Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 3928261, 15 pages
https://doi.org/10.1155/2017/3928261

Research Article

Privacy Enhancing Keyboard: Design, Implementation, and Usability Testing

Zhen Ling,[1] Melanie Borgeest,[2] Chuta Sano,[3] Jazmyn Fuller,[4] Anthony Cuomo,[5] Sirong Lin,[3] Wei Yu,[6] Xinwen Fu,[7] and Wei Zhao[8]

[1] *Southeast University, Nanjing, China*
[2] *University at Albany, SUNY, Albany, NY 12222, USA*
[3] *UMass Lowell, Lowell, MA 01854, USA*
[4] *The City University of New York, Queens, NY 11367, USA*
[5] *Quinnipiac University, Hamden, CT, USA*
[6] *Towson University, Towson, MD 21252, USA*
[7] *University of Central Florida, Orlando, FL 32816, USA*
[8] *University of Macau, Macau*

Correspondence should be addressed to Zhen Ling; zhenling@seu.edu.cn

Received 4 August 2017; Accepted 16 October 2017; Published 8 November 2017

Academic Editor: Zhipeng Cai

To protect users from numerous password inference attacks, we invent a novel context aware privacy enhancing keyboard (PEK) for Android touch-based devices. Usually PEK would show a QWERTY keyboard when users input text like an email or a message. Nevertheless, whenever users enter a password in the input box on his or her touch-enabled device, a keyboard will be shown to them with the positions of the characters shuffled at random. PEK has been released on the Google Play since 2014. However, the number of installations has not lived up to our expectation. For the purpose of usable security and privacy, we designed a two-stage usability test and performed two rounds of iterative usability testing in 2016 and 2017 summer with continuous improvements of PEK. The observations from the usability testing are educational: (1) convenience plays a critical role when users select an input method; (2) people think those attacks that PEK prevents are remote from them.

1. Introduction

The touch-screen enabled devices have been a popular target of network attacks. Sensitive information like the passwords entered on mobile devices can be stolen by attackers by exploiting the soft keyboard. For example, in residue-based attacks [1–5], the tapped keys can be inferred from the oily or heat residues left on the touch screen, the order of which can also be determined by measuring the heat residue left on the touched positions. In computer vision-based attacks [6–12], the interaction between the hand and the keyboard is exploited. The hand movement and the finger position indicate which keys are being touched [10, 11, 13, 14]. In sensor-based attacks [15–21], with the help of the accelerometer (acceleration) and gyroscope (orientation), a malware senses the slight motion of a device when the user types different keys.

To defeat the attacks introduced above, we invent a novel context aware privacy enhancing keyboard (PEK) for touch-enabled devices. It can be observed that these attacks are possible in that the keys of the keyboard are at the fixed position. With PEK, we shuffle the positions of the characters on the keyboard. Whenever a user of a touch-enabled device is to type the password, a randomized keyboard is shown to her. In other words, the user is presented with a randomly shuffled keyboard each time she taps a password. We maintain the usability of PEK through a context aware feature: a randomized keyboard shows up only when a user taps a password or pin. PEK shows a normal QWERTY keyboard or a system default one when a user inputs text like an email or a message.

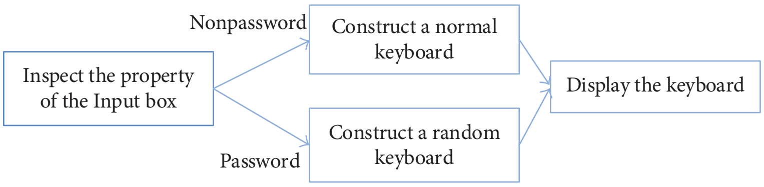

Inspect the property of the Input box
Nonpassword
Password
Construct a normal keyboard
Construct a random keyboard
Display the keyboard

Figure 1: Workflow of PEK constructing a keyboard.

We are the first to design a generic randomized keyboard for the Android system though the idea of randomizing the key layout is not new [22]. One version of PEK is implemented as a third party keyboard for Android and can replace the system keyboard once it is installed. Therefore, once PEK is chosen as the default keyboard, it can be used by any app.

After our presentation at Black Hat USA [10], we released PEK as a free Android app to Google Play in August 2014. Until the time of writing, it has been downloaded 2352 times. We have released 7 versions with corrected bugs and improved interface. Of them, PEK 1.0 is based on an Android code example. PEK 2.x.x and later versions are based on OpenWnn [23] with fixed bugs. The current version of PEK is 3.2.3.3.

For the purpose of usable security and privacy, we designed an iterative usability test to evaluate the user experience of PEK and to explore the reason for the lukewarmness of using PEK. Each iteration of usability test is a two-stage study: a pilot study and a main study. We randomly select participants to reflect different behaviors of Android users. The pilot study uses surveys and interviews and involves a small number of people for us to understand potential usability issues of PEK. We then add features to PEK based on the results of the pilot study and use the main study through a web survey hosted at Amazon Mechanical Turk to understand the usability of the improved PEK. For surveys performed through Amazon Mechanical Turk, we contact the users to make sure that they install PEK and complete our survey questions. We performed two rounds of usability test in 2016 summer and 2017 summer, respectively. After the two rounds of usability testing and app improvements, most users report the app is easy to install, configure, and use. The other observations from the usability testing are also educational: (1) convenience plays a critical role when users select an input method; (2) people may think those attacks that PEK prevents are remote from them and the risk from those attacks is small. The usability test demonstrates the worrisome phenomena that many users blindly trust their phones for security or are not much concerned with the possible breaches. These phenomena demonstrate the human factor that contributes to the vulnerabilities of the cyber space.

An early version of the usability testing of PEK is presented in [24]. In the conference version, we design a two-stage usability test and perform the first usability testing to evaluate the user experience of PEK. In this journal version, we present the detailed algorithms and techniques used to implement our privacy enhancing keyboard and the second usability testing to further evaluate the user experience of updated PEK. Related work and new experimental evaluation are presented in this extended version.

The rest of this paper is organized as follows. The design and the implementation of the third party keyboard of PEK will be introduced in Section 2. In Section 3, we present the methodology applied in the usability test and the results from the first usability testing. The second usability testing and results are presented in Section 4. We review the most related work in Section 5 and conclude this paper in Section 6.

2. Privacy Enhancing Keyboard

We introduce the basic idea of the privacy enhancing keyboard in this section. Then we present the technique details of PEK implementation and elaborate how to install and configure the PEK for Android.

2.1. Basic Idea. To thwart diverse attacks such as residue-based attacks, computer vision-based attacks, and sensor-based attacks, we shuffle the positions of the keys of a software on a touch screen to ensure that a randomized keyboard is shown to a user every time she enters a password. Therefore, the particular keys will not be easily inferred even though there is finger, oily, or thermal residue left on the screen. Moreover, vision-based attacks do not work since a touched position by a finger no longer matches a fixed key. The vibration and orientation information provided by an accelerometer does not help establish profiles for particular keys.

Figure 1 shows the basic idea and the workflow of PEK constructing a keyboard when a user touches an input box. The first step is to inspect the property of this input box to see whether it is a nonpassword or password input box. A QWERTY keyboard is shown if a nonpassword input box is identified. If a password input box is identified, we parse the property of the keys from a XML file, which stores the layout of the keyboard, and then change the label and value of the keys so that the positions of the keys are shuffled.

There are two versions of PEK that we implemented. (1) One version is a third party keyboard that can be implemented as an Android App for Google Play and installed to an Android device. Once enabled, PEK runs as an Android service in the background. A user can use the Android input setting menu to enable PEK while the location of the input setting menu could vary from device to device. Before PEK 3.0, it is the user's responsibility to find the input setting menu according to the generic introductions we provided at Google Play. (2) We are also able to revise the source code of the Android system default keyboard and the unlock screen keypad, recompile the entire Android project, and flash the system into the device. Apparently, the usability of such an implementation is an issue since most users do not have the capability to recompile the Android system and flash it into their devices. For completeness, we also introduce such an implementation of the unlock screen keyboard while the focus of our usability testing will be the third-party keyboard version of PEK.

We have two challenges for implementing a useful privacy enhancing keyboard.

(i) First, how can we generate a randomized keyboard? That is, what is our privacy enhancing technology?

(ii) Second, how do we identify the type of input box in order to show an appropriate keyboard? That is, how do we implement the context aware technology?

We answer these two questions in the following subsections.

2.2. Privacy Enhancing Technology. A general software keyboard contains three components, denoted as subkeyboards. The primary subkeyboard is the QWERTY keyboard, which is the most common keyboard layout. The second subkeyboard is the numerical keyboard that may also contain some symbols. The last subkeyboard is a symbol keyboard that contains special symbols. The layout of these three subkeyboards is stored in a XML file, which records the positions, labels (the text to display), and values (such as ASCII codes) of the keys. This keyboard is often used as the system default keyboard. The system generates the default keyboard in this way: the keys will be read from the XML file one by one and put at a right position.

To randomly shuffle the keyboard, we just need to change the label and value at a particular position on a keyboard that is defined in a XML file. Recall that an XML file defines a sequence of *position, label,* and *value*, corresponding to all the keys. We do not change the overall size of the keyboard and just change *label* and *value* at *position*. Figure 2 illustrates the randomly shuffled keyboard.

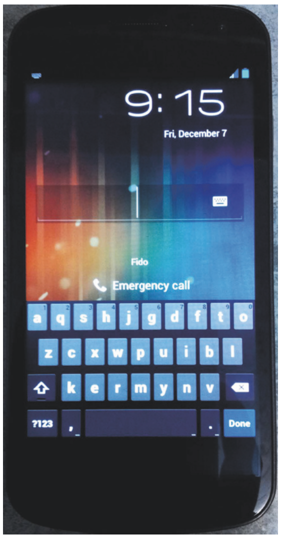


Figure 2: PEK.

2.3. Context Aware Functionality. To correctly identify the type of the input box, we can take advantage of the input box properties. The Android class "EditorInfo" can be used to detect the type of input box. In our case, "TYPE_NUMBER_VARIATION_PASSWORD," "TYPE_ TEXT_VARIATION_PASSWORD," "TYPE_TEXT_ VARIATION_VISIBLE_PASSWORD," and "TYPE_TEXT_ VARIATION_WEB_PASSWORD" are used to identify the password input box. The first type is the variation of "TYPE_CLASS_NUMBER," while these late three types are the variations of "TYPE_CLASS_TEXT." Once the password input box is triggered by the user, a new randomized keyboard will be constructed. As a result, the user presented a different randomized keyboard every time she presses the password input box.

2.4. Randomization of the PIN Keypad. Although a third party keyboard can be chosen as the keyboard for the unlock screen, we also design and implement a randomized pin keypad for the unlock screen. The implementation of such a keypad involves the revision of the Android system files. For completeness, we will present its design and implementation. However, we will skip it for the usability testing since it is impossible for general users to change the Android source code.

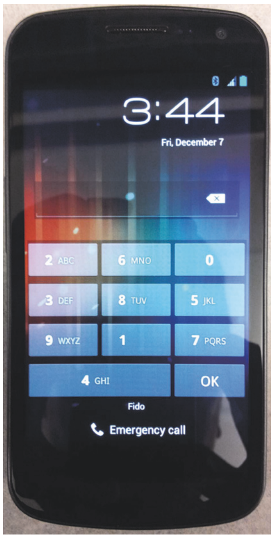


Figure 3: Randomized keypad.

2.4.1. Algorithm. Once the PIN mode is selected as a screen lock scheme, the system will provide a soft minikeyboard, that is, keypad, for the user to input a 4-digit pin. There are 10 digits on the keypad. The key icon is used to show the digits on the buttons instead of a key label. As a result, we need to not only randomize the key code but also change the corresponding icons for keys at different positions. Figure 3 illustrates the randomized keypad on a Google Galaxy Nexus phone.

Table 1: Input time and success rate.

	Normal keyboard	Shuffled keys
Median input time (seconds)	2.235	5.859
Success rate	98.50%	98.83%

2.4.2. Implementation. As shown in Figure 3, an 11-button keypad would be used if the PIN mode is set up as a screen unlock scheme. This keypad is a specially designed keyboard for the PIN mode instead of a keyboard for the system default input method. We revised the overridden method "createKeyFromXml()" in the code file "PasswordEntryKeyboard.java" to modify the key properties after the key constructor is called. However, the digit shown on the button in Figure 3 is a key icon. Consequently, we need to modify the key codes and corresponding key icons rather than key labels. We store the values of the key icons, that is, R.drawable.sym_keyboard_num1, R.drawable.sym_keyboard_num2, and so forth, into an array. We also use the method Resources.getDrawable to derive the specific key icon and replace the original key icon. Finally, we recompile the source code of the entire Android project to implement this functionality.

2.5. Installation and Configuration. We implement the PEK and release it on the Google Play Store. PEK can be found by searching for either "PEK" or "privacy enhanced keyboard" on the Google Play Store. The downloading process should be fast and relatively quick. At Google Store, we give a general introduction to how to configure the settings of an Android device and use PEK.

2.6. Evaluation of Input Time of PEK. To measure the input time of the PEK, we recruit 20 students, 5 female students, and 15 male students, whose average age is 25 years old. We implemented a test password input box and generated 30 random four-letter passwords. The students were required to input these 30 passwords using a QWERTY keyboard and a shuffled keyboard, and the test app recorded the user input time. Table 1 shows the results of our evaluation and Figure 4 gives a box plot of the input time of the two different keyboards. The median input time is around 2.2 seconds on the QWERTY keyboard and 5.9 seconds on the shuffled keyboard. The success rates of users inputting four-letter passwords on both keyboards are high, except for the PEK with a lowest rate. The participants in our experiments think PEK is acceptable if it pops up the randomized keyboard only for sensitive information input.

3. First Usability Testing

In this section, we introduce our two-stage usability study of PEK: the pilot study and the main study. The first such usability testing was performed in 2016 summer. Though similar to the former, the latter differs from it in the greater number of participants, questions, and other measurements. Generally speaking, it is not necessary to involve many participants in either the interview or the focus group study. During the process of study, we keep a good balance of the qualitative and the quantitative results. Besides the traditional qualitative research such as interview and focus group, we apply various methods to get quantitative information about users from different aspects. For example, the face-to-face interaction with participants enables us to get detailed information about their views on the interview questions while a web survey covers a larger number of target users and provides quantitative and statistic results.

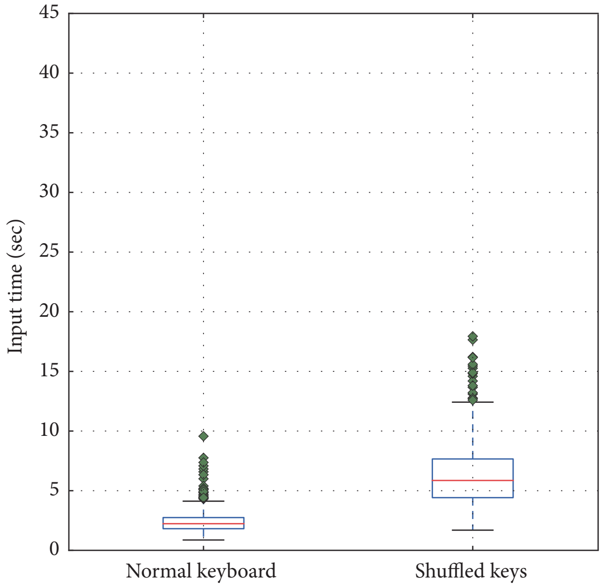


Figure 4: Input time of two distinct keyboards.

3.1. Pilot Usability Test

3.1.1. Methodology. There are two sessions in the pilot usability test, serving as the base and precondition to the later main usability test. The first session consists of a presurvey with 10 questions, an interview with 5 open ended questions, and a postsurvey with 4 questions. Multiple-choice questions are common in the two surveys with easily interpreted and classified answers. The second session follows two or three days after the first one, which is composed of 10 open ended questions. The interview is designed for the record of the answers and partly for the timing of participants installing and configuring PEK. The pilot study tries to address three major issues as follows.

(i) *PQ1*: after the release of PEK, some users complain on Google Play about the difficulty of the configuration process. Thus, we would like to investigate into the following two questions. How easily can users install and configure PEK onto their smart devices? Does the complicated installation and configuration process discourage them from using PEK?

(ii) *PQ2*: due to the lack of interest and awareness of protecting privacy, people may have little demand for PEK. Such inference reminds us of another question:

Table 2: Installation and configuration time of PEK.

Participants	Installation time (seconds)	Configuration time (seconds)
Participant 1	29.01	45.79
Participant 2	15.00	125.00

do all the users care about the security on their smart devices?

(iii) *PQ3*: once a user makes PEK work, she will meet with a randomized keyboard every time she chooses a password input box, which takes more time than typing in a regular QWERTY keyboard. Here comes the question: do all the users agree with the point of view that it is worth taking extra time to protect their passwords and/or pins?

3.1.2. Results for Pilot Usability Test. Two males with Android mobile smart phones participate in the pilot usability test. They are required to install and configure PEK on their devices and we time them. We measure how long they spend on finishing the installation and configuration and how long it takes for the randomized keyboard to successfully show up when participants try to input a password and/or pin.

Answers to Question PQ1. Users have no difficulty in finding PEK on Google Play and installation. Nevertheless, they do have problems in configuring it. Table 2 shows the time of installation and configuration during the pilot test. Apparently both spend more time on configuration. It is the researchers who give them additional instructions and help them successfully configure PEK. The participants fail to find a PEK application icon and get confused when the randomized keyboard does not show up when they log in to one of their accounts like an email. The complicated configuration process frustrates the participants and discourage them from configuring PEK.

Answers to Question PQ2. Neither of the participants have any security enhancements on their smart phones. Thus, they think it is unnecessary to use PEK since there is no sensitive information on their phones. According to Participant 1, using applications and services which request important or sensitive information on laptop or desktop instead of smart devices can be regarded as his only way of the security precaution. However, both the participants admit they are among target audience of PEK for they are educated about mobile security and precautionary measures.

Answers to Question PQ3. After two to three days in the second session of the test, Participant 1 and Participant 2 hold different views on whether the extra time they spend is worth protecting privacy. Participant 1 predicts that nobody would prefer a randomized keyboard with no keys in the fixed position than a regular QWERTY keyboard with keys in the same position, which is familiar to users. Using PEK is a challenge to multitask. For instance, if a user is on the walk, typing in a randomized keyboard is rather difficult. Using PEK wastes time, especially when the mobile phone goes sleep again and again when users attempt to enter their password. The repeated action of entering password and the wasted time frustrates Participant 1. Different from Participant 1, Participant 2 holds positive views on the use of PEK for its practicability and dependability. He regards PEK as a hand that covers the password, sparing users' trouble of covering with their own hands.

Two observations can be made from the pilot usability test.

(1) The configuration of PEK is a great challenge for both participants, which demands more instructions on the Google Play Store for users to follow and an icon for them to click when opening PEK. As can be seen from the test, neither of the participants succeeds in using PEK without the help of researchers because they waste time looking for a nonexistent icon.

(2) Since Participant 1 mentions the difficulty of using PEK when unlocking mobile phones with multiple tasks, we decide to create a separate button on the privacy enhanced keyboard disabling PEK quickly. In this way, if a user would rather use a regular QWERTY keyboard than a randomized one when unlocking the mobile phone, the button should help him.

3.2. Main Usability Test

3.2.1. Methodology. The main usability test, composed of a web survey and a focus group usability test, is based upon the findings in the pilot test. The web survey is conducted based on the Qualtrics platform on Amazon Mechanical Turk. Participants are required to follow directions and answer questions honestly and correctly with a bonus of one dollar. The focus group usability test involves an interview targeting participants who install and configure PEK on their devices and are required to answer several questions. In this test, the following four major issues are addressed.

(i) *MQ1*: what are the most frequent activities of the smart phone users? If one of the most frequent activities they do have anything to do with privacy, the users should be included as our target audience.

(ii) *MQ2*: have the smart device users already had an awareness of utilizing default security precautions? Similar questions are covered in the pilot test such as whether or not typical smart device users are concerned with the security measures on their personal devices.

(iii) *MQ3*: do users consider that their smart devices are properly protected from outsider attacks?

(iv) *MQ4*: do any smart device users think about taking more measures to ensure security of their devices?

3.2.2. Results for Main Usability Test: Web Survey. The main usability test involves 2 participants in the focus group usability test and 266 participants, including 132 females and 134 males, in the web survey. Their ages range from 18 to above

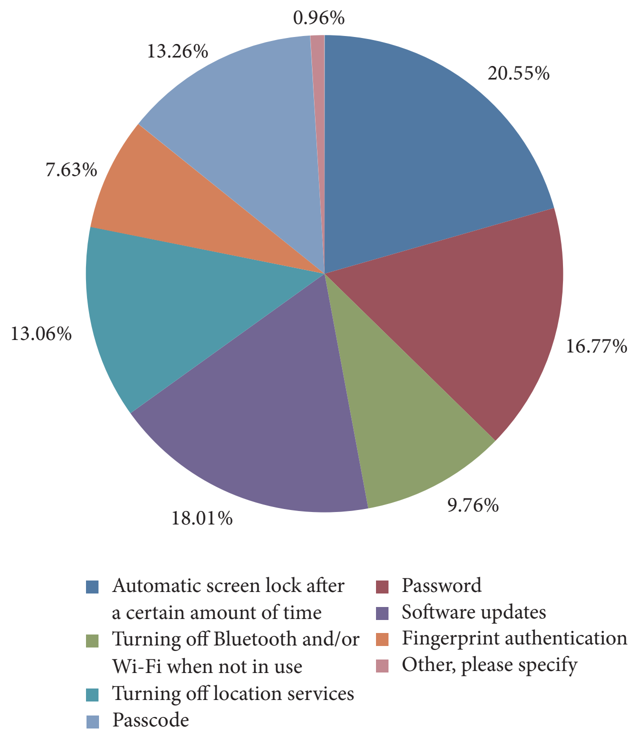


Figure 5: Distribution of security precautions.

50 years old. 136 participants use Android devices, which PEK is compatible with. 123 participants use Apple devices, with the rest 7 participants using other devices. The web survey consists of 21 questions and 266 responses as well as multiple choice questions with open ended questions.

Answers to Question MQ1. The aim of this question is to find out whether the most frequent activities performed by mobile smart device users involve their personal sensitive information. Mobile banking, online shopping, and social network increase the possibility of sensitive personal information being stolen. Figure 6 depicts the statistics from the web survey. Internet use is at the top with 8%. 5.4% of the web survey participants shop online. 5.7% of them use mobile banking and 7.1% use social networking sites. All the three activities may contribute to personal information being leaked and an account being hacked. If participants intend to protect their information involved in the activities, they should be a part of PEK's target audience.

Answers to Question MQ2. A user who has no other security precautions on her device is not likely to utilize PEK. What matters most is not the amount of security precautions, but the users' awareness of protecting their personal information from the potential attacks. Figure 5 illustrates the distribution of security precautions web survey participants implement on their devices. At 20.55%, automatic screen lock after a certain amount of time is the top answer. More questions therefore arise after the results of these particular questions. Are smart device users unconcerned with security? Or, are they uninformed of the security problems on the devices and the potential attacks?

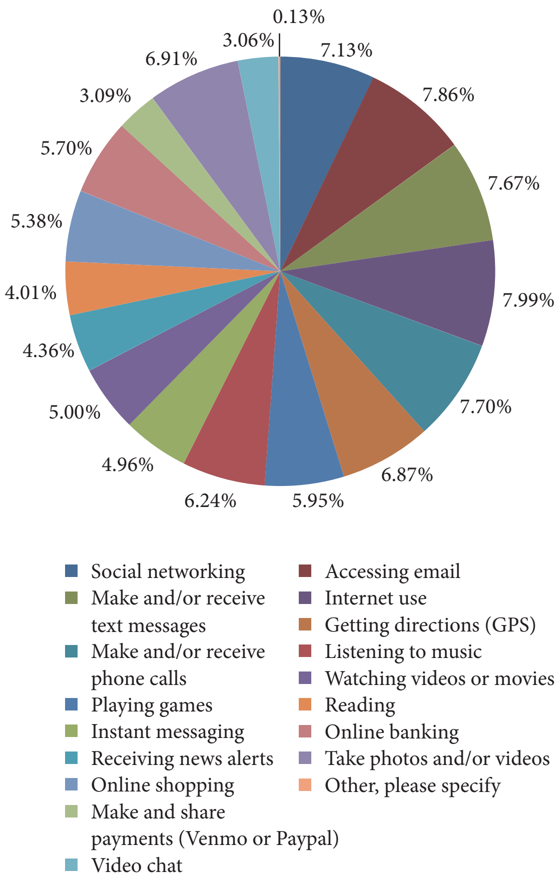


Figure 6: Answers to question MQ1.

Answers to Question MQ3. This question is designed to figure out whether or not the web survey takers are aware of the potential attacks to their own smart devices. Based on the results, we can have judgment between two reasons for users' low awareness of security-lack of education about attacks and unconcern with security. The answers to the question vary by the degree to which the web survey takers are concerned with security. The top answer at 36.59% is "probably yes," followed by "maybe" at 29.27% and "probably not" at 20.05%. It is noteworthy that the rate of the degree of protection on the mobile devices might not match how well they are really protected. What worries us is exactly the high level of certainty they show about protecting their smart devices. Figure 7 demonstrates the distribution of answers to the question of how well protected their smart devices are.

Answers to Question MQ4. It surprises us a lot that users show great interest and willingness in taking more measures to protect their devices from attacks. Despite that few of them really implement more security precautions, such a result could be a good beginning. Figure 8 shows the distribution of the answers to this question. 37.67% of the web survey takers answered "probably yes," with 30.62% of "maybe" and 19.24% of "definitely yes." These groups of people can be potential PEK users under the premise of ensured user experience and security.

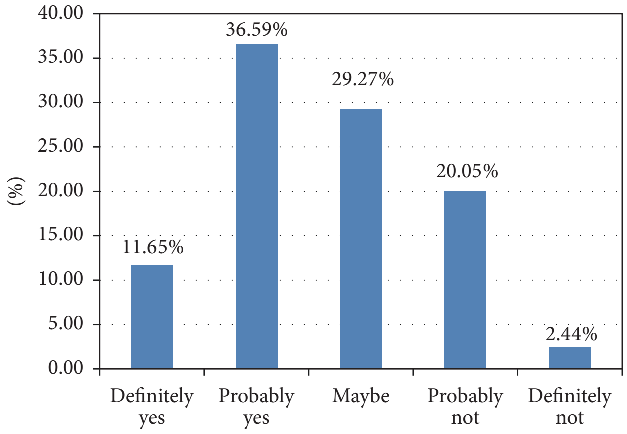


Figure 7: Distribution of answers to question MQ3.

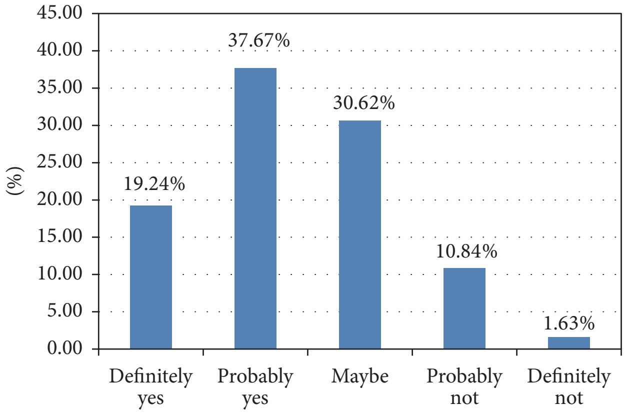


Figure 8: Distribution of answers to question MQ4.

3.2.3. Results for Main Usability Test: Focus Group Usability Test. Besides the web survey mentioned above, the focus group usability test targets 2 participants. They are interviewed at the same time with 19 open ended questions, similar to those asked in the web survey. Both use mobile Android smart phones.

(a) *What three activities do you primarily do on your mobile phone?* Participant A's list of most frequent activities contains using the alarm, reading the news, and listening to music. The top three activities Participant B performs on the mobile smart device are sending/receiving texts, taking photos, and using social network applications. Participant B is more likely to be a candidate for PEK than Participant A. None of the activities they listed are frequently chosen by the web survey takers.

(b) *What kind of security have you implemented on your mobile phone?* Both the participants answered "nope" to this question. Neither has installed any default security precautions to their smart devices.

(c) *Are you satisfied with the level of security on your mobile phone?* Both of them give an affirmative answer.

(d) *Would you ever consider adding more security features to your mobile phone?* Surprisingly the two participants are somewhat open to this question. We could infer that they do not install any security out of laziness. Or, they are confident in protecting their private data from leaking when using mobile phones.

(e) *At this point during the interview we have both participants install and configure PEK.*

(f) *Would you recommend this application to a friend?* Participant A is glad to recommend it to friends who are concerned with security since they often show up in public. Participant B thinks this application is a good recommendation to those who need more security.

(g) *Do either of you have any suggestions about improving the application?* Participant B shows little interest in PEK. He says that "it can be used, but I will not use it." One suggestion from Participant A is to get rid of the large popup of a key when hitting a key. He finds it really annoying that the large version of the letter covers the whole screen, leaving little space for other letters.

3.3. Improvements in PEK 3.x. We have noticed in the pilot usability test that it is the configuration process that takes participants long time, during which they fail to find the PEK application icon on the smart phones. We add an icon of PEK to the Android home screen as shown in Figure 9 so that a user can tap it and finish configuration as shown in Figure 10. To set PEK as a keyboard, a user can click the "Open Android Input Settings."

Moreover, many participants think it is inconvenient to use PEK in specific circumstances since PEK cannot be learned. So, we take their suggestion to create a new button enabling them to turn on/off the randomization of PEK. As is shown in Figure 11, we implement a random toggle button on the keyboard in order that users can choose between a regular keyboard and a randomized keyboard according to their own wishes.

4. Second Usability Testing

In 2017 summer, a second two-stage usability test was conducted by another researcher, who performed interviews and surveys. The format is similar to the format of the first usability test. The first test is an interview-based pilot usability test that is done to pinpoint issues. Data collected from the pilot test is used to help form a web survey. The second test, that is, the survey-based main usability testing, is conducted after PEK is improved based upon the pilot study.

4.1. Participants

Pilot Usability Test. There are 12 participants, 6 males and 4 females, for the phone based interview. Ages range from 17 to 54. 50% of the participants are iOS users, 30% are Android users, and 20% are both iOS and Android users. For this test a Samsung S8 is provided by the interviewer for them to

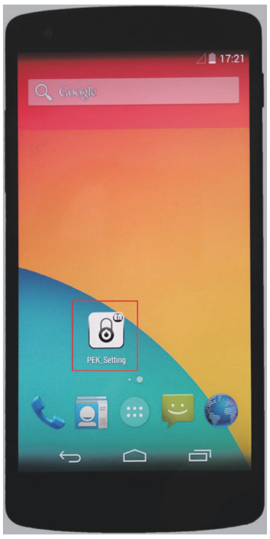
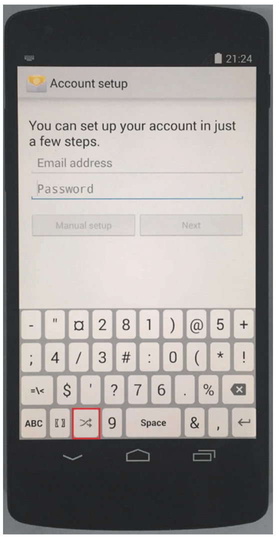


Figure 9: Home screen app.



21:24

Account setup

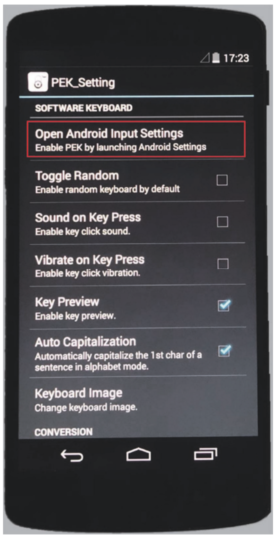
You can set up your account in just a few steps.

Email address

Password

Manual setup Next

Figure 11: Toggle button.



17:23

PEK_Setting

SOFTWARE KEYBOARD

Open Android Input Settings
Enable PEK by launching Android Settings

Toggle Random ☐
Enable random keyboard by default

Sound on Key Press ☐
Enable key click sound.

Vibrate on Key Press ☐
Enable key click vibration.

Key Preview ☑
Enable key preview.

Auto Capitalization ☑
Automatically capitalize the 1st char of a sentence in alphabet mode.

Keyboard Image
Change keyboard image.

CONVERSION

Figure 10: PEK setting.

complete the task. For the clipboard based interview of this test, both participants are female and above the age of 50. One of them is an Android/Apple user, and the other is a basic cellphone user. The clipboard provides written instructions on the installation and configuration of PEK.

Main Usability Test. The main usability test had 281 participants. There are 163 male and 118 female participants. Participants range from 18 to 65 years old and are from various backgrounds. Figure 13 shows the age distribution. All of them are Android users because it is a requirement for the web survey, also because the PEK is only currently available for the Android platform.

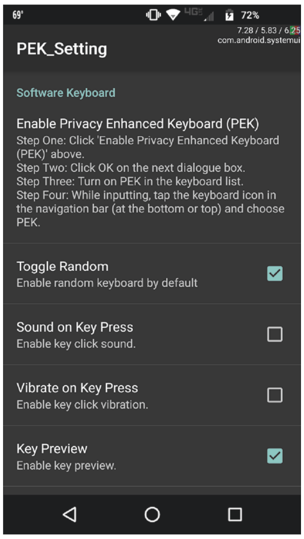
4.2. Pilot Usability Test. The pilot test had all of the participants interviewed in person. The interview task was to install and configure PEK on the Samsung S8, with minimal help from the interviewer. Participants were encouraged to think aloud and ask any questions if needed. The goal of the interview was to find any common problems that arose when participants were using the PEK. Halfway through the study, there was a realization that some of the participants were not familiar with the Android operating system or smart phone operating system in general. To compensate for this lack of familiarity, there was a step-by-step print-out of the whole installation and configuration process of the PEK (screen by screen). The print-out is called clipboard for participants that did not want to or did not know how to use the Samsung S8. In this interview, via the clipboard, participants were asked what steps they would take to download and configure the PEK successfully. There were only two participants for this type of interview. If the participant answered correctly, they were allowed to proceed to the following page. The participants were also encouraged to think aloud and ask questions like the ones in the S8 interview. However, if they could not get to a certain point without asking too many questions, the clipboard was taken away, and the test was followed by the interviewer asking for feedback on their experience of the PEK itself.

Four major issues in the pilot test are addressed and the installation as well as configuration time for the updated PEK is evaluated.

(i) *PQ1$'$*: have you heard of the PEK application? As shown in Figure 14, most of the participants never heard of the PEK so an explanation is needed.

(ii) *PQ2$'$*: did you view the visuals on the Google Play Store? As shown in Figure 15, the belief of “not being able to configure the app” was drawn from the participants not paying attention to the visuals.

(iii) *PQ3$'$*: on a scale of 1 to 5, how comfortable are you with operating your device? Figure 16 illustrates the distribution of the answers of the comfortability with users’ own device. If participants are not comfortable



PEK_Setting

Software Keyboard

Enable Privacy Enhanced Keyboard (PEK)
Step One: Click 'Enable Privacy Enhanced Keyboard (PEK)' above.
Step Two: Click OK on the next dialogue box.
Step Three: Turn on PEK in the keyboard list.
Step Four: While inputting, tap the keyboard icon in the navigation bar (at the bottom or top) and choose PEK.

Toggle Random ☑
Enable random keyboard by default

Sound on Key Press □
Enable key click sound.

Vibrate on Key Press □
Enable key click vibration.

Key Preview ☑
Enable key preview.

FIGURE 12: PEK configuration app interface.

or familiar with operating their own device, this could also be a reason why they could not set up the PEK.

(iv) *PQ4′*: do you have security on your phone, such as a pin or password? As depicted in Figure 17, if participants are password or pin users, they can be key candidates to utilize the improved PEK.

(v) *Installation and configuration times*: as seen in Figure 18, on average it takes everyone interviewed 22 seconds to install the app and 118 seconds to configure the keyboard. Overall, it takes participants approximately 5 times longer to set up the keyboard compared to their installation time.

4.3. Main Usability Test. This test is formulated after common issues are discovered by the participants in the pilot test. The issues are fixed, and then a survey for only Android users is published. Improvements to the PEK are as follows.

(i) Fixing program bugs. Apparently nobody wants to use an app that crashes all the time.

(ii) Enhancing and adding to settings (on-screen instructions for configuring the PEK). As shown in Figure 12, we add the on-screen instructions in the configuration app and instruct the users how to configure and use PEK.

The web survey is hosted by Amazon Mechanical Turk. This survey allows the participants to install and configure the PEK alone, while leaving feedback. Each participant is allotted 40 minutes to complete the survey. Each participant is also compensated for their genuine and honest feedback. Newly formulated questions for the web survey are as follows.

(i) Do you know how to use your smartphone? If participants do not feel comfortable with operating their smartphone, that can be part of the issue as to why they could not configure the app.

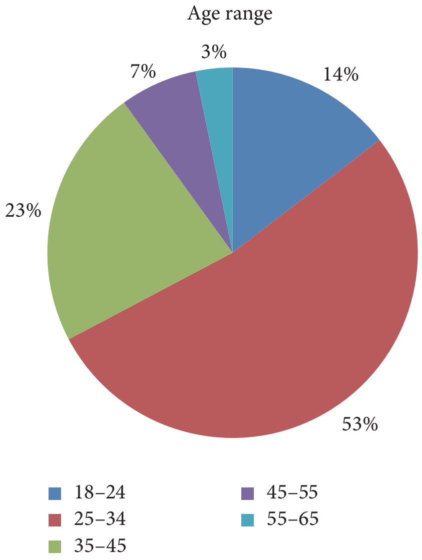


FIGURE 13: Distribution of participant ages.

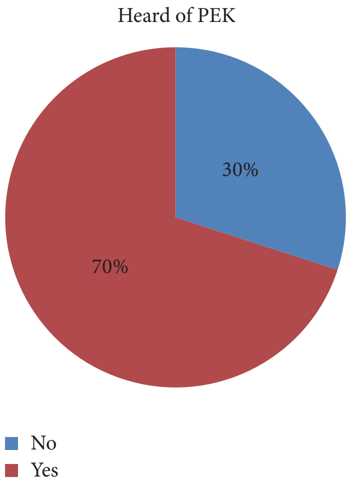


FIGURE 14: Distribution of answers to PQ1′.

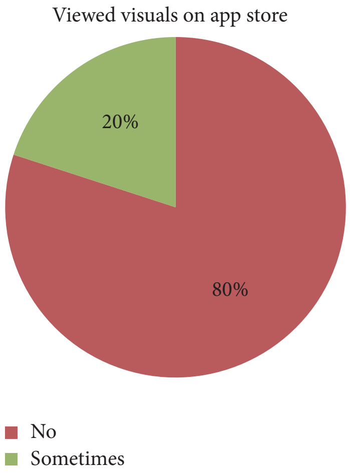


FIGURE 15: Distribution of answers to PQ2′.

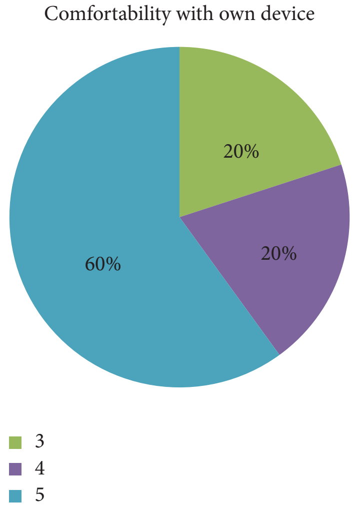


Figure 16: Distribution of answers to PQ3′.

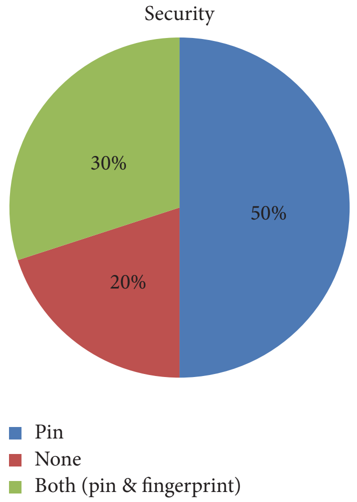


Figure 17: Distribution of answers to PQ4′.

(ii) How often do you enter a password or pin on your phone a day? If the participants enter their passwords daily at a high frequency, the PEK will be a perfect fit for them.

(iii) Did you follow the on-screen instructions after you installed the app to help configure the PEK? With the new update, the user would be forced to view the instructions on how to set up the keyboard. This is better than the visuals on the app store because users are now obligated to look at it. This is different from the app store previews because users are not forced to view the visuals to install the app.

The web survey is broken down into two parts. The first quarter of the survey was strictly demographic questions and the rest of the survey is about the users' experience with the PEK. In this test, the following ten major issues are addressed.

(i) *MQ1′*: do you understand how to use your smartphone? As shown in Figure 19, 58% thoroughly understood, 34% mostly understood, 8% somewhat understood, and less than 1% either somewhat or mostly did not understand. Because of these findings being very similar to the comfortability question in the pilot test, it is clear that issues with the PEK had nothing to do with the users' understanding of their own device.

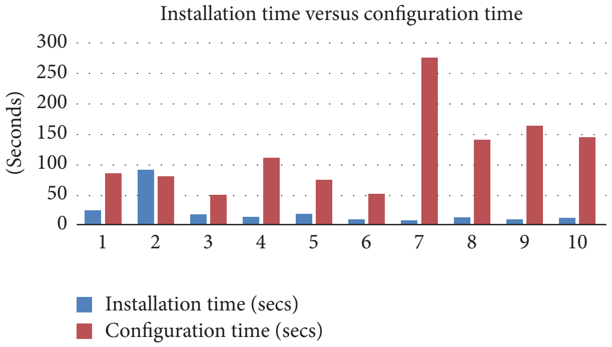


Figure 18: Installation and configuration time.

(ii) *MQ2′*: on a scale of 1 to 10, how would you rate the ease of installing the PEK app? (1 being extremely hard, 10 being extremely easy.) As can be seen in Figure 20, 49% of the participants rate the ease of installing the PEK 8 or higher. Just like the results of the pilot test, the installation is relatively easy.

(iii) *MQ3′*: on a scale of 1 to 10, how would you rate the ease of setting up the PEK (before actually using it)? (again 1 being extremely hard, and 10 being extremely easy). As depicted in Figure 21, 56% of the participants feel that the configuration process is relatively good. The comments for the ratings being an 8 or higher include "no problems at all" or "nothing." Some of the lower rated comments about the configuration complain that there is "too much/too little information" or would like that it could "show more pictures."

(iv) *MQ4′*: did you use the on-screen instructions to set up the keyboard? Suggestions to have on-screen instructions from the pilot test took on a liking in the main usability test. As seen in Figure 22, close to 90% utilized the on-screen help for configuring the app.

(v) *MQ5′*: were the instructions helpful? This question is displayed if "yes" is selected to *MQ4′*. As shown in Figure 23, 99% of the participants who use the instructions think they are either helpful or somewhat helpful. Only 3 participants, who belong to the 1%, do not think they are. One of them says "I am still unable to understand how to use this. There should be a tutorial or user guide for the same or help tool," and the others left no feedback.

(vi) *MQ6′*: were you able to configure the keyboard without any problems? This question is displayed if "no" is selected for *MQ4′*. As shown in Figure 24, only 65% are able to successfully accomplish the setup without the instructions.

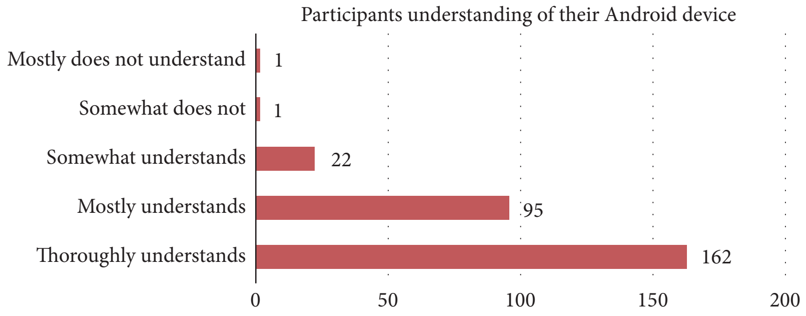


FIGURE 19: Distribution of answers to MQ1′.

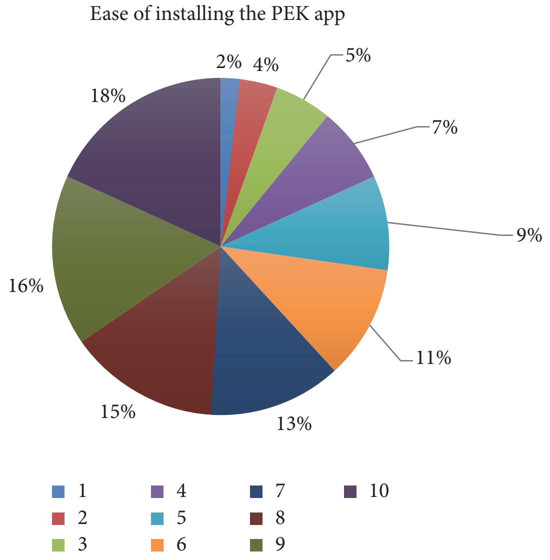


FIGURE 20: Distribution of answers to MQ2′.

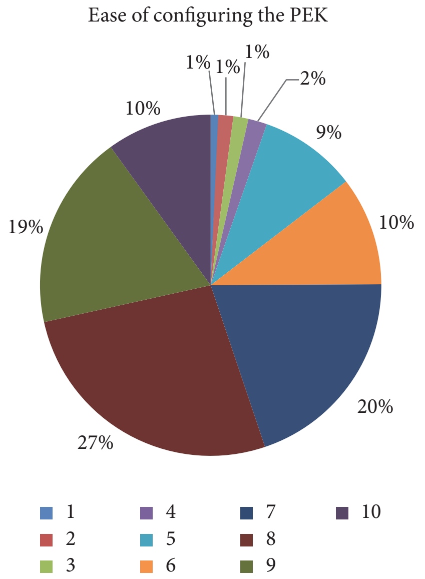


FIGURE 21: Distribution of answers to MQ3′.

(vii) *MQ7′*: did you go back to follow the instructions for help or attempt to solve them yourself? This question is displayed if "yes" is not selected for *MQ6′*. As depicted in Figure 25, 62% are able to set up the PEK on their own successfully, while the remaining 38% have to turn back to the instructions. The main issue for the ones who have to return to the instructions is locating the keyboard icon to switch keyboards outside of the settings.

(viii) *MQ8′*: the PEK is useful. As depicted in Figure 26, 88% of the participants fall within the agree range. Some of their comments also include "[liked] the idea of PEK [and] will definitely use it," "nothing was confusing," and "effective keyboard." For the participants that fall into the 12%, their responses include "could not get PEK enable[ed]" and "[the PEK] barely gives any predictions correctly."

(ix) *MQ9′*: would you recommend the PEK to anyone? As seen in Figure 27, 67% of the participants are either willing or definitely would recommend the PEK to others. However, the remaining 33% are not guaranteed or will not at all. This is a motivation to improve the app even more.

(x) *MQ10′*: would you continue using the PEK after this survey? As seen in Figure 28, almost half of the participants would continue using the app after the survey. Reasons why others would either maybe or not use it include "difficulty using the keyboard with other languages," "Google Play instructions were not [effective]," and "does not like the idea of the app collecting your passwords" while we explicitly note PEK does not collect any passwords.

4.4. Summary. In summation, the pilot and main usability test results are extremely valuable. The pilot test allows the main issue of configuring the keyboard to be found. All the iPhone, Android, and basic cellphone users are allowed to participate in the pilot test because we want to see if there is a common thought process that is reoccurring across our participants. Surely, all participants share the same thought that the PEK will automatically be enabled after they turn it on in the language and input settings. This makes them a bit frustrated and lowers their motivation to continue using

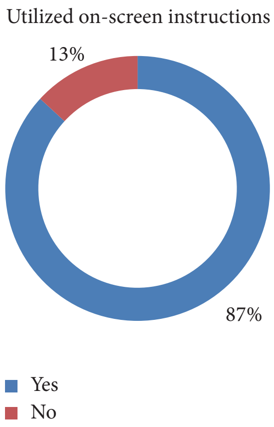


Figure 22: Distribution of answers to MQ4′.

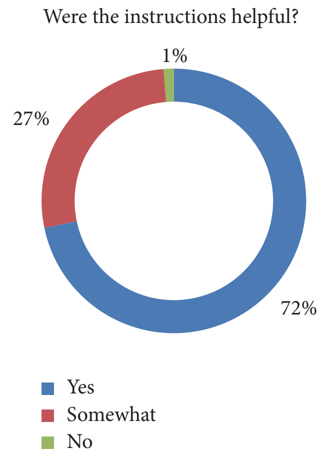


Figure 23: Distribution of answers to MQ5′.

the app. The main usability test narrows down our audience strictly to Android users. Since the app is only currently available on the Google Play Store, we want to test the updated app only on the users that are familiar with the phone's system. The main improvement of the updated app that would directly affect consumers is the added on-screen configuration instructions. While there are other bug fixes and code improvements, this fix would directly be associated with our pilot test participants' configuration problem. Only 40% of our pilot phone based interviewees say that they would/might use the app in the future. That number drastically increases with the added instruction component to 88% in the web survey. The majority of personal responses on their interaction with the PEK claim to have no issues configuring the app and think it is easy. However, because all responses do not claim this, there is still room for improvement. Some of the critiques from the web survey suggested we have a more interactive instruction for configuring the keyboard. Ideas of having a showcase view for the PEK setup have been mentioned to attend to this request. Some other thoughts within the design team are to make the keyboard available in multiple languages to diversify the audience and increase future downloads.

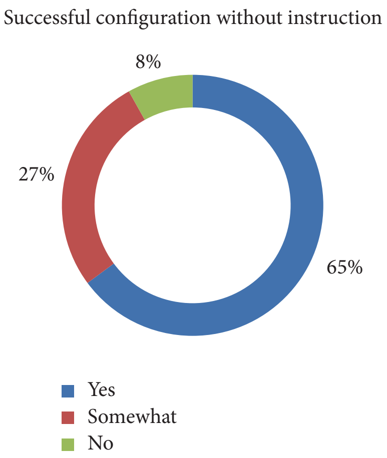


Figure 24: Distribution of answers to MQ6′.

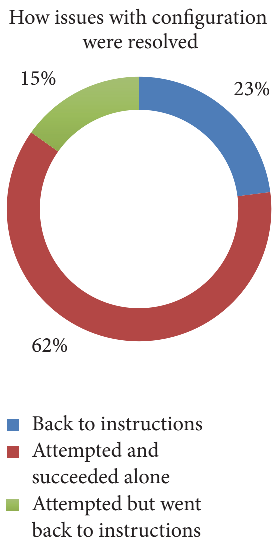


Figure 25: Distribution of answers to MQ7′.

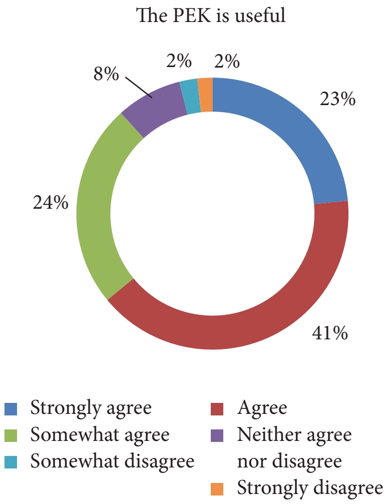


Figure 26: Distribution of answers to MQ8′.

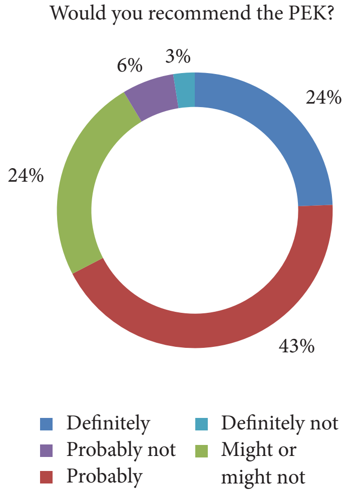


Figure 27: Distribution of answers to MQ9′.

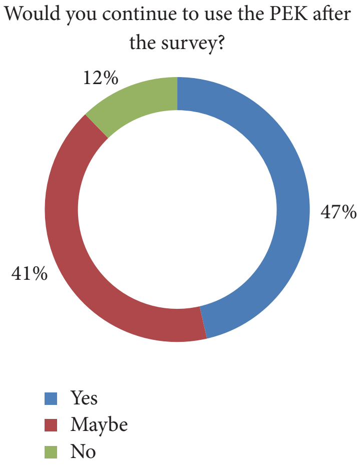


Figure 28: Distribution of answers to MQ10′.

5. Related Work

Various side channel attacks against mobile devices aim to infer a victim's sensitive information, for example, passwords, entered on the soft keyboard of the touch-enabled screen. They can be classified into two categories: internal and external side channel attacks. In internal side channel attacks, it is assumed that an attacker is able to install a malware in a victim's device and exploit diverse sensor data *inside a device*, for example, front camera and microphone [20], accelerometer [15–21], and ambient light sensor [25]. In external side channel attacks, an attacker can exploit side channels *outside a device*. Three example external side channel attacks are residue-based attacks [1–5], Wi-Fi-based attacks [26], and vision-based attacks [6–12, 27, 28].

Intensive research efforts have been made to mitigate these side channel attacks in the past decades. For example, Hirsch [29, 30] invented a secure keypad input terminal to randomly display the ten numerical digits 0 through 9. McIntyre et al. [31] proposed a random PIN pad to display a random numerical keypad layout; however, for usage purpose, it still preserved the numerical order in the horizontal or vertical direction. Moreover, they adopted a regular hexagon background pattern for each key which significantly increases the number of possible key locations. Hoanca and Mock [32] investigated the arrangements for sixteen characters on a 4 × 4 screen to randomize the distribution in the vertical, horizontal, spiraling, diagonally, and other directions while preserving the lexicographic order. Shin [22] first generated a 10-button random keypad by randomly arranging the numbers and letters together. The user should remember the mapping relationship between the letters and numbers. Then a randomized letter keypad is displayed so that the user can recall the letters corresponding to the numbers of her password and input the password. Lee [33] proposed a method to randomly display ten numerical digits in arrays, matrix, a wheel format or with different key background colors, background patterns, shapes, and fonts. Kim [34] presented a scheme to first select 5 random numbers out of 10 and displayed them in a 12-button keypad layout. Then by pressing a "next" button, the remaining 5 numbers can be randomly displayed in the keypad. In comparison, our randomized keyboard can randomly arrange the 26-letter keyboard and automatically identify the type of the input box. Therefore, our privacy enhancing keyboard can provide both privacy protection and usability.

Randomized keyboards are often applied in online banking apps. However, they are application-level randomized keyboards that can only be used in a particular application. The PEK is a system-level Android keyboard that can be used for any application including screen lock, email, and banking. Moreover, it can sense the property of the input box to pop up an appropriate keyboard so as to improve the user experience. More importantly, we are the first to design a generic randomized keyboard for Android.

6. Conclusion

This paper presents a full-scale usability testing of a generic Android privacy enhancing keyboard (PEK), which can prevent various attacks against touch-enabled devices from inferring user pins or passwords. We perform an iterative two-round two-stage usability test including pilot usability tests and main usability tests for improving PEK for broad adoption. Based on the findings of the two usability tests in the first usability test, we implement new features in the current PEK. After the iterative improvement efforts, most users find our app easy to use and install. However, the usability test demonstrates the worrisome phenomena that many users blindly trust their phones for security or are not much concerned with the possible breaches. These phenomena demonstrate the human factor that contributes to the vulnerabilities of the cyber space.

Disclosure

Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

Conflicts of Interest

There are no conflicts of interest in the manuscript.

Acknowledgments

This work was supported in part by National Key R&D Program of China under Grant 2017YFB1003000, by National Natural Science Foundation of China under Grants 61502100, 61532013, 61402104, 61572130, 61602111, 61632008, and 61320106007, by US NSF Grants 1461060, 1642124, 1547428, and CNS 1350145, by University System of Maryland Fund, by Ant Financial Research Fund, by Jiangsu Provincial Natural Science Foundation of China under Grants BK20150637 and BK20140648, by Jiangsu Provincial Key Technology R&D Program under Grant BE2014603, by Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201, by Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grant 93K-9, and by Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] M. Zalewski, "Cracking safes with thermal imaging," 2005, http://lcamtuf.coredump.cx/tsafe/.

[2] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the Workshop on Offensive Technology WOOT*, 2010.

[3] K. Mowery, S. Meiklejohn, and S. Savage, "Heat of the moment: characterizing the efficacy of thermal camera-based attacks," in *Proceedings of the Workshop on Offensive Technologies (WOOT)*, 2011.

[4] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu, "Fingerprint attack against touch-enabled devices," in *Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, pp. 57–68, Raleigh, NC, USA, October 2012.

[5] Y. Abdelrahman, M. Khamis, S. Schneegass, and F. Alt, "Stay cool! understanding thermal attacks on mobile-based user authentication," in *Proceedings of 35th Annual CHI Conference on Human Factors in Computing Systems (CHI)*, pp. 3751–3763, Denver, CO, USA, May 2017.

[6] M. Backes, M. Duermuth, and D. Unruh, "Compromising reflections - or - how to read lcd monitors around the corner," in *Proceedings of the 29th IEEE Symposium on Security and Privacy (S&P)*, 2008.

[7] M. Backes, T. Chen, M. D1rmuth, H. P. A. Lensch, and M. Welk, "Tempest in a teapot: Compromising reflections revisited," in *Proceedings of the 30th IEEE Symposium on Security and Privacy (S&P)*, 2009.

[8] D. Balzarotti, M. Cova, and G. Vigna, "Clearshot: eavesdropping on keyboard input from video," in *Proceedings of the 29th IEEE Symposium on Security and Privacy (S&P)*, 2008.

[9] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, "A fast eavesdropping attack against touchscreens," in *Proceedings of the 2011 7th International Conference on Information Assurance and Security, IAS 2011*, pp. 320–325, Melaka, Malaysia, December 2011.

[10] Q. Yue, Z. Ling, X. Fu, B. Liu, W. Yu, and W. Zhao, "My google glass sees your passwords!," in *Proceedings of the Black Hat USA*, 2014.

[11] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *Proceedings of the 21st ACM Conference on Computer and Communications Security, CCS 2014*, pp. 1403–1414, November 2014.

[12] J. Sun, X. Jin, Y. Chen, J. Zhang, R. Zhang, and Y. Zhang, "Visible: Video-assisted keystroke inference from tablet backside motion," in *Proceedings of the 23rd ISOC Network and Distributed System Security Symposium (NDSS)*, 2016.

[13] L. Zhang, Z. Cai, and X. Wang, "FakeMask: A Novel Privacy Preserving Approach for Smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[14] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[15] L. Cai and H. Chen, "TouchLogger: Inferring keystrokes on touch screen from smartphone motion," in *Proceedings of the 6th USENIX Workshop on Hot Topics in Security (HotSec)*, 2011.

[16] Z. Xu, K. Bai, and S. Zhu, "TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 113–124, Tucson, Ariz, USA, April 2012.

[17] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: password inference using accelerometers on smartphones," in *Proceedings of the Proceeding of the 13th Workshop on Mobile Computing Systems and Applications (HotMobile '12)*, no. 9, New York, NY, USA, February 2012.

[18] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: your finger taps have fingerprints," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pp. 323–336, Ambleside, UK, June 2012.

[19] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12)*, pp. 41–50, ACM, Orlando, Fla, USA, December 2012.

[20] L. Simon and R. Anderson, "Pin skimmer: Inferring pins through the camera and microphone," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, 2013.

[21] D. Ping, X. Sun, and B. Mao, "Textlogger: inferring longer inputs on touch screen using motion sensors," in *Proceedings of the 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2015.

[22] H.-S. Shin, "Device and method for inputting password using random keypad," United States Patent No. 7,698,563, 2010.

[23] OMRON SOFTWARE Co., Ltd. Openwnn, 2012. https://sourceforge.net/u/lluct/me722-cm/ci/890e9a90d9a7fe5f0243b9392ea-a787d1381e987/tree/packages/inputmethods/OpenWnn/.

[24] Z. Ling, M. Borgeest, C. Sano et al., "A case study of usable security: Usability testing of android privacy enhancing keyboard," in *Proceedings of the 12th International Conference on Wireless Algorithms, Systems, and Applications (WASA)*, 2017.

[25] R. Spreitzer, "Pin skimming: Exploiting the ambient-light sensor in mobile devices," in *Proceedings of the 4th ACM Workshop*

on Security and Privacy in Smartphones and Mobile Devices (SPSM), 2014.

[26] M. Li, Y. Meng, J. Liu et al., "When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security, CCS 2016*, pp. 1068–1079, October 2016.

[27] Z. Li, Q. Yue, C. Sano, W. Yu, and X. Fu, "3D vision attack against authentication," in *Proceedings of the ICC 2017 - 2017 IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017.

[28] K. Jin, S. Fang, C. Peng et al., "Vivisnoop: Someone is snooping your typing without seeing it!," in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2017.

[29] S. B. Hirsch, "Secure keyboard input terminal," United States Patent No. 4,333,090, 1982.

[30] S. B. Hirsch, "Secure input system," United States Patent No. 4,479,112, 1982.

[31] K. E. McIntyre, J. F. Sheets, D. A. J. Gougeon, C. W. Watson, K. P. Morlang, and D. Faoro, "Method for secure pin entry on touch screen display," United States Patent No. 6,549,194, 2003.

[32] B. Hoanca and K. Mock, "Screen oriented technique for reducing the incidence of shoulder surfing," in *Proceedings of the 2005 International Conference on Security and Management, SAM'05*, pp. 334–340, June 2005.

[33] C. Lee, "System and method for secure data entry," United States Patent Application Publication, 2011.

[34] I. Kim, "Keypad against brute force attacks on smartphones," *IET Information Security*, vol. 6, no. 2, pp. 71–76, 2012.

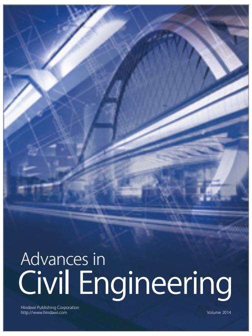
Journal of
Engineering

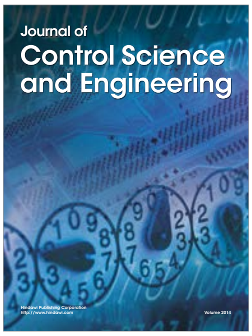
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

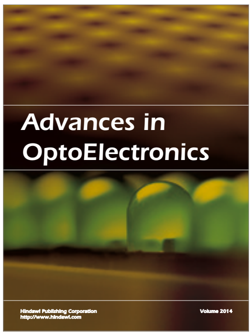
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

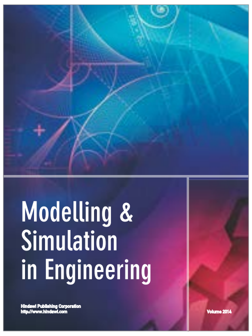
Journal of
Robotics

Journal of
Electrical and Computer Engineering

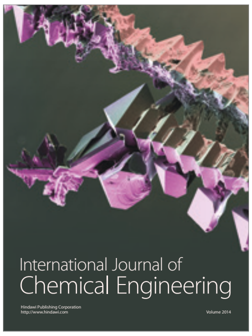

Advances in
OptoElectronics

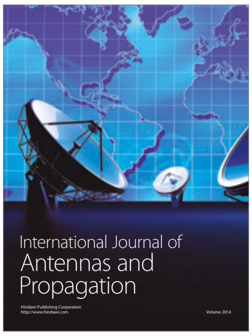
VLSI Design

International Journal of
Navigation and Observation

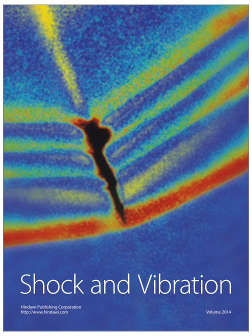

Modelling &
Simulation
in Engineering

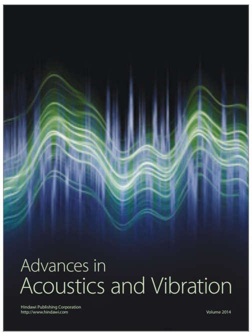
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration