What's the deal with privacy apps? A comprehensive exploration of user perception and usability

Hala Assal
School of Computer Science
Carleton University
HalaAssal@scs.carleton.ca

Stephanie Hurtado
School of Computer Science
Carleton University
Stephanie.Hurtado@carleton.ca

Ahsan Imran
School of Computer Science
Carleton University
Ahsan.Imran@carleton.ca

Sonia Chiasson
School of Computer Science
Carleton University
Chiasson@scs.carleton.ca

ABSTRACT

We explore mobile privacy through a survey and through usability evaluation of three privacy-preserving mobile applications. Our survey explores users' knowledge of privacy risks, as well as their attitudes and motivations to protect their privacy on mobile devices. We found that users have incomplete mental models of privacy risks associated with such devices. And, although participants believe they are primarily responsible for protecting their own privacy, there is a clear gap between their perceived privacy risks and the defenses they employ. For example, only 6% of participants use privacy-preserving applications on their mobile devices, but 83% are concerned about privacy. Our usability studies show that mobile privacy-preserving tools fail to fulfill fundamental usability goals such as learnability and intuitiveness—potential reasons for their low adoption rates. Through a better understanding of users' perception and attitude towards privacy risks, we aim to inform the design of privacy-preserving mobile applications. We look at these tools through users' eyes, and provide recommendations to improve their usability and increase user-acceptance.

CCS Concepts

•**Security and privacy** → **Usability in security and privacy;**
•**Human-centered computing** → *HCI design and evaluation methods;*

INTRODUCTION

Mobile privacy is becoming an increasing concern in today's society as more people are using their mobile devices to perform daily online activities and access sensitive accounts. Statistics show that the number of global mobile users surpassed the number of desktop users in 2014, with an increasing number of people switching to mobile devices for their daily online activities [15].

In this paper, we present a comprehensive look at the topic of mobile privacy. We particularly focus on three privacy aspects: private/secure messaging, privacy-aware photosharing, and anonymity. First, we seek to assess users' knowledge of mobile privacy and determine whether users rely on privacy-preserving applications (apps henceforth) to protect their privacy. Second, for users who do not, we discern whether they are consciously rejecting the use of such apps. Third, we evaluate the usability of one representative app in each of the aforementioned privacy aspects.

We conducted an online survey with 285 participants to investigate users' knowledge of privacy risks associated with the use of mobile devices, their privacy concerns, steps they take to protect their privacy, and their preferences and attitudes towards privacy-preserving apps. The survey uncovered some interesting results. For example, only 10% of the participants have taken measures to protect their anonymity. Additionally, the majority of participants regarded usability aspects to be a major requirement for privacy-preserving apps.

In addition, we evaluated the usability of three representative privacy preserving tools: ChatSecure [1], ObscuraCam [5], and Tor [8] for mobile (particularly Orbot [6], and ProxyMob [3]). All these tools were graded *A* (*excellent privacy sensitivity*) on the Privacy Grade website [17], based on their privacy-related behaviours. *ChatSecure* is an Off-the-Record Messaging application allowing two users to have an encrypted conversation; preventing eavesdropping by third parties. This app scored all seven stars on the Electronic Frontier Foundation (EFF) Secure Messaging Scorecard [7], which evaluates the security of messaging technologies. *ObscuraCam* is a secure camera app that keeps metadata associated with pictures and videos private, by removing identifying data, such as the user's location, mobile, and camera information. We chose this app because it allows obscuring photos, as well as removing private information from them. *Tor* is an Onion Routing

MUM '15, November 30-December 02, 2015, Linz, Austria
 ISBN 978-1-4503-3605-5/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2836041.2836044

system that encrypts network traffic and allows users to obtain location-anonymity by hiding among other Tor users. Orbot and ProxyMob are the official Tor-enabled tools for mobile devices. All three studies uncovered critical usability issues with these apps. For example, participants of ChatSecure mistook their conversation as encrypted on multiple occasions, when in fact it was not. ObscuraCam was sometimes unintuitive, and used overly technical language. The usability of the Tor-based apps was evaluated against nine usability guidelines. Among multiple critical issues, the guideline *be sufficiently comfortable with the user interface (UI) to keep using it* was the most violated, which potentially provides insight into users' reluctance to leverage such apps for privacy and anonymity preservation. In all three cases, users' mental models suffered due to usability issues, leading users to situations where they could unknowingly disclose private information.

The paper is organized as follows. We first offer background on mobile privacy and the three apps. Next, we discuss our survey and usability evaluations, along with some recommendations. Following, we discuss users' mental models of mobile privacy and how the usability issues uncovered in each app affect their mental models.

BACKGROUND

Many studies have centred on understanding why people seek privacy. Conti and Sobiesk [18] noticed a cultural shift in self-identified responsibility for protecting personal information: younger respondents were almost three times more likely to assume responsibility for protection than middle-aged respondents. Kang *et al.* [26] interviewed people who had previously sought anonymity and found that prior experiences, and a desire to manage both their online and offline worlds were deciding factors. Although users could purposefully opt-out of reporting information such as their location, metadata stored in multimedia (i.e., photos, videos) can still pose a privacy risk for users engaging in photosharing since it can include GPS coordinates, unique device identifiers, and textual descriptions of places [24].

Several studies [20, 25] call for the security and privacy community to advocate awareness of privacy policies for mobile applications and services. They also emphasize the need to make users aware of available privacy-preserving tools. With this in mind, we examine three categories of privacy applications: Off-the-record communication, private photo-sharing, and anonymity tools.

Off-the-Record Communication

Borisov *et al.* [14] introduced the Off-the-record messaging (OTR) protocol, to deliver a level of privacy to online social communications similar to that of face-to-face conversation [28]. The conversations are encrypted—the shared key is necessary to decrypt the content, thus conversations are protected from third party attacks [10, 14]. To access this shared key, OTR implements two authentication methods for communicating parties: shared secret and fingerprint verification [22, 23]. Studies examining OTR instant messaging desktop applications [28] revealed usability problems that could decrease privacy.

ChatSecure is an open-source OTR messaging mobile app. Following OTR protocol, messages are not kept in the system memory to prevent third party applications from discovering them. If a user loses control of their private keys, none of their previous conversations would be compromised. If an ongoing conversation is intercepted, all the attacker will see are random alphanumeric strings generated by the encryption. ChatSecure offers three methods of authentication: Shared Secret Question, QR code, and Manual Fingerprint verification.

Private Photosharing

People exhibit different photo sharing behaviours depending on who they share photos with, where the sharing takes place, and what value a picture represents to its owner. Besmer *et al.* [13] explored the needs and privacy concerns of users, resulting in a set of design considerations for tagged photo privacy. The study identifies the social tensions that tagging generates, and the needs of privacy tools to address the social implications of photo privacy management. Similarly, Ahern *et al.* [9] used context-aware camera-phones to examine privacy decisions in mobile and online photo sharing. More recently, Ames *et al.* [11] developed design criteria for mobile imaging and sharing software and online photo software by identifying emerging practices in mobile photo sharing. These findings highlight the importance of understanding sharing behaviour to design a successful photo sharing application that provides ease of use while keeping certain information private.

Obscuracam is a photo app for Android that provides secure picture messaging by removing identifying metadata stored in photos such as the user's location, mobile and camera information. Additionally, ObscuraCam automatically detects faces in pictures and highlights them with a 'tag'. When a user touches any of these tags, a small menu banner appears below the picture. It provides the user with the options to obscure, redact, invert pixelate that region, or clear the tag.

Anonymity

In 2007, Goldberg [21] surveyed different privacy enhancing techniques and found Tor [8] most popular. Still popular, an average of 5.8 million users relied on Tor, the second generation Onion Routing system, everyday in 2014 to protect their privacy [29]. Users hide among other users in the Tor network to achieve anonymity, so the degree of anonymity grows with the number of users successfully participating in the network [19]. Tor is meant for use, not only by technical users, but also by everyday users such as activists and journalists. Dingledine *et al.* [19] explain that Tor does not require modifications to Transport Control Protocol-based programs as it uses the standard SOCKS proxy interface. It can, for example, be used through a plugin to the user's Firefox browser. A user who makes mistakes while installing/using Tor software jeopardizes, not only her own privacy, but the privacy of other users in the Tor network as well [16]. Accordingly, in order to attract more users, ensure their successful participation in the network, and improve privacy, the usability of Tor tools is considered "*a security requirement*" [19]. Clark *et al.* [16] used a Cognitive Walkthrough to evaluate the usability of different Tor clients on desktop computers and identified usability

issues. Norcie *et al.* [27] evaluated and provided recommendations to improve the usability of the Tor Browser Bundle. Our work complements previous research by extending it and focusing on evaluating and providing recommendations for mobile apps.

Orbot, the official Tor app for Android devices, provides anonymity for other apps. If a user has a fully-privileged account (root), Orbot can intercept and route all outbound traffic to the Tor network. Otherwise, users must manually configure individual apps with proxies to route their traffic through Orbot. ProxyMob allows users to configure proxy settings for a Firefox session on Android devices to route its traffic through Orbot.

EVALUATION

To explore the different aspects of mobile privacy and get a more complete understanding of the topic, we conducted several evaluations. This paper presents the cumulation of multiple loosely-coordinated studies which together offer a more comprehensive view of the subject than any one study alone. However, variations in methodology are apparent as a result. We feel that the insight gained from the set of studies outweighs the differences in methodology, particularly since we are not comparing the apps against each other. All of our studies involving participants were reviewed and cleared by our institution's Research Ethics Board.

Our first goal was to investigate users' knowledge, preferences, and habits with respect to mobile privacy. To this end, we conducted an online survey with 285 participants, where we asked general questions, as well as questions more targeted towards understanding the circumstances surrounding our three privacy-preserving apps. We present a subset of the survey results in the following section. Our second goal was to examine specific mobile privacy apps covering several categories of applications. We conducted two user studies and one expert evaluation, assessing ChatSecure, ObscuraCam, and Tor for mobile devices, all open-source projects by the *Guardian Project* [4]. To the best of our knowledge, we present the first usability study for each of these mobile apps.

Our usability evaluations were guided by Whitten & Tygar's [30] principles of usable security. We reiterate their definition of usability for security: *will the mobile app's current design allow a user to realize what needs to be done, figure out how to do it, and avoid dangerous errors, without becoming so frustrated that they give up on using the mobile app altogether?*

SURVEY

Survey Methodology

We conducted an online survey with 285 participants using CrowdFlower, a crowdsourcing service [2]. Participation was restricted to *Highest quality* contributors who maintain the highest level of accuracy across CrowdFlower jobs. Each participant was paid $0.50 for completing the survey. The recruitment notice explained the purpose of the survey was to learn how people use their mobile devices in their everyday lives. We discarded data from 28 participants who entered invalid responses to survey questions. The data reported herein is from the remaining 257 valid responses.

The majority of participants (77%) completed at least a post-secondary degree, and 23% were employed in the areas of Science, Engineering, or Information Technology. There were 189 male and 68 female participants with an average age of 32.6 years (SD = 9.4). 81% of participants owned Android devices, 11% owned iOS devices, and 8% owned either Blackberry or Windows mobile devices. On average, participants have owned a mobile device for 7 years and spend 4.2 hours per day on their mobile devices.

Survey Results

Most participants (83%) reported, on a 4-point Likert scale, they were *very* or *somewhat concerned* about their personal information privacy on mobile devices. When it comes to unintended information disclosure, the top concern for 90% of participants was cybercriminals. Most participants showed motivation and a sense of responsibility towards preserving their own privacy—72% of participants reported that they were the entity primarily responsible for protecting their privacy and not app developers or Operating System manufacturers. This finding aligns with the cultural shift prediction by Conti and Sobiesk [18].

When rating the importance of different features in chatting apps on a 4-point Likert scale, the top three considered *very* or *somewhat important* were all privacy oriented: making sure they are talking to the intended person (chosen by 89% of participants), making sure no one can eavesdrop on their conversations (84%), and making sure their mobile chatting history is accessible only through chatting apps (84%).

As shown in Figure 1, the most common steps taken by participants to protect their privacy on their mobile devices were locking their device (73%) and avoiding sharing it (66%). Interestingly, these two steps only protect users' privacy against physical threats, not against *online* threats. For example, sharing a digital photo without removing metadata could reveal the user's location [24].

To investigate users' knowledge of the implications of sharing these photos, we asked participants to identify information that could be shared with photos taken by a mobile device. Participants were given eight options (6 valid). Most participants (70%) agreed that *time and date* could be automatically shared, while 49% chose the *GPS coordinate of the location where the photo was taken*. One of the invalid options, *your username*, was chosen by 23% of participants, and 13% believed no information is stored with a digital photo.

When we asked about likelihood to install a privacy-preserving app on their mobile device, 77% said they were *very* or *somewhat likely* to do so. However, fewer than 20% of participants reported having used privacy-preserving apps, such as Tor, ChatSecure, ObscuraCam, or any other on their mobile devices. Out of this small subset, only one quarter thought these apps were useful.

More specifically, fewer than 10% of participants took steps to protect their anonymity online. Approximately, 72% had

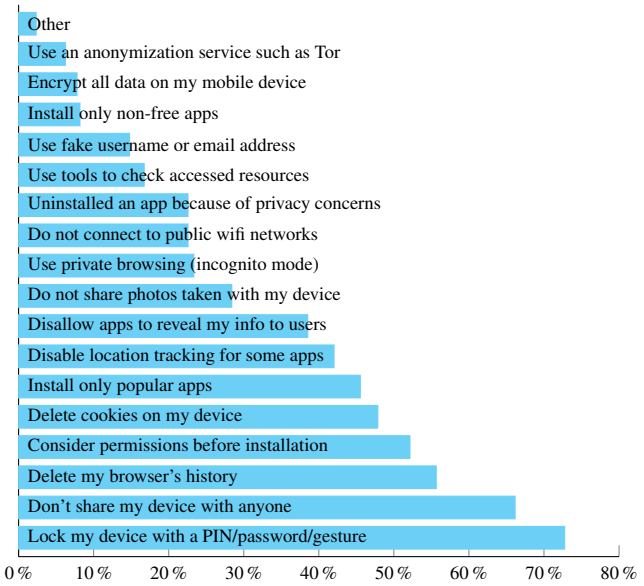


Figure 1. Percentage of participants taking each step to protect their privacy.

considered it, but either had never tried protecting their privacy or did not know how. The remaining 19% of participants have never thought about protecting their anonymity online.

We explored characteristics and features offered by privacy apps which users consider important. We asked our participants to rate different characteristics and features on a 4-point Likert scale ranging from *very important* to *not important at all*. As might be expected, the majority of participants considered *guaranteed protection* as the most important feature of a privacy app. The next three most important ones were usability-oriented. Participants wanted an app that is easy to use, easy to learn, and enables them to easily understand the status of their privacy protection. 74% of participants were willing to pay for privacy apps; half of whom were willing to pay $3 or more.

USER STUDY OF CHATSECURE

ChatSecure Methodology

We conducted a lab study consisting of 45 minute one-on-one sessions with 20 participants (9 females), each received $5 as compensation for their time. Data from one participant was eliminated because she did not have previous experience with smartphone devices. The majority of participants were familiar with the Android platform. All participants were university students from various degree programs. Four had technical backgrounds but none were majoring in computer security. The participants' ages ranged from 18-28. The average hours reportedly spent using smartphone apps was 5 hours a day, and on average participants rated their smartphone skills (i.e, familiarity with smartphone devices, comfort with downloading/using apps) as "average". Sessions unfolded according to the following steps:

1. Introduction to the study and app being tested.
2. Completion of pre-defined tasks.
3. Verbal feedback about the app and session.

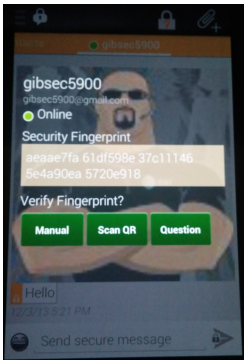


Figure 2. ChatSecure Authentication Methods

4. Completion of post-test and demographic questionnaire.

Participants were assigned six tasks relating to authenticating another user before starting a chat conversation with them. The tasks covered all three authentication methods available in the app (fingerprint, shared secret, and QR code), as seen in Figure 2. Participants tried both the "sending" and "receiving" sides of the process where appropriate. Study-specific email accounts were used for messaging between the participant and their "friends". Three Android mobile phones were used, one by the participant and two by the experimenter role-playing two different "friends". ChatSecure v.13.0.3 was tested on a LG G2 Android smartphone.

Data Collection: Two experimenters were present during all sessions; one conducted the study and role-played, while the other observed and took notes. Successful completion of each task was recorded as follows:

Success: Participant was able to complete the task without any issues or difficulties.
Success with errors: Participant was able to complete the task after multiple attempts. The task was completed with less efficiency and previous attempts may have caused security exposures.
Success but unaware: Participant was able to complete the task after multiple attempts, however, she was unaware of its completion.
False Success: Participant incorrectly thought she had completed the task.
Unsuccessful: Participant could not complete the task.

The post-test questionnaire contained both Likert-scale and open-ended questions requesting participants' opinion and perceptions of the app.

ChatSecure Results

We analyzed the performance data for the six tasks, the questionnaire data, and our observations of the sessions.

Performance: Figure 3 summarizes the performance results for each task. Many participants had difficulty with authentication, regardless of the method used. This is worrisome as this step is crucial for proper OTR communication. False Successes are especially concerning, because users are unaware their communication is unprotected.

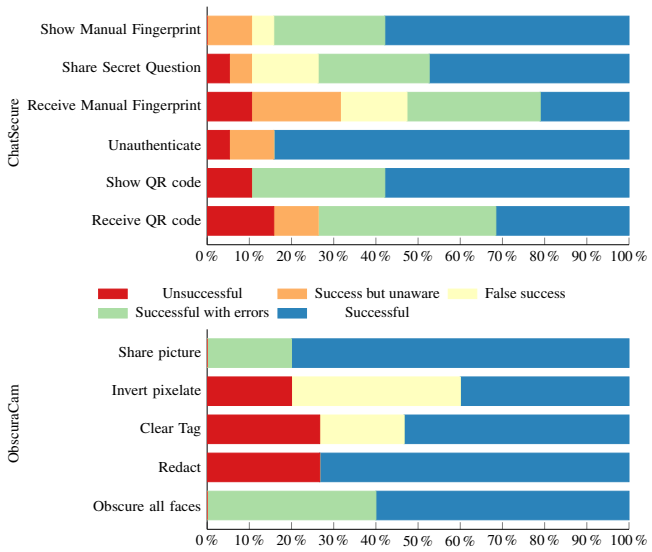


Figure 3. Task performance results for ChatSecure and ObscuraCam

Send/Receive Manual Fingerprint: This task requires pairs of users to visually compare their digital fingerprints – an alphanumeric string unique to each user– as displayed on their devices to see if they match. This task was divided into two parts to allow the participant to both verify and show their fingerprint. Only 20% of participants found the alphanumeric string and correctly verified it against their friend's fingerprint. 15% of participants just clicked 'OK' on the fingerprint verification screen without comparing it with their friend's, hence falsely authenticating. Nevertheless they believed they had successfully completed the task. 58% of participants successfully showed their manual fingerprint to the experimenters, partially because they had previously completed the first verification task for this method. 25% of the participants were confused by the menu options and initially went to the QR code method.

Share Secret Question: For this method, users establish a shared secret offline, then one user asks the other to answer the secret question online. If she answers correctly, she appears as verified on the first user's screen. For this task, participants had to answer a secret question sent by their friend (the experimenter) and state whether they thought they had answered it correctly. 85% of participants completed this task successfully. 10% were able to answer the question after multiple attempts, yet they did not believe they had done so correctly. They were confused as to whether their answer authenticated their friend or not. One participant failed the task—rather than answering the question on the pop up window, she switched to the unporotected chat and answered it there.

Send/Receive QR Code. This unique code is generated on ChatSecure's first run. The task was divided such that participants had the opportunity to (1) scan a friend's QR code and (2) show their own. To find the QR Code, participants should use the 'Fingerprint' menu item. 10% of participants did not believe they had properly verified the QR code and stated that their friend remained unauthenticated. These participants failed to notice any colour changes or checkmarks within the application. Only 30% of participants successfully completed QR code verification task without errors. The second half of the task required participants to retrieve their own QR code and show it to their friend for scanning. About 30% of participants found their QR code after first searching through the app's interface in places like the settings menu and contact list. The association between the term 'Fingerprint' and QR code was not evident, as a result, 10% of participants quit searching and had to be directed to the next task.

Unauthenticate: Since there is no unauthenticate functionality in the app, after each authentication method was completed successfully, the experimenter would in turn use the Shared Secret method to 'unauthenticate' the participant in order to continue to the next task. To accomplish this, the experimenter would ask the participant a general question (i.e., what colour is the sky?) and set the expected answer to be something unusual (i.e., orange). The participant's answer would never be correct, thus 'unauthenticating' them. When this occurred, participants should notice the switch from a secure to a non-secure conversation.

Questionnaire: The results from the post-test questionnaire showed mixed reactions to ChatSecure. Responses were skewed slightly positive, but the distribution spanned the entire scale. Table 1 shows the mean score based on responses for two representative 10-point Likert-scale questions.

Observations: We noted several usability problems leading to poor performance. First, the feedback was unintuitive. Half of participants did not notice the checkmark that appeared on the lock icon upon successful authentication so they were unaware of the app status. A colour change from orange (insecure) to purple (secure) was also difficult to interpret—users either did not notice the change or did not associate these colours with secure/insecure states.

Another major issue with the interface was inconsistency. For example, ChatSecure uses the term 'Fingerprint' to reach the QR code and also to refer to a unique alphanumeric string in the Manual Fingerprint authentication method. Results from the qualitative data demonstrated that all participants had trouble finding their QR code within the app due to this confusing terminology.

For Manual Fingerprint authentication, most users initially returned to the QR code when they saw the word 'Fingerprint'. When they actually found the correct method, 60% did not understand what the alphanumeric string meant or what they were supposed to do with it. Instead, they just clicked the 'OK' button without verifying with the other person. This lack of understanding defeats the purpose of the app, leading the user to accept a fingerprint without actually verifying the match.

Other issues were due to unintuitive behaviour from the app. We observed 25% of participants complete the Shared Secret method with errors. After a first attempt, they realized they remained unauthenticated and re-attempted authentication. Of the 53% who made errors with this method, the most common mistake was sending their answer with extra blank lines. Participants pressed what appeared be the send button, but in fact

Table 1. Mean Likert-scale scores out of 10 (most positive) for ChatSecure and ObsuraCam

	Perceived security	Overall impression
ChatSecure	6.3	6.8
ObscuraCam	8.3	8.2

was a carriage return key. This method is also case sensitive which created another type of answer mismatch.

In another example of inconsistency, five participants believed the fingerprints did not match because the letters appeared in uppercase on their screen and in lowercase on their friend's screen. Perhaps reasonably, yet mistakenly, they assumed that fingerprints are case sensitive similar to the Shared Secret Question method. Participants did not complete the authentication on the false belief that the fingerprints did not match, and instead continued with an unauthenticated conversation.

Interestingly, we noticed that the more trouble participants had with a task, the more likely they were to blame themselves or the device, rather than the app. Perhaps because participants felt they had eventually learned how to navigate through the tasks (even if incorrectly), 57% say they would consider using the app in the future. Given the number and magnitude of misunderstandings we observed, it is likely that many would have a false sense of security.

ChatSecure Recommendations

ChatSecure's interface violates several standard design guidelines. This affected participants' ability to develop an accurate mental model of the app, consequently causing some direct security exposures. We believe that a few simple changes to the interface could significantly improve ChatSecure's usability and security. We recommend simple changes such as ensuring buttons/menu options clearly communicate their functionality, and improving the clarity in the app's feedback.

More specifically, the button named 'Fingerprint' should be changed to 'QR code' to avoid confusion. Secondly, the colours for secure (orange) and insecure (purple) conversations would be more intuitive if they were red for insecure and green for secure. The app should provide feedback saying that the operation was sent successfully or received by the contact whenever a shared secret question is used. For Manual Fingerprint verification, fingerprints for both parties should have a consistent format (i.e., uppercase or lowercase) to avoid confusion while verifying. Standard interface conventions should be followed. For example, the small icon with a lock representing the send function should instead use the standard Android 'send' icon. Lastly, participants often did not noticed the colour change; a more obvious 'verified' message should confirm successful authentication.

USER STUDY OF OBSCURACAM

ObscuraCam Methodology

We conducted this lab study with 15 participants (7 females). Each participant took on average 20 minutes to complete the session, and received $10 as compensation for their time. Thirteen participants were university students and 2 were recent graduates. All were familiar with picture sharing applications

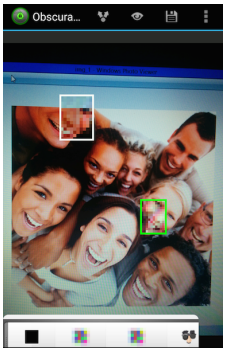


Figure 4. Tagging faces using ObscuraCam

such as WhatsApp, Instagram, and Facebook. The majority of participants were familiar with the Android platform. Four participants had technical backgrounds but none were majoring in the field of computer security. Participants' age ranged from 18-28 with the average being around 22 years. This age group was most suitable for the study as most of the online picture sharing is done by youth within this age range.

The study followed the same general methodology as ChatSecure's study. Participants completed five tasks. The tasks included taking photos with an Android smartphone and using ObscuraCam to modify them before posting to a study-specific email address. A nearby laptop displayed three images with a different number of clearly visible faces in each image. Participants took pictures of these images using ObscuraCam, completed anonymization tasks (obfuscate faces, redact faces, and inverse pixelate an image, clear metadata), and compared the metadata of ObscuraCam photos to that of photos taken with the phone's normal camera. Figure 4 shows a participant's attempt to obfuscate faces using ObscuraCam.

Data Collection: Since tasks could be completed individually, one experimenter was sufficient to observe and take notes during the sessions. The same task completion categories as in ChatSecure's study were applied to ObscuraCam and the questionnaire was similar. ObscuraCam version 2.0-RC2b was tested on a LG G2 Android smartphone.

ObscuraCam Results

As in the previous study, we examined performance, questionnaire responses, and observations.

Performance: As shown in Figure 3 , we observed no "Success but unaware" outcomes. We briefly describe participants' actions as they attempted to complete each task.

Obfuscate faces: All participants were eventually successful in obscuring faces in the first image. Although a simple task, some participants had trouble determining how to start—there was no indication of how faces could be obscured. Participants eventually realized that tapping on faces in the picture creates a box (a 'tag') inside which the face appears obscured. Nine participants successfully obscured the faces by tapping on them one by one such that only the faces were obscured. Six participants tried another approach, they tapped on the picture once to create the obscuring box and then increasing its size by pinching out on the screen to cover the entire picture. Even

though they obscured all faces, they also obscured the whole picture—we counted this as an error. When participants compared the metadata of each picture, 10 successfully noticed that sensitive details like location and phone model were missing in the ObscuraCam picture. The remaining 5 participants recognized differences in resolution, and file size, but not the important privacy-preserving differences in metadata.

Redact: In this task participants were asked to 'redact' a face. Most participants did not understand this ObscuraCam terminology and were instructed that it meant blacking out a face. Only 11 participants successfully completed this task.

Clear tag: ObscuraCam can automatically detect and 'tag' faces to indicate which ones will be obscured. We asked participants to remove some of the automatic tags, leaving only two faces obscured. Only 8 participants completed this task. Four participants could not find the "Clear Tag" button, and 3 dragged the existing tags away from the faces, thinking they had successfully completed the task.

Invert: The app can inverse pixelate an image—completely obscuring it while only the selected faces are visible. Only 6 participants successfully completed this task using the "CrowdPixel" button in the menu. Three participants failed to complete this task, while 6 tried to accomplish it by enlarging a single tag to cover the entire picture except for one face.

Share: ObscuraCam's "Obscure Photo" takes users to the picture gallery from which they can select a picture to obscure and then share. Twelve participants were successful at this task, while 3 erroneously shared an unobscured picture

Questionnaire: Participants had mixed opinions about ObscuraCam, but their responses skewed positive and were not necessarily reflective of the difficulties observed. Perhaps they felt the usability issues encountered were surmountable with additional practice. By the end of the session, most participants were satisfied with the perceived security of ChatSecure and were aware that it offers secure picture sharing by obscuring pictures and hiding private details embedded in them.

Despite existing usability and functionality issues, 10 participants would consider using ObscuraCam in the future. However, 2 participants rejected the app due to functionality issues, and 3 reported they were not concerned about the risks of picture sharing.

Observations: Eight participants formed good mental models of the app by discovering how to perform tasks on it on first use. These participants were able to discover the app's major features, as well as being able to tell the important difference in details while comparing an ObscuraCam picture with a non-privacy protected one. Four participants had slightly under developed mental models of the app—they were able to successfully obscure faces but missed features like "redact" or "invert pixelate", and they identified differences in details of the pictures but missed important ones like location. Three participants were unable to form a correct mental model of the app. They made errors while obscuring faces and could not discover any app features or differences in picture details.

It is worth noting that while participants were obscuring faces in ObscuraCam, they encountered a lot of difficulties with zooming on the picture being obscured. They also had problems resizing the tags to fit the faces. Pinching the tag outward/inward to make it bigger/smaller would sometimes result in the picture itself being zoomed in/out. This problem of resizing tags caused delays in task completion. We also observed odd app behaviour—pictures would suddenly change orientation when obscuring faces.

ObscuraCam Recommendations

In order to improve the usability of the app, we recommend the following user interface changes. First, the orientation of pictures should remain fixed. Secondly, the app should offer hints to the user on how to start obscuring faces (e.g., a notification on first-use advising the user to tap faces to obscure them, or offering a tutorial detailing the features of the app).

To improve usability, features should always be clearly visible to users. The menu is initially hidden from view and pops up only when a user touches a face to obscure it, which caused confusion among our participants. The app should also clearly list all its feature using icons in a single menu bar at the bottom of the screen. Currently, the menu banner pops up as a horizontal scroll menu with five features, only four of which are visible at a time.

We recommend renaming ObscuraCam labels to avoid confusion—changing "Redact" to "BlackOut", "Pixelate" to "Blur", and lastly "CrowdPixel" to "HideBackground". Furthermore, the icon for the "Pixelate" button should be different from the "CrowdPixel" button. Additionally, the ObscuraCam button at the top left of the screen, opposite to several participants' expectations, performs no action. Rather, it could provide all the features in a drop down list. Finally, to fix the zooming issue, we recommend making the picture resizable using the magnifying glass icon only and restricting the functionality of pinching to resizing tags.

USABILITY EVALUATION OF TOR

Tor Methodology

We studied the usability of Orbot and ProxyMob using a combination of Cognitive Walkthrough (CW) and Heuristic Evaluation (HE) [12].

We provided three evaluators with personas, core tasks to perform, and task context through scenarios. We also provided the app descriptions from the Play Store. After completing the CW, and guided by its results, we evaluate the apps' usability against a set of guidelines in a Heuristic Evaluation.

We conducted three CW sessions, each with one evaluator and one researcher, where the evaluator carried out three tasks on each app. Sessions were video recorded and the evaluators were encouraged to comment on their persona's experience. The three evaluators have background in usable security. Our study was performed on an *unrooted* Nexus 7 tablet running Android. We evaluate Orbot 12.0.5, and ProxyMob 0.0.10.

Tasks: Each persona performed three main tasks per app: (1) Install (and configure) the required components. (2) Run the

apps and configure web-traffic first to be anonymized to any location different from the real location and secondly to a specific location. (3) Disable traffic anonymizing and return to a direct connection.

Guidelines: Guided by Whitten & Tygar [30]'s definition of usable security, we also evaluated the apps against nine usable security guidelines. The first eight used by Clark *et al.* [16]. The ninth was inspired by Yee's *Path of Least Resistance* design principle [31]. The guidelines state that *Users should...*
G1 be aware of the steps needed to complete a task.
G2 be able to determine how to perform these steps.
G3 know when they have successfully completed a task.
G4 be able to recognize, diagnose, and recover from non-critical errors.
G5 not make unrecoverable dangerous errors.
G6 be comfortable with the terminology used.
G7 be sufficiently comfortable with the UI to keep using it.
G8 be aware of the application's status at all times.
G9 be guided to take secure actions.

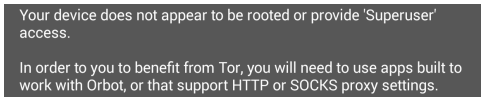
Tor Results

We highlight areas where the apps meet the usability guidelines, as well as critical usability issues. Figure 6 shows the severity of guideline violations, in terms of the number of times a guideline was missed by an app; 0 indicates full compliance with the guideline.

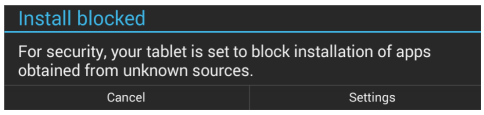
Installation and configuration - Orbot: Upon installing Orbot, the user is advised to follow the wizard for configuration. She is now aware of the task, which fulfills G1 (denoted as +G1), and how to perform it (+G2). Since Orbot alone does not guarantee Torified traffic, the wizard presents a *Warning* screen to caution the user (+G1). However, it does not provide information on how to configure other apps to use Orbot, violating G2 (denoted as −G2). The *Warning* screen is intended to protect the user from making the dangerous error of thinking her traffic is anonymous, when indeed it is not (+G5). For an unrooted device, the wizard then shows the *Permissions* screen (Figure 5(a)), which contains technical jargon (−G6) and may be confusing; the user is unsure if she needs to perform other steps before continuing with the configuration (−G2). On the next screen, the wizard provides the user with a list of recommended apps specifically developed to work with Orbot (+G6). Finally, the wizard confirms successful configuration (+G3). It is worth mentioning that when a user tries to install *Orweb* [1] through Orbot, she is directed to install the app through a user-chosen browser, after which she is presented with an error message shown in Figure 5(b). This message prompts the user to modify device's settings to allow installation from unknown sources, which makes her more vulnerable to downloading malicious software (−G9). Another important usability issue with Orbot is that the link between it and other Tor-enabled apps is unclear and unintuitive (−G7).

Installation and configuration - ProxyMob: Through Orbot, the user is presented with a list of browsers to use for downloading ProxyMob. However, installing the add-on is only

[1]Orweb is a Tor-enabled minimal web-browser. Available at https://guardianproject.info/apps/orweb/



(a) Orbot's permissions screen



(b) Orweb install blocked

Figure 5. Error messages

successful if it was downloaded from Firefox. ProxyMob suffers from a lack of feedback: (1) The user is unaware of the add-on's status after accepting the installation (−G8). (2) The user does not receive an indication of whether the add-on is enabled after restarting Firefox (−G3). If the user does not restart Firefox, ProxyMob is disabled and the user might think she is anonymous when she actually is not (−G5, −G8). When Firefox does restart, ProxyMob is enabled by default (+G5), taking the user out of the loop (+G7), and anonymizing her traffic automatically (+G5).

Torify your traffic - Orbot: Orbot is the app responsible for anonymizing traffic. On first use, it shows only vague instructions on checking the connection (−G2). Orbot's power button is unintuitive (−G7); users must long press it to (de)activate Orbot instead of the typical touch. The colour of the background logo (green) can be a reliable indicator that Orbot is activated (+G3). However, the forged location is not shown, making it difficult to verify proper anonymization (−G8). If the user does not manually check the connection through her Orbot-enabled browser (−G7), she could be risking her privacy in case of a malfunction (−G5).

To Torify traffic to a specific location, the user must specify an exit node in Orbot's Settings page, a step that a user is unlikely to realize without prior Tor-experience (−G1, −G7). In our study, neither the novice nor the expert personas completed Task 2(b) (−G4). First, the Settings page is full of technical jargon such as *exit nodes* and *obfuscated bridges* (−G6). Second, the exit node-configuration is a text box rather than a pre-populated list of available nodes, which is both incomprehensible for a novice user (−G2), and unintuitive for an expert (−G7). Additionally, the lack of feedback after specifying an exit node (−G3) forces the user to manually check the connection (−G7). This could lead to a dangerous error (−G5); the user is at risk of using an IP-address that does not correspond to her desired location.

Return to normal browsing - ProxyMob: Users are likely unaware of the need to disable ProxyMob to return to normal browsing on Firefox (−G1) and they are unaware of the required steps (−G2). Due to ProxyMob's invisibility (−G8), a user may mistakenly assume her Firefox traffic is anonymized just by activating Orbot (−G5). On the other hand, if a user wrongfully assumes deactivating Orbot is sufficient to return to normal browsing, Firefox shows a user-unfriendly message: "*The proxy server is refusing connections...*"(−G6), which may intimidate novice users (−G7). Although the message gives the user possible solutions (+G1), it does not say how to

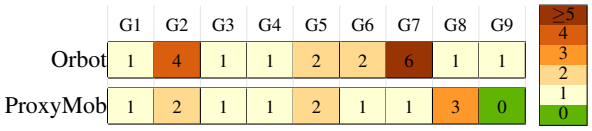


Figure 6. Severity of Tor's guideline violations

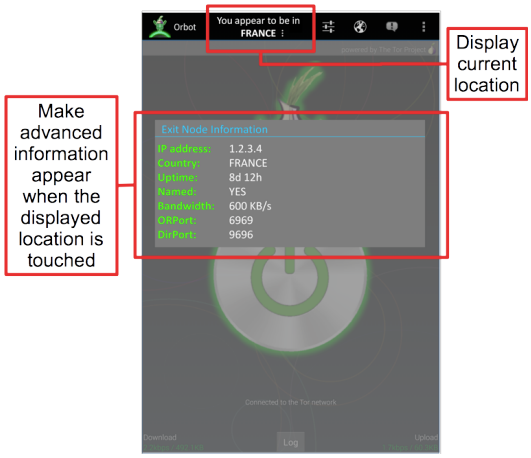


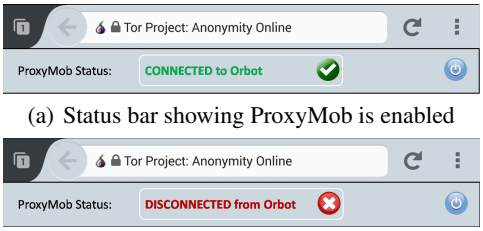
Figure 7. Orbot with our proposed modifications

perform them (−G2). Thus the user might not be able to recover from this error and end up with an unresponsive Firefox browser app (−G4).

Tor Recommendations

Our results revealed usability issues that may negatively affect the security of even the most technically-aware user. We propose modifications to the evaluated tools and general recommendations for this class of software. We aim to improve usability, help ensure the goal of anonymity is met, and help expand Tor's user base to include novice users. The most prominent issue with Orbot is that it is unintuitive to use, expecting users to have a clear understanding of Tor's underlying infrastructure. This issue is aggravated by the overall use of very technical language (e.g., obfuscated bridges, exit nodes) that is not understandable to most users and lacks appropriate detail in many instances. For example, the forged location is not shown, making it difficult to verify proper anonymization. If the user does not manually verify the connection, she could be risking her privacy in case of a malfunction.

We recommend using more natural language throughout Orbot's interface. On the *Exit Node*, we could provide users with a list of available nodes. The list should be filterable by country, bandwidth, etc. Upon choosing an exit node, users should be presented with feedback about their new location. In addition, we recommend displaying the user's forged/real location when Orbot is activated/deactivated. Advanced connection information would be available for interested users by tapping on the displayed location (Figure 7). Having location information readily available in the main screen is intuitive, eliminates the need for manually checking the connection, and makes users constantly aware of the app's status. This design potentially decreases the chances of a user erroneously assuming anonymity.



(a) Status bar showing ProxyMob is enabled

(b) Status bar showing ProxyMob is disabled

Figure 8. The proposed ProxyMob status bar

To avoid users dangerously changing the global settings of their mobile devices when installing Tor-enabled apps through Orbot, we recommend directing the user to the apps' Play Store page rather than downloading them through a user-chosen browser.

The unclear link between Orbot and Tor-enabled apps may cause users to face inconvenient situations. For example, a user deactivating Orbot may end up with a non-operational Orbot-enabled app. Thus when deactivating Orbot, the user should be alerted of apps that are configured to use Orbot and prompted to return to the apps' non-Torified mode, if possible.

The most critical issue with ProxyMob is that it is not visible to users once installed, so users have no indication of whether ProxyMob is enabled or functioning properly. Due to ProxyMob's invisibility, a user may mistakenly assume her Firefox traffic is anonymized just by activating Orbot. On the other hand, if a user wrongfully assumes deactivating Orbot is sufficient to return to normal browsing, Firefox remains in a non-operational state.

We recommend placing a ProxyMob status bar beneath Firefox's address bar to make it more visible. As shown in Figure 8, the status bar informs the user whether the add-on is enabled (i.e., the user is connected to Orbot) or disabled (i.e., disconnected from Orbot). The status bar has a power button which can enable or disable the add-on. The ideal status bar would also include information about the user's location.

Installing ProxyMob is only successful when downloaded from Firefox. However, when installing through Orbot, users are prompted to choose a browser to download the add-on. We recommend automatically downloading ProxyMob through Firefox since this is the only viable option.

DISCUSSION

Users' incomplete mental models result in a clear mismatch between their perceived risks and steps they take to protect their privacy. For instance, the majority of users are primarily concerned about cybercriminals learning unintended information, yet the most popular privacy action is locking their device with an authentication mechanism (e.g., PIN). Although a security-sound action that may be helpful against prying individuals in close proximity, locking a mobile device does not guarantee online privacy-protection.

Privacy-preserving apps should help raise users' awareness about risks they are mitigating. For example, privacy-aware photo-sharing apps could explain to users the meaning of

each metadata field that was removed from photos. We also note that current anonymizing apps provide users with network-based location-anonymity. However, from previous research [26], we know that users also use behavioural techniques to remain anonymous online, such as using fake usernames and email addresses. Thus, a potential enhancement for anonymizing apps is to support users in obtaining full anonymity; perhaps in addition to location-anonymity, they could help users manage their identities on different platforms.

Since the majority of users do not use dedicated privacy-preserving apps – although they expressed willingness to use them if available – and these apps already exist, then why are they not using them? One possible reason could be that the importance and advantages of using such apps is not properly communicated to the public; or perhaps users are unmotivated to find them, since they do not consider security and privacy as primary tasks. We could deduce that the current low adoption rate of such apps is not necessarily due to users consciously rejecting the apps. However, if users finally do decide to use current privacy-preserving apps, there is a high probability they will consciously reject them due to usability problems.

Our survey results show that users consider general usability aspects to be the most important features of a privacy app after guaranteed protection. Thus, usability issues, such as those uncovered by our usability evaluations, are likely to hinder user-adoption of these privacy apps. Unfortunately, the privacy-preserving apps evaluated herein fail to aid users in developing complete mental models of the apps or the associated privacy risks on mobile devices. We found parallels between Whitten and Tygar's [30] security properties and users' mental model development. The evaluated apps fail to address the *lack of feedback*, the *abstraction*, the *unmotivated user*, and the *weakest link* properties.

We return to the usability guidelines from our Tor study to guide more general observations of these apps.

Lack of feedback. Many participants in our usability studies could not determine if their actions were successful, violating G3. For example, in ChatSecure's Shared Secret Question Authentication task, one participant commented, "I'm waiting for it [ChatSecure] to tell me if my friend got the answer right". Due to the lack of feedback in ChatSecure's UI, participants could not recognize the security state of chats. This led to a dangerous error (violating G5)—insecure chats vulnerable to attacks and interceptions. As for Orbot, it does not inform users of the location where they appear after connecting to the Tor network, again leaving users in an uncertain state.

The *abstraction property.* The UIs of the apps we evaluated were found to be unintuitive or using complex terminologies; a violation of *both* G7 and G6. For example, ChatSecure refers to the QR code method as *fingerprint*, a technical term that everyday users could mistake for the physical fingerprint. ObscuraCam uses unfamiliar terms such as "pixelate" and "redact". Orbot is plagued with technical terminology throughout its UI and is unintuitive to use. For example, the link between Orbot and Tor-enabled apps is unclear and is hard to understand for non-expert users.

Additionally, the *unmotivated user property* is not handled properly, also violating G7. For example, it is unlikely that users would use Orweb to check the status of their connection every time they activate Orbot (violating G8). Thus, a user who ignores this step could dangerously assume her connection is anonymized when indeed it is not (violating G5).

The *weakest link property.* The three apps evaluated herein do not provide proper guidance to users to complete their intended tasks (violating G1 and G2), sometimes leading to dangerous errors. For example, if ObscuraCam did not automatically detect faces in the photo, participants could not easily realize the steps needed to obscure individual faces. In addition, when trying to email an image, some participants did not know how to complete the task and sent insecure photos.

Limitations

The user studies for ChatSecure and ObscuraCam were limited to mostly university student participants. However, since university students had trouble using these apps, it is likely that other users would too. The Tor evaluation did not involve participants, but we did consider different user backgrounds in our study. Our survey was done online which might have affected the quality of results. However, we tried to increase the quality of the survey by restricting it to highly-qualified workers and discarding responses from participants who entered invalid responses. Although our sample is skewed towards Android users, we note that the current market share is also skewed in the same direction, although not to the same extent.

CONCLUSION

We offer a comprehensive view of three mobile privacy aspects; private/secure messaging, privacy-aware photosharing, and anonymity. We discuss the topic from users' perspective through an online survey with 285 participants, as well as look into the usability of a representative app for each of these privacy aspects.

We found users (as represented by our survey participants) have incomplete mental models of privacy risks associated with mobile devices. The main reasons are (i) users' knowledge of privacy risks is mediocre at best and (ii) they are not aware of sensitive information they might be leaking through their actions, such as when sharing digital photos.

Through our explorations, we found that the majority of users who have previously used privacy-preserving apps did not find them very useful, and that these apps do not help users develop proper mental models of the mitigated risks, so users find little merit in using them. Furthermore, currently available privacy-preserving apps offer poor usability, which could contribute to their low adoption rate.

ACKNOWLEDGMENT

Sonia Chiasson holds a Canada Research Chair in Human Oriented Computer Security acknowledges the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding her Chair and Discovery Grant. Hala Assal acknowledges her NSERC Postgraduate Scholarship (PGS D).

REFERENCES

1. ChatSecure (formerly Gibberbot). https://guardianproject.info/apps/chatsecure/. [Accessed August-2015].
2. CrowdFlower. http://www.crowdflower.com. [Accessed August-2015].
3. Firefox Mobile: Privacy Enhanced. https://guardianproject.info/apps/firefoxprivacy/. [Accessed August-2015].
4. The Guardian Project. https://guardianproject.info. [Accessed August-2015].
5. ObscuraCam: Secure Smart Camera. https://guardianproject.info/apps/obscuracam/. [Accessed August-2015].
6. Orbot: Mobile Anonymity + Circumvention. https://guardianproject.info/apps/orbot/. [Accessed August-2015].
7. Secure Messaging Scorecard. Which apps and tools actually keep your messages safe? https://www.eff.org/secure-messaging-scorecard. [Accessed August-2015].
8. Tor Project: Anonymity Online. https://www.torproject.org. [Accessed August-2015].
9. Shane Ahern, Dean Eckles, Nathaniel S. Good, Simon King, Mor Naaman, and Rahul Nair. 2007. Over-exposed?: Privacy Patterns and Considerations in Online and Mobile Photo Sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 357–366. DOI:http://dx.doi.org/10.1145/1240624.1240683
10. Chris Alexander and Ian Goldberg. 2007. Improved User Authentication in Off-the-record Messaging. In *Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society (WPES '07)*. ACM, New York, NY, USA, 41–47. DOI: http://dx.doi.org/10.1145/1314333.1314340
11. Morgan Ames, Dean Eckles, Mor Naaman, Mirjana Spasojevic, and Nancy Van House. 2010. Requirements for mobile photoware. *Personal and Ubiquitous Computing* 14, 2 (2010), 95–109. DOI: http://dx.doi.org/10.1007/s00779-009-0237-4
12. Hala Assal and Sonia Chiasson. 2014. Extended Abstract: Will this onion make you cry? A Usability Study of Tor-enabled Mobile Apps. In *Tenth Symp On Usable Privacy and Security (SOUPS)*. https://cups.cs.cmu.edu/soups/2014/posters/soups2014_posters-paper27.pdf
13. Andrew Besmer and Heather Richter Lipford. 2010. Moving Beyond Untagging: Photo Privacy in a Tagged World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1563–1572. DOI: http://dx.doi.org/10.1145/1753326.1753560
14. Nikita Borisov, Ian Goldberg, and Eric Brewer. 2004. Off-the-Record Communication, or, Why Not to Use PGP. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society (WPES '04)*. ACM, New York, NY, USA, 77–84. DOI: http://dx.doi.org/10.1145/1029179.1029200
15. Danyl Bosomworth. Statistics on mobile usage and adoption to inform your mobile marketing strategy. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/. [Accessed August-2015].
16. Jeremy Clark, Paul C Van Oorschot, and Carlisle Adams. 2007. Usability of anonymous web browsing: an examination of TOR interfaces and deployability. In *Symp on Usable Privacy and Security*. ACM, 41–51.
17. CMU. PrivacyGrade: Grading The Privacy Of Smartphone Apps. http://privacygrade.org/home. [Accessed August-2015].
18. Gregory Conti and Edward Sobiesk. 2007. An Honest Man Has Nothing to Fear: User Perceptions on Web-based Information Disclosure. In *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS '07)*. ACM, New York, NY, USA, 112–121. DOI: http://dx.doi.org/10.1145/1280680.1280695
19. R Dingledine. 2004. Tor: The Second-Generation Onion Router. In *USENIX Security Symp*.
20. Gerald Friedland and Robin Sommer. 2010. Cybercasing the Joint: On the Privacy Implications of Geo-tagging. In *Proceedings of the 5th USENIX Conference on Hot Topics in Security (HotSec'10)*. USENIX Association, Berkeley, CA, USA, 1–8.
21. Ian Goldberg. 2007. Privacy enhancing technologies for the Internet III: Ten years later. *Digital Privacy: Theory, Technologies, and Practices* (2007), 3–18.
22. Ian Goldberg, Chris Alexander, and Nikita Borisov. Off-the-Record Messaging: Authentication. http://www.cypherpunks.ca/otr/help/authenticate.php?lang=en. [Accessed August-2015].
23. Ian Goldberg, Chris Alexander, and Nikita Borisov. Off-the-Record Messaging: Fingerprints. http://www.cypherpunks.ca/otr/help/fingerprints.php?lang=en. [Accessed August-2015].
24. Google+ Help. Show where your photos were taken. https://support.google.com/plus/answer/6008918?hl=en. [Accessed August-2015].
25. Benjamin Henne, Christian Szongott, and Matthew Smith. 2013. SnapMe if You Can: Privacy Threats of Other Peoples' Geo-tagged Media and What We Can Do About It. In *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '13)*. ACM, New York, NY, USA, 95–106. DOI: http://dx.doi.org/10.1145/2462096.2462113

26. Ruogu Kang, Stephanie Brown, and Sara Kiesler. 2013. Why Do People Seek Anonymity on the Internet?: Informing Policy and Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2657–2666. DOI: http://dx.doi.org/10.1145/2470654.2481368

27. Greg Norcie, Jim Blythe, Kelly Caine, and L Jean Camp. 2014. Why Johnny Can't Blow the Whistle: Identifying and Reducing Usability Issues in Anonymity Systems. In *Network and Distributed System Security Symp (NDSS) Workshop on Usable Security (USEC)*. DOI: http://dx.doi.org/10.14722/usec.2014.23022

28. Ryan Stedman, Kayo Yoshida, and Ian Goldberg. 2008. A User Study of Off-the-record Messaging. In *Proceedings of the 4th Symposium on Usable Privacy and Security (SOUPS '08)*. ACM, New York, NY, USA, 95–104. DOI: http://dx.doi.org/10.1145/1408664.1408678

29. Tor Metrics. Estimated number of clients in the Tor network. https://metrics.torproject.org/clients-data.html. [Accessed August-2015].

30. Alma Whitten and J Doug Tygar. 1999. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *USENIX Security Symp*.

31. Ka-Ping Yee. 2002. User Interaction Design for Secure Systems. In *Information and Communications Security*. Springer Berlin Heidelberg, 278–290. DOI: http://dx.doi.org/10.1007/3-540-36159-6_24