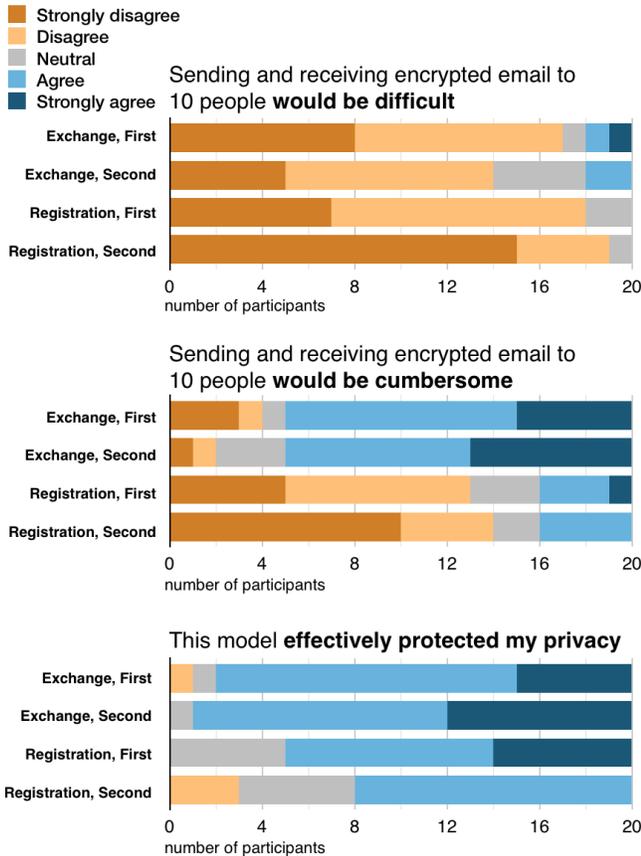


Figure 4: Participants’ ratings of difficulty and cumberbsomeness (Task 8) as well as whether participants thought the model protected their privacy. Labels indicate which model participants evaluated along with whether they saw that model first or second; e.g., “Exchange, First” indicates ratings for the exchange model among those who saw it first, which includes ET and ES participants.

4 and 5 show the results of the CLMM for difficult and cumberbsome, respectively, for Task 8: imagining sending email to a group of 10 people. In reading the CLMM tables, the exponent of the coefficient indicates how much more or less likely participants were to move up one step on the Likert scale of agreement.

For cumberbsomeness the exchange model was associated with a more than 26x increase in likelihood of indicating more agreement. For difficulty, in contrast, the only significant predictor was the order of exposure to the two models, with the model seen second slightly less likely to be perceived as difficult.

Participants’ comments generally supported this finding: that the exchange model was more cumberbsome but not necessarily more difficult. Within the exchange model, the most tedious task was manually exchanging locks (agreed or strongly agreed for 25 participants), and the most commonly mentioned reason was waiting for a correspondent’s public lock.

ES9 was concerned that the exchange model was “time-consuming, especially sending urgent emails. I have no choice but to wait for” the correspondent’s public lock. RS5 agreed, saying “There are so many steps to exchange locks.” While few participants considered any of the tasks very difficult, choosing a mechanism for exchanging locks was considered the most difficult step by a few participants, such as RS4, who mentioned having to “think about a safe way to exchange public locks,” and RS10, who was concerned about making an exchange error while multitasking.

Other concerns related to the general issue of convenience included scalability and misconfiguration. As RT9 said, “When I send to more people, I have to be very careful, especially when I choose to send them my public locks separately. I need to control every step.” A few participants were concerned about the difficulty of recovering from misconfiguration, and ET10 was particularly worried that others’ mistakes could cause additional hassle for her: “If other people lose their private keys and send me new public locks, I will be overwhelmed.”

The inconvenience of the exchange model could potentially be mitigated somewhat by posting the key publicly or semi-publicly (on a key server or Facebook profile), rather than sending it individually to different recipients. Few of our participants, however, chose this option: 26 used the primary email, 18 used the secondary email, nine used Facebook chat, four posted to the Facebook profile, and 10 used the key server. (Some participants chose multiple methods during different tasks.) ET7 uploaded his public lock to key server because he thought it was more secure than other choices we provided, but no participants mentioned the potential convenience of posting once to a public forum.

The perceived security gap is small

We found that participants understood and thought critically about the security properties we explained to them for each model. Surprisingly, they found the exchange model to be only marginally more secure than the registration model, for a variety of reasons.

Exchange: Manual effort may lead to vulnerability

Most participants believed the exchange model was most secure overall, with 37 agreeing or strongly agreeing that this model protected their privacy. Nonetheless, participants also expressed concern that managing key exchange themselves would create vulnerabilities. More than half of participants

Factor	Coef.	Exp(coef)	SE	p-value
tasks first	0.393	1.482	0.673	0.559
second model	-0.161	0.852	0.446	0.719
exchange	3.275	26.435	0.688	<0.001*

Table 4: Regression table for cumberbsomeness, Task 8. The non-interaction model was selected. Non-significant values are greyed out; significant values are indicated with an asterisk.

were concerned about the security of the medium used to exchange locks—ET4 worried that “the key server [could] be manipulated or compromised,” and RT7 suggested that an attacker could break into her Facebook account and post an incorrect public lock. Others, like RS5 worried that their internet service provider could “sit between my recipient and me” and switch locks to execute a man-in-the-middle attack. ET7 was one of several participants who realized that “If I send the public locks and encrypted emails using the same email provider, it’s not very secure.” ES10 asked his recipients to send back his public lock, both through Facebook and via email, so he could verify for himself that the received public locks were not altered. Other participants were concerned about making a mistake during the ongoing responsibility of managing keys. As ET10 put it, “Every time when I send or get a public lock ... there is a probability, even though not high, that my privacy is compromised. Then when I exchange public locks with many people, this probability will increase exponentially.” A few participants mentioned that compromised users could ruin the security of a whole group. ET8 said that “Within a company, if one person is hacked, then the whole company is hacked. It’s hard to track the source, just like rotten food in the refrigerator.”

Registration: Some concern but generally trusted

As expected, several participants were concerned about the need to trust email providers in the registration model. As ES5 said, having the email provider store “all public locks ... is not very comfortable.” Despite this, however, the CLMM results in Table 6 and Figure 4 indicate that participants who saw the registration model first were comfortable with it; 15 of the 20 who saw registration first agreed or strongly agreed that the model protects their privacy. Only those participants who had already heard about the more-secure exchange model were significantly less confident in the registration model’s security (12 of 20).

This general confidence in the registration model reflects many participants’ belief that even though email providers could compromise the security of the primary registration model, they would be unlikely to. Several participants mentioned that they trust their own email provider (presumably if they didn’t they would switch services). A few were specific about which kind of providers they would trust: RT8 would trust “certain big companies, not small companies,” because big companies must protect their reputations. RT10 felt similarly, with an important caveat, mentioning that big companies like “Google and Yahoo! don’t do such things [violate users’ privacy] usually, unless the government forces them to

Factor	Coef.	Exp(coef)	SE	p-value
tasks first	-0.126	0.882	0.914	0.891
second model	-2.503	0.082	1.201	0.037*
exchange	0.291	1.337	1.031	0.778
second model :: exchange	3.560	35.150	1.946	0.067

Table 5: Regression table for difficulty, Task 8. The interaction model was selected. Non-significant values are greyed out; significant values are indicated with an asterisk.

do so. In general, it’s secure.” RT2, on the other hand, preferred to trust institutions like universities that “own their own email server” to better protect her privacy.

Also contributing to the general comfort level with the registration model is that participants don’t believe most of their communication requires high security. RT4 said “encryption is not necessary for me,” and RS8 agreed, saying “If I have some private information, I won’t put it on the Internet.”

CaaS and auditing: Some added security for registration

Nineteen participants preferred the CaaS variation to the primary registration model, and eight preferred the primary model to CaaS; the rest rated the two variations the same. The most popular explanation for preferring CaaS was a belief that different companies would not collude. RS7 said that the two parties would not collude because they do not “trust each other.” Relatedly, ET7 suggested that the CaaS approach was more secure because “only when attackers attack both of the parties can they know the emails.” These comments have implications for the auditing model as well; belief that different parties are unlikely to collude and recognition that distributing trust spreads out possible points of failure would also point to more trust in the auditing model. On the other hand, two users thought the primary registration model was more secure than the CaaS variation because adding more intermediate systems and providers reduces overall security.

Those participants who were exposed to the auditing variation gave generally positive feedback. ES9 was happy that “somebody is supervising,” lock distribution and watching for problems, and ET8 appreciated that “if something goes wrong, I will be notified.” The presence of many auditors reassured participants that collusion was unlikely; for example, RT10 commented that “it’s less likely that all auditors [would] do something bad.” A few participants, however, were concerned about the reliability of the auditors: RS9 said, “I want to know who these auditors are, and whether they are truly independent.” One user (ET10) was even concerned that auditors from competing companies might have incentives to lie about each others’ behavior, making it hard to know who to trust. Two participants were concerned about the time lag for notification, noting that “a lot emails have already been sent” with even an hour’s delay (ES10). Others, however, were more pragmatic: “Immediate notification is ideal, but I don’t expect that in reality” (RT9).

Other security responses

Several participants liked that the exchange model allowed them to explicitly control who would be able to send them en-

Factor	Coef.	Exp(coef)	SE	p-value
tasks first	-0.5634	0.570	0.627	0.369
second model	-1.975	0.139	0.840	0.019*
exchange	0.292	1.340	0.786	0.710
second model :: registration	2.866	17.562	1.327	0.031*

Table 6: Regression table for privacy. The interaction model was selected. Non-significant values are greyed out; significant values are indicated with an asterisk.

encrypted email. ES2 said he would “know the person whom I sent the public locks to,” and RT3 liked that “who can send me encrypted emails [is] controlled by myself.” A similar level of control can be implemented in a registration model; our findings suggest this is a feature at least some users value.

Some participants also expressed concern about using public computers. This is potentially a problem for both encryption models, which assume the private key is securely stored on the user’s local device. ES10 expressed concern that an email encryption provider (in either model) might collect your private key, especially if you are using Apple email on an Apple device or Google email in Chrome, etc. One participant RS9 was concerned that the act of sending a lock might itself catch the interest of attackers; a similar concern was raised in [14].

Although most participants understood and reasoned effectively about the security properties we presented, a few retained incorrect mental models that have implications for the ongoing design of encryption systems. RS1 incorrectly believed that since he could not understand an encrypted message, no one else (including his email provider) would be able to either. Others were concerned about keeping their public locks secret in the exchange model; three split their locks across different channels in an effort to be more secure. For example, RS2 sent half of his public lock through the secondary email account and posted the other half on the key server. Several participants also had concerns and misconceptions about how keys are managed across multiple devices, regardless of model. System designers may want to provide help or information on these points.

Overall comparison between systems

After exposing them to both models, we asked participants whether they would use or recommend the exchange model, the primary registration model, or the CaaS registration model. Figure 5 shows that the exchange model and CaaS variation were slightly preferred to the primary registration model. The number of participants who agreed or strongly agreed to use or recommend each model were 22, 15, and 22 (use) and 24, 13, and 23 (recommend). The CLMM results (Tables 7 and 8), which take the exchange model as a baseline, show no significant difference between exchange and either variation of registration for would-use, but do show that the primary registration was recommended less frequently than the exchange model.

Factor	Coef.	Exp(coef)	SE	p-value
tasks first	-0.572	0.565	0.347	0.099
second model	-0.190	0.828	0.332	0.569
registration (primary)	-0.654	0.520	0.413	0.113
registration (CaaS)	-0.016	0.984	0.411	0.968

Table 7: Regression table for whether participants would use the model. The non-interaction model was selected. Exchange is the base case for model type. Nothing is significant.

Participants who said they would use the exchange model generally described using it for high-security information only, or only at a small scale. ES6 exemplified this trend,

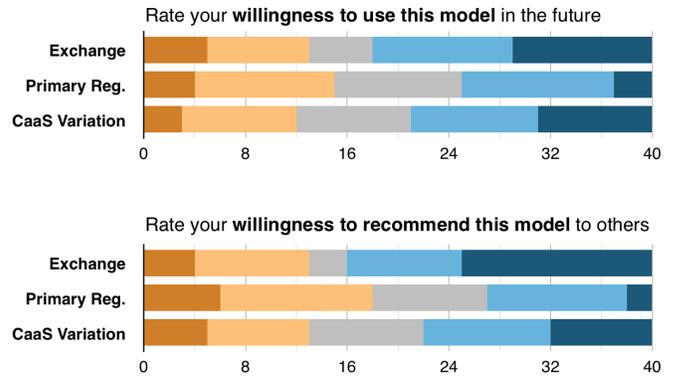


Figure 5: Participants’ ratings of whether they would use or recommend each model.

saying the exchange model is “the safest one. I want to use it in small scale, like one or two people, ... like private and personal things. But I don’t want to use it every day.” RS9 felt similarly: “I think this system is more effective with fewer people, maybe under ten. I would use it when I send my credit card information to my Mom, instead of QQ or Wechat [two instant messaging services].” ES10 said he would use the exchange model for client projects, which should be kept secret until they are finished. Among the 22 participants who agreed they would want to use the exchange model, none mentioned using it with a large group; 13 said they would use it for very private information while only one said she would use it for general or everyday emails. All 18 participants who said they would not use the exchange model cited its inconvenience.

In contrast, participants who said they would use either variation of the registration model mentioned “contacting a large number of customers” for payroll or financial information (RS9) as well as “party and meeting announcements” (ET8). RT8 said she would use the registration model for information that was “overall private, but would not be a disaster if disclosed, e.g., my daughter is sick.” ES7, a teacher, said she would use the exchange model only for “extremely sensitive information, such as SSNs,” while she would use the registration model to send “location information or grade information.” In total, 11 participants who wanted to use either variation of the registration model mentioned general email or large-scale communications.

Factor	Coef.	Exp(coef)	SE	p-value
tasks first	-0.587	0.556	0.331	0.076
second model	-0.295	0.745	0.330	0.373
registration (primary)	-1.210	0.298	0.421	0.004*
registration (CaaS)	-0.3901	0.677	0.415	0.347

Table 8: Regression table for whether participants would recommend the model to others. The non-interaction model was selected. Exchange is the base case for model type. Primary registration is significant (less recommended vs. exchange), while CaaS is not significantly different from exchange.

These results suggest that although most participants said they would use both systems at least sometimes, quite a few wanted encryption only in specific circumstances. Between the exchange and registration models, however, our participants found the registration model more broadly useful.

Using vs. recommending

As expected, most participants (34) who said they would use a system also said they would recommend it to others, and vice versa, but a few gave interesting reasons for answering differently. ET4 said he would not use the exchange model because it was too cumbersome, but would recommend it to others who have stronger privacy requirements. Similarly, RT4 said that “encryption is not necessary for me,” but recommended the CaaS variation of the registration model because it is “easier to use [than the exchange model] and more secure than the vanilla [primary] registration system.”

Registration is better than no encryption

We did not explicitly ask participants to compare these encryption models to unencrypted email. However, 19 participants who had concerns about the security of the registration model also mentioned that it does provide a valuable security improvement over unencrypted email. ET7 said “The email is not raw, which is another layer of security. ... Doing encryption gives me a security sense that I lock my door.” In line with findings from prior work [26], for some participants the process of taking any manual steps (such as generating a key pair in either model) increased their confidence that protection was occurring; for example, RS6 said “extra steps give me a security sense.”

Auditing model

Among the 12 participants who were introduced to the auditing model, six said they preferred it to any of the other models discussed. Another four said it was superior to the other registration models, but preferred the exchange model in at least some circumstances: for example, RS10 said he would send personal information including banking data using the auditing model, but “if I worked in a government department, I would still use the exchange model.” One participant liked the auditing model and exchange model equally. The final participant did not believe that auditors would actually monitor the system correctly and therefore found the auditing model least useful of any discussed. This generally positive reaction, combined with the preference to split risk among different parties in the CaaS model, suggests that the auditing model has strong potential to meet with user approval.

DISCUSSION AND CONCLUSION

We conducted the first study examining how non-expert users, briefly introduced to the topic, think about the privacy and convenience tradeoffs that are inherent in the choice of encryption models, rather than about user-interface design tradeoffs. Our results suggest that users can understand at least some high-level security properties and can coherently trade these properties off against factors like convenience. We found that while participants recognized that the exchange model could provide better overall privacy, they also recognized its potential for self-defeating mistakes. Our participants found the security of the registration model to be “good

enough” for many everyday purposes, especially when offered the option to split trust among several parties. This result is particularly encouraging for approaches like CONIKS and Google’s end-to-end extension, which spread trust among many potential and actual auditors.

We provide a few recommendations for designers of encryption tools as well as researchers, policymakers and security commentators:

- Registration models can be made even more convenient by turning them on by default, as Apple does with iMessage. Adding default encryption to more providers could achieve marginal security improvements broadly.
- Additionally, providing encryption that users do not need to go out of their way to activate will help protect them in situations where they do not think they are sending messages or information that they later decide they should have better secured.
- Showing some indication of security happening may reinforce feelings of security. Currently systems like iMessage which do encryption *completely* silently may not provide enough reinforcement to help users understand what security they have.
- Explain in clear language what the high-level risks of a given encryption approach are and trust users to make decisions accordingly. The Electronic Frontier Foundation’s *Secure Messaging Scorecard*, which tracks the security properties of encryption tools, provides an excellent start in this direction [12].
- On a similar note, alarmed denunciations of tools that do not offer perfect privacy may only serve to scare users away from any encryption at all. Instead, making clear the marginal benefit (as well as the risk) can support better decision making.
- Although most participants understood the encryption models and their security properties at a high level, there were still smaller misunderstandings that impacted their ability to make informed decisions. Despite years of effort from the security community, effectively communicating these subtleties remains difficult; however, we believe our findings demonstrate the benefits of continuing to try.

As end-to-end encryption is increasingly widely deployed, designers and companies must make choices about which models to adopt. Further work in this area—for example, testing a completely transparent registration model, examining an auditing model in greater detail, and comparing different approaches to framing security properties for non-experts—can provide further insight into how to optimize these choices.

REFERENCES

1. Apple. 2015a. iOS Security Guide, iOS 8.3 or Later. https://www.apple.com/business/docs/iOS_Security_Guide.pdf. (June 2015).
2. Apple. 2015b. We've built privacy into the things you use every day. <https://www.apple.com/privacy/privacy-built-in/>. (Aug. 2015).
3. J. Ball. 2014. GCHQ views data without a warrant, government admits. *The Guardian* (Oct. 2014). <http://www.theguardian.com/uk-news/2014/oct/29/gchq-nsa-data-surveillance>.
4. O. Bowcott. 2015. Facebook case may force European firms to change data storage practices. *The Guardian* (Sept. 2015). <http://www.theguardian.com/us-news/2015/sep/23/us-intelligence-services-surveillance-privacy>.
5. K. P. Burnham. 2004. Multimodel Inference: Understanding AIC and BIC in Model Selection. *Sociological Methods & Research* 33, 2 (Nov. 2004), 261–304. DOI : <http://dx.doi.org/10.1177/0049124104268644>
6. L. J. Camp, T. Kelley, and P. Rajivan. 2015. *Instrument for Measuring Computing and Security Expertise*. Technical Report TR715. Indiana University.
7. C. Cattiaux and gg. 2013. iMessage Privacy. <http://blog.quarkslab.com/imessage-privacy.html>. (Oct. 2013).
8. C. A. Ciocchetti. 2011. The Eavesdropping Employer: A Twenty-First Century Framework for Employee Monitoring. *American Business Law Journal* 48, 2 (2011), 285–369.
9. W. Diffie and M. E. Hellman. 1976. New directions in cryptography. *Information Theory, IEEE Transactions on* 22, 6 (Nov 1976), 644–654. DOI : <http://dx.doi.org/10.1109/TIT.1976.1055638>
10. S. Fahl, M. Harbach, T. Muders, and M. Smith. 2012a. Confidentiality as a Service – Usable Security for the Cloud. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. 153–162. DOI : <http://dx.doi.org/10.1109/TrustCom.2012.112>
11. S. Fahl, M. Harbach, T. Muders, M. Smith, and U. Sander. 2012b. Helping johnny 2.0 to encrypt his facebook conversations. In *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12)*. ACM, Article 11, 11:1–11:17 pages. DOI : <http://dx.doi.org/10.1145/2335356.2335371>
12. Electronic Frontier Foundation. 2015. Secure Messaging Scorecard. (2015). <https://www.eff.org/secure-messaging-scorecard>.
13. S. L. Garfinkel and R. C. Miller. 2005. 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 (SOUPS '05)*. ACM, 13–24. DOI : <http://dx.doi.org/10.1145/1073001.1073003>
14. S. Gaw, E. W. Felten, and P. Fernandez-Kelly. 2006. Secrecy, flagging, and paranoia: Adoption criteria in encrypted email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 591–600. DOI : <http://dx.doi.org/10.1145/1124772.1124862>
15. Google. 2014. Google End-To-End wiki. <https://github.com/google/end-to-end/wiki>. (Dec. 2014).
16. Google. 2015. Email encryption in transit. *Transparency report* (Sept. 2015). <https://www.google.com/transparencyreport/saferemail/?hl=en>.
17. B. Greenwood. 2012. The Legality of Eavesdropping in the Workplace. *Chron* (Dec. 2012). <http://work.chron.com/legality-eavesdropping-workplace-15267.html>.
18. P. Gutmann. 2004. Why isn't the internet secure yet, dammit. In *AusCERT Asia Pacific Information Technology Security Conference 2004; Computer Security: Are we there yet?* AusCERT Asia Pacific Information Technology Security.
19. D. Hedeker. 2012. Mixed Models for Longitudinal Ordinal and Nominal Outcomes. (2012). <http://www.uic.edu/classes/bstt/bstt513/OrdNomLS.pdf>.
20. C. Johnston. 2014. NSA accused of intercepting emails sent by mobile phone firm employees. *The Guardian* (Dec. 2014). <http://www.theguardian.com/us-news/2014/dec/04/nsa-accused-intercepting-emails-mobile-phone-employees>.
21. G. Kumparak. 2014. Apple explains exactly how secure iMessage really is. *TechCrunch* (Feb. 2014). <http://techcrunch.com/2014/02/27/apple-explains-exactly-how-secure-imessage-really-is/>.
22. B. Laurie, A. Langley, and E. Kasper. 2013. *Certificate Transparency*. RFC 6962. RFC Editor. <http://www.rfc-editor.org/rfc/rfc6962.txt>
23. M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman. 2015. CONIKS: Bringing Key Transparency to End Users. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, 383–398.
24. I. Paul. 2015. Yahoo Mail to support end-to-end PGP encryption by 2015. *PCWorld* (Aug. 2015). <http://www.pcworld.com/article/2462852/yahoo-mail-to-support-end-to-end-pgp-encryption-by-2015.html>.
25. B. Ramsdell. 1999. *S/MIME Version 3 Message Specification*. RFC 2633. RFC Editor. <http://www.rfc-editor.org/rfc/rfc2633.txt>

26. S. Ruoti, N. Kim, B. Ben, T. van der Horst, and K. Seamons. 2013. Confused Johnny: when automatic encryption leads to confusion and mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS '13)*. ACM, Article 5, 5:1–5:12 pages. DOI :
<http://dx.doi.org/10.1145/2501604.2501609>
27. M. D. Ryan. 2014. Enhanced Certificate Transparency and End-to-End Encrypted Mail. In *21st Annual Network and Distributed System Security Symposium, NDSS'14*.
28. S. Sheng, L. Broderick, C. A. Koranda, and J. J. Hyland. 2006. Why johnny still can't encrypt: evaluating the usability of email encryption software. In *Proceedings of the Second Symposium on Usable Privacy and Security (SOUPS '06)*.
29. D. J. Solove. 2007. 'I've got nothing to hide' and other misunderstandings of privacy. *San Diego Law Review* 44 (2007), 745.
30. S. Somogyi. 2014. Making end-to-end encryption easier to use. *Google online security blog* (June 2014).
<http://googleonlinesecurity.blogspot.com/2014/06/making-end-to-end-encryption-easier-to.html>.
31. M. Sweikata, G. Watson, C. Frank, C. Christensen, and Y. Hu. 2009. The Usability of End User Cryptographic Products. In *2009 Information Security Curriculum Development Conference (InfoSecCD '09)*. ACM, 55–59. DOI :
<http://dx.doi.org/10.1145/1940976.1940988>
32. W. Tong, S. Gold, S. Gichohi, M. Roman, and J. Frankle. 2014. Why King George III can encrypt. <http://randomwalker.info/teaching/spring-2014-privacy-technologies/king-george-iii-encrypt.pdf>. (2014).
33. M. Viswanathan. 2005. *Measurement Error and Research Design*. Sage Publications.
34. N. Weaver. 2015. iPhones, the FBI, and Going Dark. *Lawfare* (Aug. 2015). <https://www.lawfareblog.com/iphones-fbi-and-going-dark>.
35. A. Whitten and J. D. Tygar. 1999. Why Johnny Can'T Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8 (SSYM'99)*. 14–14.
36. P. Zimmermann. 1994. PGP Version 2.6.2 User's Guide. <ftp://ftp.pgp.org/pub/pgp/2.x/doc/pgpdoc1.txt>. (Oct. 1994).