RandomPad: Usability of Randomized Mobile Keypads for Defeating Inference Attacks

Anindya Maiti[†], Kirsten Crager[†], Murtuza Jadliwala[†], Jibo He[†], Kevin Kwiat[◇] *and* Charles Kamhoua[◇]
[†]Wichita State University, Wichita, KS, USA
[◇]Air Force Research Laboratory, Rome, NY, USA
Email: a.maiti@ieee.org, krcrager@shockers.wichita.edu, murtuza.jadliwala@wichita.edu, jibo.he@wichita.edu,
kevin.kwiat@us.af.mil, charles.kamhoua.1@us.af.mil

***Abstract*—The feasibility of malicious keystroke inference attacks on mobile device keypads has been demonstrated by multiple recent research efforts, but very little has been accomplished in the direction of protection against such attacks. One common assumption in these attacks is that the adversary has knowledge of the size and layout of the keypad employed by the target user, which is reasonable as keypad layouts and sizes are generally standard. Thus, an effective protection strategy against such keystroke inference attacks would be to randomly change the layout of the target keypad. However, before proposing unconventional changes to the widely used and highly familiar default keypads, a comprehensive usability evaluation is required. This paper accomplishes this goal by comprehensively studying the usability of randomized keypads that employ varying degrees of randomization in terms of key size, sequence and position. The privacy-usability trade-off of different randomized keypad strategies is then analyzed by empirically comparing their ease-of-usage and security assurance.**

I. Introduction

Users have been increasingly using their mobile devices and smartphones to enter personal and private information, such as, PIN, credit card numbers, passwords and telephone numbers. However, touchscreen-based numeric keypads on these mobile devices and smartphones are becoming increasingly more vulnerable to *side-channel keystroke inference attacks*, which results in a serious invasion of privacy of mobile users. Kune et al. [8] leveraged on a common assumption that an audio feedback to the user is imparted for each button pressed, and demonstrated the possibility of inferring keystroke sequences based on time delays between keystrokes. Yue et al. [32] used computer vision to analyze the shadow formation around the fingertip to automatically locate the touched points. Simon et al. [24] used microphone to detect touch events, while the camera is used to estimate the smartphone's orientation, and correlate it to the position of the digit tapped by the user. Sun et al. [29] used video recordings of the backside of a tablet to infer typed keystrokes, based on the observation that keystrokes on different positions of the tablet's soft keyboard cause its backside to exhibit different motion patterns. Zhang et al. [33] analyzed finger smudges left on the touch screen surface to infer touch patterns, with remarkable success.

Motion sensors, such as, accelerometer and gyroscope, represent another class of side-channels for accomplishing keystroke inference attacks that have been highly researched. Tapping at different locations on a touchscreen results in unique movements of the mobile device which can be captured by eavesdropping on-board motion sensors. Cai et al. [4] were one among the first to use this observation to train multi-class classifiers for each of the ten spatially separated numbers of a keypad, and were able to correctly predict up to 70% of test keystrokes. Owusu et al. [18] extended the side-channel attack from numeric keypads to soft QWERTY keyboards. Maiti et al. [17] used a smartwatch to demonstrate that an external device's motion sensors can also be used to infer keystrokes made on a mobile keypad. Lack of any access control to motion sensors on existing mobile (and wearable) operating systems further improves the feasibility of such motion side-channel based inference attacks.

Interestingly, all the above attacks share one common assumption: *the numeric keypad employed by the target user has a standardized key layout (Figure 1) known to the adversary.* Intuitively, this means that if the keypad layout is changed from the standardized layout unbeknownst to the adversary then the above attacks will perform poorly. Thus, such a dynamic keypad layout strategy is an appealing defense strategy against side-channel keystroke inference attacks. However, as an adversary can also re-train the attack framework for the new keypad layout, changing the keypad layout just once (or in a predictable fashion) will not be an effective defense. In order to prevent an adversary from knowing the keypad layout in use at any given time, this change in layout should be *randomized.* In Section IV, we present different keypad randomization strategies, in terms of key size, sequence and location. The primary goal of these strategies is to reposition the on-screen keys such that an adversary cannot correctly predetermine the keypad layout in use at any given time. Without an accurately predetermined keypad layout, the adversary will be unable to train or set up the attack framework, and an improperly trained attack framework will result in erroneous inference of keystrokes. Interestingly, a major smartphone manufacturer recently introduced a custom authentication method called "Random PIN entry" [16], which implements a randomization strategy in order to restrict side-channel attacks.

While randomized keypads could provide an effective defense against keystroke inference attacks, it also raises us-

[†]This research was partially supported by the National Science Foundation (NSF) under award number CNS - 1523960.

[◇]Approved for Public Release; Distribution Unlimited: 88ABW-2016-5162, Dated 17 Oct 2016.

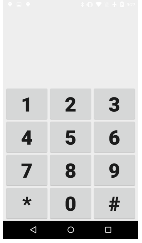


Fig. 1: Default keypad.

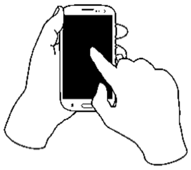


Fig. 2: A common typing scenario.

ability concerns. The default keypad (Figure 1) is widely used and many users have gradually become habituated to the static layout. Thus, randomizing the keypad brings two new challenges: (a) users may be uncomfortable typing on a keypad different from the one they are habituated to, and (b) as the keypad changes randomly, users will always face an unfamiliar keypad. If users are discomforted to a level where they may opt not to use random keypads for the sake of privacy, then it cannot be proposed as an effective defense mechanism against inference attacks. Therefore, before recommending the use of randomized keypads for privacy protection, it is critical to evaluate the usability of the various layout randomization strategies. In order to achieve this goal, we comprehensively assess the *usability* and *perceived workload* of typing on keypads generated by each of the proposed randomization strategies with the help of actual typing experiments involving a diverse set of human subjects. We also compare the *rate of mistyping* among all strategies, and attempt to determine whether one strategy is more usable than the others. In addition to this, we also evaluate if usability is positively influenced by distinguishable visual features, i.e., by coloring each key with an ascending shade of gray. Finally, by comparing their empirical ease-of-usage with their analytical security assurance, we attempt to study the privacy-usability trade-off (if one exists) in using different types of randomized keypads.

II. Attack Description

We consider the scenario of a potential victim typing on a smartphone's *numeric touchscreen keypad* (Figure 2) in the presence of an adversary whose goal is to infer the keystrokes typed by the victim. For *on-device* inference attacks using device sensors as information side-channels, the adversary may bug or eavesdrop on the target's smartphone [4], [18], [24] (and/or paired smartwatch [17]) by installing a malicious application which records the activity of certain on-board sensors. This step can be achieved by exploiting known software vulnerabilities or by tricking the victim into installing a *Trojan* or a malicious code hidden within a useful application. The malicious application also maintains a *covert communication channel* with the adversary, and periodically uploads the eavesdropped data to an adversarial server by means of this channel. For *external* inference attacks [8], [29], [32], [33], the adversary captures relevant keystroke characteristics from a physically close position, using appropriate eavesdropping devices, such as, microphones or wireless transceivers. We assume that the adversary has sufficient storage and computational resources to process the eavesdropped data and successfully carry out both types of attacks. However, ***we assume that the adversary cannot visually eavesdrop or observe the keypad (and the victim's keystrokes)*** and does not have the ability to install system level key-logging applications to directly obtain the typed keystrokes.

III. Related Work

A. Protection Against Side-Channel Attacks

Due to the increasing use of various sensors in mobile and wearable devices as information side-channels to accomplish privacy invasive inference attacks, the topic of defending against such attacks has gained prominence. Cai et al. [5] drew attention on the limitations of current mobile systems in mitigating side-channel attacks. They also pointed out the following desirable properties in any defense solution: (i) **Security:** the solution must protect against side-channel inference attacks, (ii) **Usability:** ideally, the solution should require no extra effort from users and if extra effort is unavoidable, it should not disrupt the users' work flow, (iii) **Backward and Forward Compatibility:** the solution should require no or minimal modification to existing applications and operating systems, (iv) **Performance:** the solution should have no or minimal overhead, and (v) **Versatility:** the solution should be deployable on various types of mobile hardware, software, and user interfaces. If a defense solution fails to fulfill any of these properties, it may not be well accepted by users.

Controlling access to sensors that has the potential to be used as side channels is one form of defense mechanism that can be used. However, as mobile and wearable systems currently offer very limited access control options (mostly restricted to location and microphone sensors), fine-grained access control to all sensors will require major modifications to these systems. Context-aware access controls [6] could relieve users from manually changing and adjusting access settings, however they would add significant performance and energy overhead, are non-versatile and difficult to setup and would require major modifications to current operating systems. *Furthermore, sensor access controls do not protect against applications that gain legitimate access to these sensors.* Enforcing system-wide reduced sensor sampling rate or disabling sensors is one suggested defense against on-device keystroke inference attacks [17], [18]. However, while system-wide down-sampled or disabled sensors may provide protection, it may disrupt useful non-malicious applications as well. *Moreover, neither access control, nor limiting sampling rate, protects against external inference attacks.* There has been limited work on protecting smartphone users against external side-channel attacks. Shrestha et al. [23] proposed the injection of noise in motion sensor readings, in order to protect against motion sensor based inference attacks. However, their solution is ineffective against most other classes of keystroke inference attacks. Alternate forms of authentication (e.g., biometrics) are also becoming popular, but the vast majority of mobile devices are not equipped with the enabling hardware and/or software. Thus, mobile users will continue to use touch screen-based keypads to enter sensitive information, including authentication data, and there is an increasing need for a keypad protection mechanism that satisfies most of the design criteria identified by Cai et al. [5].

B. Protection by Randomization

Randomizing the keypad prevents an adversary from predetermining the keypad layout, which can serve as an effective defense against both external and on-device attacks. Randomized keypads are already known to be used commercially in electronic door access control systems [26], although with limited flexibility in terms of available set of randomization strategies. Ryu et al. [21] were among the first to study randomized keypads and they observed that their randomized keypad resulted in longer completion times compared to a conventional keypad. However, their study was not geared towards mobile devices, considered only one randomization strategy and did not comprehensively evaluate user workload and other usability parameters except completion times. In this research effort, we propose, implement, and comprehensively evaluate different randomized keypads (or *RandomPad*) for mobile devices. RandomPad does not add significant overhead on system performance as it is essentially a rearranged keypad layout. It can be easily implemented as a third party application on popular mobile operating systems such as Android and iOS, without requiring support from operating system developers. RandomPad can also be versatile, when implemented according to scalable design principles [12]. In this paper we analyze the remaining two design properties outline in Section III-A: *security* and *usability*.

IV. Randomization Strategies

We outline six representative strategies that span the entire spectrum of strategies from purely-random to partially-random keypad layouts, i.e., the latter preserves some characteristics of the default layout. For stronger security, keypad randomization can be performed either at the beginning of every keystroke or at the beginning of each typing session.

A. Key Sequence Randomization

The default keypad follows a sequence of ascending numbers. Key sequence randomization strategies reposition the keys by changing the order of keys, by not following the ascending order. Following are the three key sequence randomization strategies we use in our study, all of which keep the key sizes unchanged:

- **Row Randomization (RR)**: In row randomization (RR), rows from the default keypad are randomly ordered while preserving the order of the numbers within each row. Figure 3(a) shows an example of RR.
- **Column Randomization (CR)** In column randomization (CR), columns from the default keypad are randomly ordered while preserving the order of the numbers within each column. Figure 3(b) shows an example of CR.
- **Individual Key Randomization (IKR)** In individual key randomization (IKR), individual keys are randomly rearranged without maintaining any column or row order. Figure 3(c) shows an example of IKR.

B. Key Size Randomization (KSR)

A key size randomization (KSR) strategy preserves the sequence of numbers on the default keypad. Instead, the randomization factor is incorporated within the size of each key. Changing the key sizes also repositions them from their default locations on screen. In our design of KSR, we use a hidden 7×6 grid layout (scaled to fit the width of the screen), as shown in Figure 3(f). One randomly selected key is enlarged to appear as a 4×4 block on the grid, other keys in the same row as the large key appear as 4×1 blocks, and all other keys appear as 1×2 blocks on the grid. Figure 3(d) shows an example of the KSR strategy. Note that the 7×6 grid layout is not visible to users; only the overlaid keys are visible. As the default sequence is preserved, it may be necessary to randomize key sizes after each key press to prevent relative positioning based attacks.

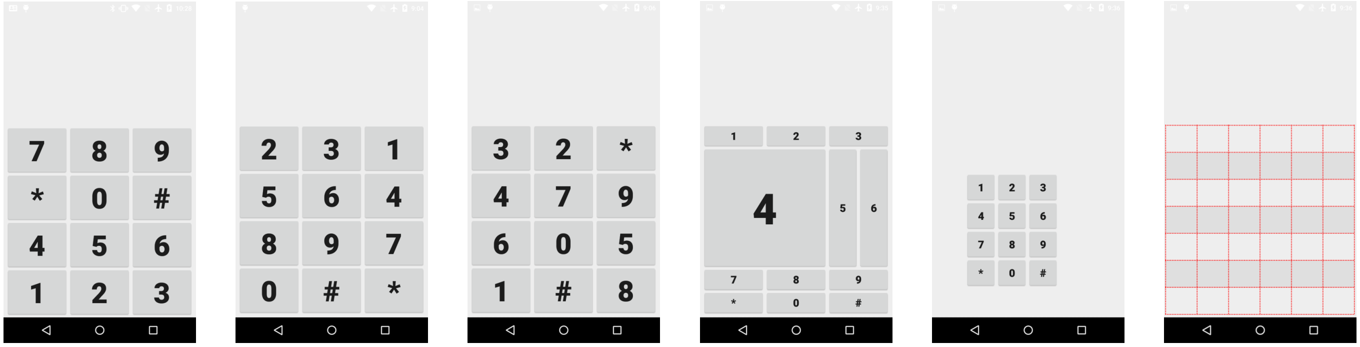
C. Keypad Location Randomization (KLR)

A keypad location randomization (KLR) strategy also preserves the sequence of numbers of the default keypad. The randomization factor is instead incorporated in the location of the keypad, because changing the keypad location repositions all keys from their default locations on the screen. Figures 3(e) shows an instances of the keypad location randomization we consider in our study. In this case, we again use a hidden 7×6 grid layout similar to KSR. Each key appears as a 1×1 block, and the entire keypad appears as a randomly selected contiguous 4×3 block on the grid (16 possibilities). The distribution of keys on the hidden 7×6 grid layout happens to be same for both KSR and KLR (Figure 4). Similar to KSR, keypad randomization in KLR may need to be done at every key press to prevent relative positioning based attacks. Moreover, key sequence randomization strategies can also be combined with KSR and KLR for additional security.

D. Security Analysis

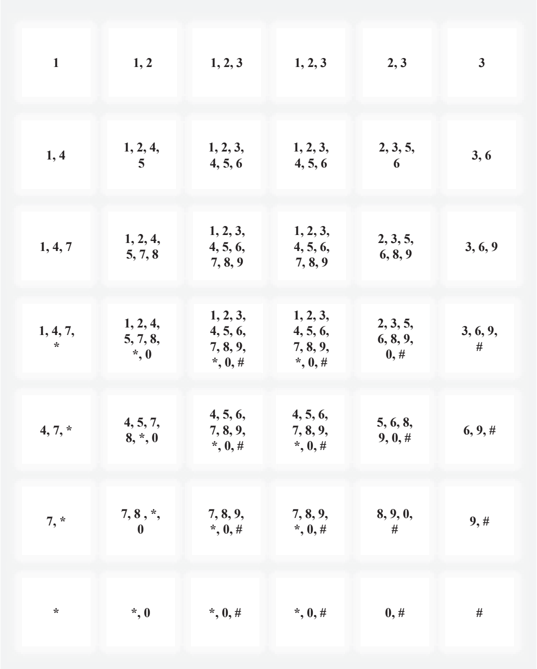
Next, we probabilistically analyze the security offered by these five randomization strategies. For this analysis we assume that the adversary is able to cluster keystroke positions not just on a default sized 4×3 keypad, but also for smaller blocks of a 7×6 layout (used in KSR and KLR), without any error (i.e., 100% accuracy). The adversary is also assumed to know the randomization strategy currently in use and that a new randomized keypad layout within the corresponding strategy is used by the user (victim) for every keystroke (that the adversary is attempting to infer). As the keypad layout is randomized, the best an adversary can do is to guess the mapping between the randomized and default keys. We derive the *successful guessing probability of the adversary* as an indication of the *security assurance* or *guarantee* each strategy provides under such a strong attack scenario. The lower this probability for a particular randomization strategy, the higher is its security assurance.

Consider a twelve key (including "$*$" and "$\#$" keys) keypad in IKR. The probability that an adversary guesses the mapping of a digit correctly is $\frac{1}{12}$ and the probability of correctly guessing the entire mapping (of all the keys) is $\frac{1}{12!}$. However, in case of RR and CR, the adversary can improve its guessing, which is intuitive. Knowing that keys within a row remain in order, for a RR keypad, the adversary only needs to guess the row mapping. The probability that an adversary correctly



(a) (b) (c) (d) (e) (f)

Fig. 3: Examples of (a) RR, (b) CR, (c) IKR, (d) KSR and (e) KLR; (f) The hidden 7 × 6 grid layout used in KSR and KLR.



1	1, 2	1, 2, 3	1, 2, 3	2, 3	3
1, 4	1, 2, 4, 5	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	2, 3, 5, 6	3, 6
1, 4, 7	1, 2, 4, 5, 7, 8	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5, 6, 7, 8, 9	2, 3, 5, 6, 8, 9	3, 6, 9
1, 4, 7, *	1, 2, 4, 5, 7, 8, *, 0	1, 2, 3, 4, 5, 6, 7, 8, 9, *, 0, #	1, 2, 3, 4, 5, 6, 7, 8, 9, *, 0, #	2, 3, 5, 6, 8, 9, 0, #	3, 6, 9, #
4, 7, *	4, 5, 7, 8, *, 0	4, 5, 6, 7, 8, 9, *, 0, #	4, 5, 6, 7, 8, 9, *, 0, #	5, 6, 8, 9, 0, #	6, 9, #
7, *	7, 8 , *, 0	7, 8, 9, *, 0, #	7, 8, 9, *, 0, #	8, 9, 0, #	9, #
*	*, 0	*, 0, #	*, 0, #	0, #	#

Fig. 4: KSR and KLR on-screen key distribution possibilities on the hidden 7 × 6 grid layout.

guesses a row (and thus keys in it) is $\frac{1}{4}$, and all four rows is $\frac{1}{4!}$. Similarly for CR, the probability that the adversary correctly guesses a column (and thus keys in it) is $\frac{1}{3}$, and all three columns is $\frac{1}{3!}$. Due to crossover regions on the keypad, accurately guessing key mappings in KSR and KLR is a bit more complicated. In case of KSR, one large key displaces the position of other keys, leading to non-zero probabilities of multiple keys being at the same on-screen location. As the key distribution possibilities are same for KLR, it also shares the same probability distribution as KSR. Assuming that a keystroke touch can occur uniformly on any of the 42 (7 × 6) on-screen blocks, the probability of an adversary correctly guessing a key's position is:

$$\frac{1}{42}\sum\frac{1}{N_{i,j}}, \forall i, j \tag{1}$$

where, $N_{i,j}$ is the number of keys that could possibly occupy block i, j. Solving Equation 1 using the distribution of keys (Figure 4) results in a success probability of 0.34193. Guessing the entire keypad in KSR and KLR is relatively uncomplicated, as the adversary has to guess only the large key in KSR (probability $\frac{1}{12}$) and one among the sixteen possible locations of the keypad in KLR (probability $\frac{1}{16}$). Table I in ranks the adversary's success probabilities for the different randomization strategies. *These probabilities represent a best-case scenario for the adversary.*

TABLE I: Security assurance of the five proposed randomization strategies. Lower rank is better security.

Randomization Strategy	Correct Entire Keypad Guessing Probability	Security Assurance Rank
CR	$\frac{1}{3!} = 0.16667$	5
IKR	$\frac{1}{12!} = 2.08 \times 10^{-9}$	1
KLR	$\frac{1}{16} = 0.0625$	3
KSR	$\frac{1}{12} = 0.08333$	4
RR	$\frac{1}{4!} = 0.04167$	2

V. Human Factors

The above security analysis shows that randomizing keypad layouts is an effective protection strategy against side-channel keystroke inference attacks. It is also efficient from a system performance perspective, easy to implement and versatile. However, a significant concern remains to be answered: "*Will users employ and effectively be able to use such a protection mechanism*"? As the proposed protection mechanism is simply a different and highly dynamic user-interface, we attempt to answer this broad question by using principles and techniques from the area of human-computer interaction (HCI) [7] and cognitive psychology [15]. Designing usable input interfaces, e.g., keypads, for mobile devices has been a significant technical challenge [11]. For mobile devices, the main constraint in designing usable input interfaces is the screen size, however earlier research has shown that smaller keypad sizes do not negatively affect the efficiency or accuracy of user input [22]. Thus, in this work we will focus only on how random positions and sizes of the keys on the keypad impact their usability. In our quest for answering the above usability question, we will primarily focus on measuring user effort and workload while using these randomized input interfaces (or RandomPad) by means of well-known quantitative and qualitative metrics, as discussed below. We would also like to investigate if certain design changes would improve or reduce user workload.

Keypads with randomized key sequences (e.g., RR, CR, IKR keypads) pose a unique challenge to human cognition. Users may often find themselves searching for a particular key, which would slow down overall typing speed. Physiological factors, such as, visual acuity, light accommodation, dexterity, working memory, and reaction times [9], [28] can further impact this. Thus, time required for the typing task completion and the number of errors during the task are some of the metrics that will be used to evaluate user-effort while using RandomPad. Another commonly used HCI technique to empirically measure the user-effort of interfaces, and thus its usability, is *eye-tracking*. Eye-tracking devices can capture fixation duration and number of fixations while the user is interacting with the interface. The average *fixation duration* indicates how long it takes for users to encode the visual information, which is influenced by the readability of the characters, such as, font size, font style, spacing and contrast of background and foreground, etc. [20]. The *number of fixations* to complete a task is correlated with the difficulty to locate the target (within the task). In this work, we also use these metrics (captured by means of an eye-tracking device) to quantify the difficulty of the user in locating the keys on RandomPad.

Besides these, we also analyze the usability of RandomPad by employing user-provided subjective workload and usability measures. For instance, *NASA-TLX* [10] is a well-known scale for subjectively measuring *mental workload*. Mental workload measures the subjective experience of the effort to complete a task [2]. A high mental workload is often detrimental to task performance and can reduce the chances of the product or interface being adopted or used by users. The NASA-TLX is a multidimensional scale to measure the perceived workload, including, the mental, physical and temporal demand, overall performance, frustration level and effort. We employ the NASA-TLX scale in our experiments to capture the mental workload of participants after they have used RandomPad. Similarly, we also measure the overall usability of the RandomPad design by using another subjective scale called the *System Usability Scale (SUS)* [3]. The SUS is a 10-item 5-point scale, which produces a usability score ranging from 0 to 100, with a larger value indicating a more usable interface. We feel that a combination of task completion performance measures, eye fixation measures using eye-tracking, and subjective mental workload and usability measures will provide a converging evidence to illustrate the usability of RandomPad, and thus provide some answers to the broad usability related question posed earlier.

As discussed earlier, certain physiological or environmental factors may impact human cognition of the interface, and thus its usability. Pattison and Stedmon [19] suggested that certain physiological factors impacting interface usage can be combated with a design that has improved illumination and provides certain distinguishing visual cues/feedback to the user. Luminance differences and contrasting shades (e.g., using a gray-scale) have been particularly successful in capturing user attention [1], [27], as well as, in distinguishing objects in medical diagnostic images [13], [30]. This motivated us to adopt a similar approach where our goal is to evaluate whether usability of our RandomPad interface improves when additional visual cues are provided to the users, for instance, by using contrasting shades of gray to represent each of the keys. More specifically, we study an enhancement to RandomPad, where the keys on the randomized keypad are colored with an ascending shade of gray, i.e., shade of key “0” being the lightest (#D8D8D8) and “9” being the darkest (#000000). The luminance of keys between “0” and “9” are increased in uniform steps. Figure 5 shows an exemplary instance of IKR keypad in our ascending gray-scale scheme. Note that keys “*” and “#” are excluded from this particular usability study. As the key sequence is preserved in KSR and KLR, we do not expect any potential benefit from the gray-scale enhancement, and thus, are not evaluated either.

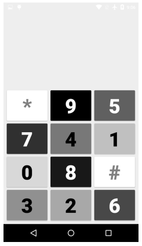


Fig. 5: Aiding usability with contrast: Instance of IKR keypad in ascending gray-scale.

VI. Study

Our study comprised of lab-based experiments involving human subjects, where participants performed typing tasks on a set of assigned smartphone numeric keypads. Each participant is randomly assigned only one of the five keypad randomization strategy, and all experiments administered to each participant are based on the assigned strategy. The assigned strategy is uniformly distributed (balanced) across all participants. The entire experiment for each participant is divided into two sessions: *Natural Typing* and *Dictated Typing*. In each scenario, the participant types using the default, randomized (with the randomly assigned strategy) and gray-scale (assigned only if the participant was assigned RR, CR or IKR) keypads. Our experiment (methodology and data collection process) was reviewed and approved by Wichita State University’s Institutional Review Board.

A. Participants

Our study was conducted by recruiting 100 participants, the majority of whom were affiliated with our university. As an incentive, students were offered participation credits which would partially satisfy certain academic requirements, while non-students were compensated with $10. The participants first completed a pre-survey demographic questionnaire, which included questions on smartphone usage and privacy preferences. Then an introductory video, explaining the concept of randomizing keypads and how it can help protect against certain eavesdropping or side-channel attacks, was shown to them. Participants were then introduced to the specific randomization strategy assigned to them using another video.

However, they were not introduced to the remaining (non-assigned) randomization strategies. This was done to prevent any bias in their response(s). Participant demographic information and preferences are outlined in Table II. Interestingly, when shown a sample random keypad screenshot (according to the assigned randomization strategy), less than 25% were in favor of using the random keypad for typing sensitive information. Note that this response was recorded before they were introduced to the side-channel keystroke inference attacks and how randomization can help protect against it.

TABLE II: Demographics and preferences of participants.

Gender	56% Female 44% Male
Occupation	33% Employed 67% Student
Smartphone Ownership Duration	26% Less than 5 Years 74% More than 5 Years
Current Smartphone	59% iOS (iPhone) 41% Android
Willingness to Use Random Keypad (Before Study)	22% In Favor 78% Not in Favor

B. Apparatus

Our experiments were conducted by using an Android implementation of the RandomPad application, designed specifically for this study. The application would display the keypad (Figure 3) and a short instruction of the task to perform. As the participants type on the keypad, the application records the dictated number (if applicable) along with the typed number and the corresponding time-stamps. The application was also programmed to the flow of our experiments, and it automatically enforced certain aspects of the experiments, such as random ordering of the natural and dictated typing scenarios, rest periods between various parts in each scenarios, pausing to record responses to the NASA-TLX and SUS scales, etc. The order in which the two experimental scenarios (discussed next) are presented to the participants is *counterbalanced* to prevent bias in the results. The keypad design in our application (for each randomization strategy) followed well-accepted standards [14], for example, the smallest height and width of a key was 57dp, which is comfortably higher than the standard minimum of 48dp. We used the Moto E smartphones (1st generation) in our study. The Moto E features a 4.3 inch touch screen with 540×960 pixels ($\sim$ 256 ppi pixel density). We also used the head-mounted Ergoneers Dikablis Professional Eye-Tracking system, equipped with two eye movement tracking cameras and a forward scene camera, to measure participants' eye activity while typing.

C. Session 1: Dictated Typing (DT)

In this experimental session, participants were prompted with visually and acoustically dictated sequences of pseudo-random single digit numbers. Length of each number sequence was uniformly varied between 3 (representing length of credit card security codes), 4 (representing length of phone unlock codes), 5 (representing length of zip codes), 7 (representing length of phone numbers without area code), 8 (representing length of birth dates), 10 (representing length of phone numbers with area code), and 16 (representing length of credit card numbers). This session of the study is further divided into three *parts*: *Default Keypad Typing*, *Randomized Keypad Typing*, and *Gray-scale Randomized Keypad Typing*. Each part consisted of ten *activities*, where in each activity the participants typed the dictated sequence of single digit numbers on the displayed keypad. There was a ten second time separation between each activity and a one minute separation between the three parts, allowing participants enough opportunities to rest.

1) Task: The primary task for participants is to follow the dictation and type the dictated digits on the displayed keypad. Each activity begins when participants are ready and they tap on the "Start" button on the smartphone screen. Immediately after tapping the "Start" button, the keypad appears and dictation starts. Participants can see the dictated digits on screen or hear the corresponding audio prompt, or both. No time restriction is imposed on participants, and new digits are dictated after the participant presses a key in response to the last dictated digit.

2) Part 1.1 – Default Keypad: In this part, participants type on the default keypad, with no randomization in key size or sequence. This serves as a reference point for our performance and accuracy evaluation.

3) Part 1.2 – Randomized Keypad: In this part, the RandomPad application generates and displays an instance of random keypad, using the randomization strategy assigned to the participant. For KSR, KLR randomization strategies, a new instance of random keypad is generated and displayed after every key press. For other (RR, CR, IKR) randomization strategies, the instance of random keypad generated at the beginning of each activity is used for the entire activity.

4) Part 1.3 – Gray-scale Randomized Keypad: This part is administered only to those participants who are assigned RR, CR, and IKR strategies. The RandomPad application generates and displays an instance of random keypad, using the randomization strategy assigned to the participant. Additionally, keys on the randomized keypad are shaded with an ascending shade of gray, with color of key "0" being lightest and "9" the darkest, as discussed earlier.

D. Session 2: Natural Typing (NT)

In this session, participants were instructed to type information already known to them at their own natural pace. Participants were asked to type their residence area code (3 digits), zip code (5 digits), phone number without area code (7 digits), birth date (8 digits), or phone number with area code (10 digits), in random order. This session is also divided into three parts, i.e., Default Keypad Typing, Randomized Keypad Typing, and Gray-scale Randomized Keypad Typing, with each part consisting of ten activities. The time intervals between parts and activities remain the same as before.

1) Task: The primary task for participants is to type familiar numbers on the random keypad. Before beginning each typing activity, the participants are visually communicated about the number they have to type in that activity. The activity begins when participants are ready and tap on the "Start" button on the smartphone screen. Immediately after tapping the "Start" button, the keypad appears and participants are expected to start typing. No time restriction is imposed on participants, and activity finishes when the participants are finished typing in the expected number of digits (based on what was asked to type).

2) Parts: Similar to the dictated typing session (i.e., session 1), the natural typing session (i.e., session 2) has three parts: (i) Part 2.1 – Default Keypad, (ii) Part 2.2 – Randomized Keypad, and (iii) Part 2.3 – Gray-scale Randomized Keypad. The description of the activities performed by the participants and the keypads used in each of these parts is similar to the previous session; the only difference is that rather than typing a dictated sequence of numbers in each activity, the participants type a known sequence of numbers (as outlined before) at their own pace.

E. Procedure and Data Collection

Participants were seated in a lab environment and given a smartphone installed with the RandomPad application. Before beginning each session of the study, a short video was shown to the participants explaining the task to be completed in each part. If participants made a mistake during typing, they were instructed to continue to the next number without attempting to rectify it. The mistyping is recorded for evaluating accuracy in typing. To measure accuracy during the Natural Typing session, the residence area code (3 digits), zip code (5 digits), phone number without area code (7 digits) and date of birth (8 digits) were collected from each participant beforehand in the pre-survey. As discussed in sections V, subjective usability and mental workload perceptions of participants is captured using the SUS and NASA-TLX scales. The SUS and NASA-TLX surveys were completed by the participants after each part of either session 1 or session 2, whichever came temporally later (as the order of the Dictated Typing and Natural Typing sessions is counterbalanced). After finishing both sessions of the experiment, the participants completed a post-survey.

VII. Results

In this section, we outline results from both the natural and dictated typing sessions of our experiments.

Q1: Do randomized keypads increase the task completion time when compared to the default keypad?: We investigate the difference in task completion times between default and randomized keypads with the null hypothesis that their means are not significantly different. Figures 6 and 7 show the average time taken by the participants to type a key, in the dictated and natural typing sessions, respectively. The results are further categorized by the keypad randomization type. It is evident that the average task completion time on random keypads is increased in both cases, compared to the default keypad. However, the overall task completion time is less in natural typing, compared to dictated typing. This is most likely due to the extra cognitive task performed to follow the dictation while typing. Among the five randomization strategies, typing on IKR (mean differences w.r.t. default keypad, $d^{\mu DT}_{IKR} = +249.78$ ms, $d^{\mu NT}_{IKR} = +140.66$ ms) and KSR ($d^{\mu DT}_{KSR} = +263.61$ ms, $d^{\mu NT}_{KSR} = +137.07$ ms) took relatively more time compared to CR ($d^{\mu DT}_{CR} = +105.14$ ms, $d^{\mu NT}_{CR} = +98.75$ ms), KLR ($d^{\mu DT}_{KLR} = +116.32$ ms, $d^{\mu NT}_{KLR} = +107.75$ ms), and RR ($d^{\mu DT}_{RR} = +106.60$ ms, $d^{\mu NT}_{RR} = +144.62$ ms). In two-tailed paired sample t-test [25], the combined mean increase in time taken by participants to type a key are $d^{\mu DT} = +168.3$ ms and $d^{\mu NT} = +125.8$ ms, with $p < 0.001$ in both DT and NT. Since $p < 0.05$ (the assumed significance level), the null hypothesis is rejected. In other words, we found that randomized keypads do increase task completion times, by approximately 21% for dictated and 16% for natural typing.

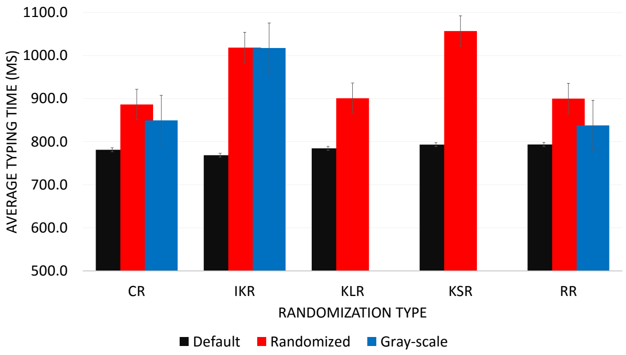


Fig. 6: Average time taken per key typed in Dictated Typing.

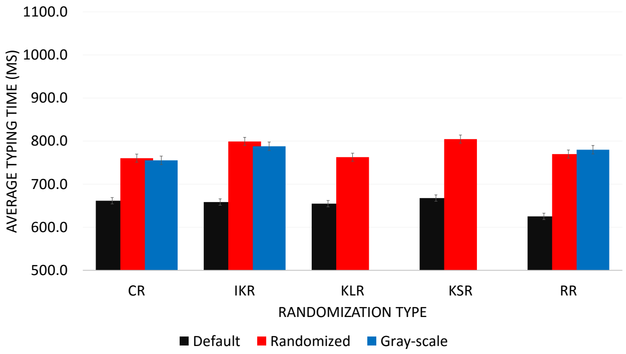


Fig. 7: Average time taken per key typed in Natural Typing.

Q2: Do randomized keypads increase the error rate in the primary task?: We investigate the difference in typing accuracy between default and randomized keypads with the null hypothesis that their means are not significantly different. Figures 8 and 9 shows the average accuracy of the typed numbers, for the natural and dictated typing sessions, respectively. The results are further categorized by the keypad randomization type. In two-tailed paired sample t-test, the combined mean decrease in typing accuracy are $d^{\mu DT} = -0.53\%$ and $d^{\mu NT} = -0.77\%$, with $p = 0.06$ in DT and $p = 0.003$ in NT. As $p > 0.05$ for dictated typing, the null hypothesis is marginally accepted. However, the null hypothesis is rejected in case of natural typing, which means the typing accuracy may be lower on the randomized keypads. Nonetheless, mean accuracies in all five randomization strategies are above 95% for both default and randomized keypads, with mean difference less than 1% compared to the default keypads. As the participants' primary task was to type the number sequences correctly, and not to type as fast as possible, it may be concluded that the task completion time was traded off for higher accuracy by the

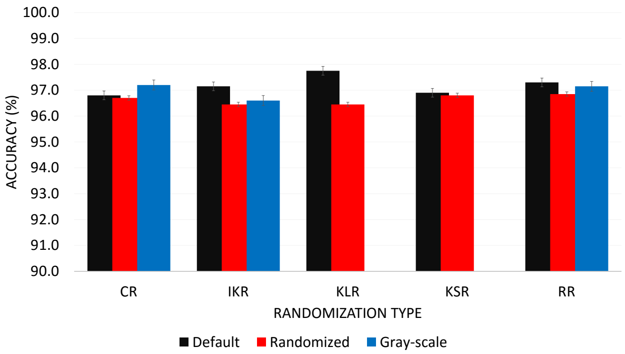


Fig. 8: Dictated Typing accuracy.

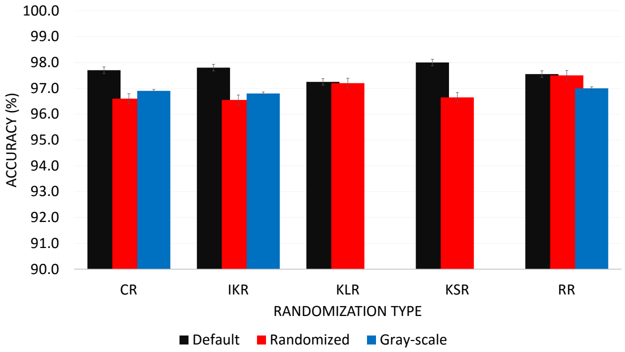


Fig. 9: Natural Typing accuracy.

participants.

Q3: Is there a learning curve associated with randomized keypads?: In order to analyze if the typing performance (speed and accuracy) improves with more usage of the randomized keypad, we compare the average per key typing time for the first and last ten numbers typed with random keypads, in the natural typing session. We observed that the average task completion time is significantly lower for the last ten numbers (compared to the first ten numbers), for all five randomization strategies. The overall mean drop in per key typing time is recorded as $d^{\mu NT}_{L10-F10} = -163.09$ ms, with $p < 0.001$. However, we did not observe any significant improvement in accuracy. This suggests that there exists a learning curve in using RandomPad, primarily to learn the randomization strategy, rather than memorizing an instance. As we see marked improvements within a relatively short experimental duration, we are optimistic that randomized keypad usage performance will only further improve with prolonged use.

Q4: How much more effort do randomized keypads take, compared to the default keypad?: We investigate the difference in user effort required to type on the default versus randomized keypads with the null hypothesis that the mean of their NASA-TLX scores are not significantly different. However, the randomized keypads scored higher on the NASA-TLX scale for all five randomized keypads (Figure 10), with an overall mean difference $d^{\mu} = +16.87$ and $p < 0.001$. This suggests that participants required considerably more perceived effort to use the randomized keypads. This observation is somewhat expected because of their familiarity with the default keypad. KLR is reported to take the least effort ($d^{\mu}_{KLR} = +9.94$) compared to the other four randomization strategies.

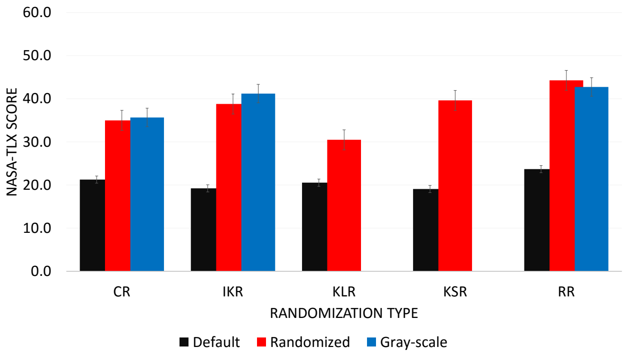
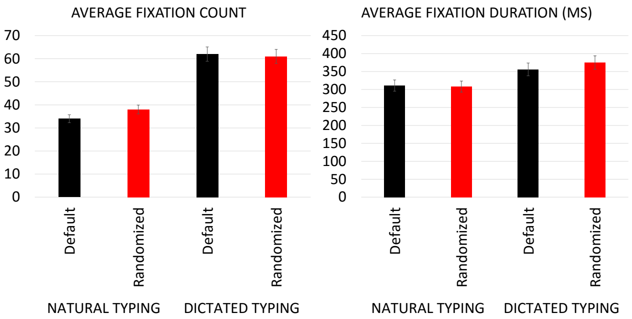


Fig. 10: NATA-TLX scores for all the five randomization strategies. Dictated and Natural Typing are combined. Lower scores signify lesser workload for the user.



(a)

(b)

Fig. 11: (a) Average fixation count and (b) average duration per fixation, for Natural and Dictated Typing. Lower scores signify lesser workload for the user.

In addition to subjective metrics such as NASA-TLX, we complement our results for workload by means of quantitative metrics such as fixation counts and fixation duration from the eye tracking data. Due to the significant setup time and overhead involved, we collected eye tracking data from only a subset of participants (more specifically, 53 participants). Figure 11 shows the average fixation count and fixation duration recorded during the experiments involving these participants. Higher fixation count indicates that the participants had to view more areas of interest (AOI) before they were able to locate the target key. The average fixation count on randomized keypads is increased (+4) in case of natural typing, but marginally decreased (-1) in case of dictated typing. On the other hand, the average fixation duration (time spent per AOI) increased (+19 ms) in case of dictated typing, but marginally decreased (-3 ms) in case of natural typing. These results show that, in certain scenarios, randomized keypads do increase the difficulty in locating keys, resulting in increased workload.

Q5: How much less usable randomized keypads are, compared to the default keypad?: We investigate the difference in perceived usability of the default versus randomized keypads with the null hypothesis that the mean of their SUS scores are not significantly different. However, the randomized keypads faired lower than the default keypad (Figure 12). The overall mean difference $d^{\mu} = -25.80$ and $p < 0.001$ suggests that participants felt that the default keypad is more usable.

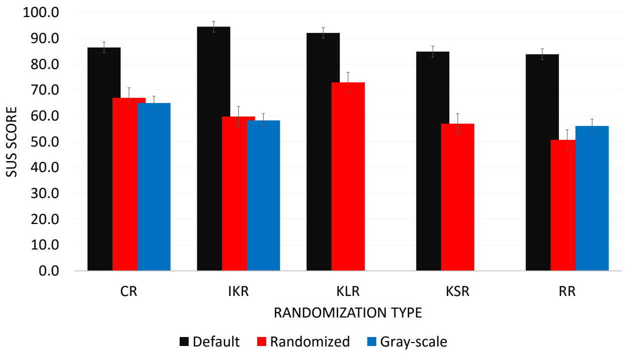


Fig. 12: SUS scores for all the five randomization strategies. Dictated and Natural Typing are combined. Higher scores signifiy better usability.

This observation is also somewhat expected because of their familiarity with the default keypad. KLR is again reported to be the most usable ($d^{\mu}_{KLR} = -19.12$) compared to the other four randomization strategies.

In the SUS scale, a score lower than 50 indicates unacceptable for use; a SUS score larger than 70 indicates acceptable for use; A score between 50 to 70 indicates marginally usable. RR (mean SUS score = 56.13) is significantly lower on the SUS scale compared to the other five randomization strategies. On the other end, KLR (mean SUS score = 73) provides acceptable usability and CR (mean SUS score = 67) is just below the 70 mark. Thus, from a perceived usability perspective, KLR is preferred by the users over other randomization strategies. Again, we are optimistic that randomized keypads can achieve better usability scores if users get familiarized with the distinct layouts.

Q6: Does gray-scale shading of randomized keypads improve usability?: On the NASA-TLX (Figure 10) and SUS (Figure 12) scores, there are no significant differences between the randomized keypads without gray-scale shading versus randomized keypads with gray-scale shading, indicating that contrasting gray-scale shades on the keypad does not lower the perceived workload or improve the perceived usability. However, we observe that gray-scale shaded randomized keypads marginally lower the task completion times. The average task completion time on gray-scale randomized CR, IKR and RR keypads during dictated typing session is 901.70 ms versus 935.09 ms for just randomized CR, IKR and RR keypads. This shows that our gray-scale shading scheme does not significantly improve the usability of the RandomPad interface. This may be because in our gray-scale shading scheme the contrast between the shade of the digits and that of the background is not optimal for certain keys, which can create difficulty during the reading of those keys [34]. The usability of gray-scale keypads could be potentially improved by adjusting and optimizing this contrast between the different shades.

Q7: Are smartphone users interested in adopting randomized keypads?: In the initial pre-survey recorded before the participants were introduced to side-channel keystroke inference attacks, only 22% of the participants reported that they would be willing to use a randomized version of the keypad. On being more informed about the dangers of side-channel keystroke inference attacks and how randomized keypads help protect against such attacks, and after completing the experimental trials, as many as 80% of the participants reported in the post-survey that they would be willing to use a randomized keypad in order to protect their privacy. Those participants who reported "No" to this question (i.e., were not willing to use the randomized keypad) in the post-survey reported that they were more familiar, therefore more comfortable, with the default keypad and that the randomization (of the keypad) was confusing to them. Those who changed their answer to "Yes" in the post-survey (i.e., were willing to use the randomized keypad) primarily reported that their reason for using an unfamiliar interface, such as, a randomized keypad, would be to primarily enhance their privacy and to prevent hackers from stealing their personal information.

VIII. Discussions

In this section, we discuss some of the implications of our study, and how researchers and developers can use our evaluation results in order to implement and/or improve RandomPad.

A. *Privacy-Usability Trade-Off*

In our evaluation, it was clear that RandomPad negatively affects users' performance, workload and perceived usability. While this was intuitive and an expected result, the effect on usability, although present, was not large enough to make the interfaces completely unusable. It should also be noted that 80% of the participants were still willing to use RandomPad on a regular basis in order to input their sensitive information. As participants were introduced to only one randomization strategy (to receive an unbiased opinion about each strategy), it is also likely that they may like another strategy better. Therefore, we analyzed the privacy-usability trade-off of the five different randomization strategies based on security assurance ranking (Table I) and usability ranking (calculated using typing speed, workload and perceived usability)[1]. Table III shows the usability ranking calculations of the five different randomization strategy. Comparing Table I and III, we see that KLR ranks relatively highest on both (3 + 1 = 4) tied with IKR (1 + 3 = 4), followed by RR (2 + 4 = 6), and CR (5 + 2 = 7), and KSR (4 + 4 = 8), respectively. In other words, KLR and IKR provides the best balance between security and usability, while KSR provides the least.

B. *Recommendations to Developers*

Users type sensitive information only a fraction of the time they use a keypad. Having an always-on randomized keypad may be an inconvenience to the users, whom may then choose to not use randomized keypads altogether. A good design should have an easily accessible and user-controllable (soft) switch to turn on or off the key randomization, as and when desired by the user. Whenever users feel that the information they are going to type is sensitive in nature, they should be able to easily turn on the randomization of the keypad. After

[1] As accuracy was marginally varying, we exclude it as a factor in the usability ranking calculations.

TABLE III: Usability rankings of the five proposed randomization strategies calculated using average typing speed (lower better; dictated and natural typing combined), workload (lower better) and perceived usability (higher better). Lower least rank is better usability.

Randomization Strategy	Typing Speed Rank	Workload Rank	Perceived Usability Rank	Summed Usability Rank (Least Rank)
CR	1	2	2	5 (2)
IKR	4	3	3	10 (3)
KLR	2	1	1	4 (1)
KSR	5	4	4	13 (4)
RR	3	5	5	13 (4)

they finish typing the sensitive information, or in the case they are typing non-sensitive information, they should be able to easily turn off the randomized keypad and continue to use the default keypad.

C. Limitations

Even though RandomPad is able to protect users against several types of side-channel keystroke inference attacks, it fails to protect against visual eavesdropping, also known as *shoulder-surfing*. There are certain authentication schemes that can defend against visual eavesdropping [31], but they (i) require more effort from users and (ii) cannot be used to type sensitive information other than device unlock codes.

IX. Conclusion

With the increasing number of side-channel attacks targeting mobile keypads, user privacy is at stake. In this paper, we proposed to used randomized keypads for typing sensitive information on mobile device keypads. Randomized keypads are able to sufficiently alter keystroke characteristics, such that most of the side-channel attacks will fail. However, with users accustomed to the default keypad for years, randomized keypads face usability issues. Therefore, we comprehensively evaluate the usability of randomized keypads, with the help of 100 participants. We found that randomized keypads can increase task completion time. We also found that randomized keypads are perceived to be less usable and more work. However, the learning curve associated with randomized keypads can improve user performance and usability with prolonged use. Interestingly, even with the degraded usability of randomized keypads, participants were willing to use it for improved privacy.

References

[1] E. H. Adelson. Perceptual Organization and the Judgment of Brightness. *Science*, 262, 1993.

[2] E. Beck, M. Christiansen, J. Kjeldskov, N. Kolbe, and J. Stage. Experimental evaluation of techniques for usability testing of mobile systems in a laboratory setting. 2003.

[3] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 1996.

[4] L. Cai and H. Chen. TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion. In *HotSec*. USENIX Association, 2011.

[5] L. Cai, S. Machiraju, and H. Chen. Defending against sensor-sniffing attacks on mobile phones. In *MobiHeld*. ACM, 2009.

[6] M. Conti, V. T. N. Nguyen, and B. Crispo. CRePE: Context-Related Policy Enforcement for Android. In *Information Security*. Springer, 2010.

[7] A. Dix. *Human-Computer Interaction*. Springer, 2009.

[8] D. Foo Kune and Y. Kim. Timing Attacks on PIN Input Devices. In *CCS*. ACM, 2010.

[9] R. Haigh. The Ageing Process: A Challenge for Design. *Applied Ergonomics*, 24(1), 1993.

[10] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in Psychology*, 52, 1988.

[11] K.-Y. Huang. Challenges in Human-Computer Interaction Design for Mobile Devices. In *WCECS*, volume 1. IAENG, 2009.

[12] S. R. Humayoun, S. Hess, F. Kiefer, and A. Ebert. Patterns for Designing Scalable Mobile App User Interfaces for Multiple Platforms. In *28th British HCI Conference*. BCS, 2014.

[13] M. Ishida, P. H. Frank, K. Doi, and J. L. Lehr. High Quality Digital Radiographic Images: Improved Detection of Low-Contrast Objects and Preliminary Clinical Studies. *Radiographics*, 3(2), 1983.

[14] W. Jackson. Android UI Layout Conventions, Differences and Approaches. In *Pro Android UI*. Springer, 2014.

[15] M. Jones and G. Marsden. Mobile Interaction Design. 2006.

[16] LG User Guide. Lock screen. https://www.lg.com/us/mobile-phones/VS985/Userguide/426.html.

[17] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic. (Smart)Watch Your Taps: Side-channel Keystroke Inference Attacks Using Smartwatches. In *ISWC*. ACM, 2015.

[18] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang. ACCessory: Password Inference Using Accelerometers on Smartphones. In *HotMobile*. ACM, 2012.

[19] M. Pattison and A. W. Stedmon. Inclusive Design and Human Factors: Designing Mobile Phones for Older Users. *Psychnology Journal*, 4(3), 2006.

[20] K. Rayner, T. J. Slattery, and N. N. Bélanger. Eye movements, the perceptual span, and reading speed. *Psychonomic Bulletin & Review*, 17(6), 2010.

[21] Y. S. Ryu, D. H. Koh, B. L. Aday, X. A. Gutierrez, and J. D. Platt. Usability evaluation of randomized keypad. *Journal of Usability Studies*, 5(2), 2010.

[22] A. Sears and Y. Zha. Data Entry for Mobile Devices using Soft Keyboards: Understanding the Effects of Keyboard Size and User Tasks. *International Journal of HCI*, 16(2), 2003.

[23] P. Shrestha, M. Mohamed, and N. Saxena. Slogger: Smashing motion-based touchstroke logging with transparent system noise. In *WiSec*. ACM, 2016.

[24] L. Simon and R. Anderson. Pin skimmer: Inferring pins through the camera and microphone. In *SPSM*. ACM, 2013.

[25] G. W. Snedecor. Statistical methods: Applied to experiments in agriculture and biology. 1946.

[26] Software House. Scramble Keypad SP-100. http://www.swhouse.com/products/.

[27] B. Spehar and C. Owens. When Do Luminance Changes Capture Attention? *Attention, Perception and Psychophysics*, 74(4), 2012.

[28] L. Steenbekkers, J. Dirken, and C. Beijsterveldt. Design-Relevant Functional Capacities of the Elderly, Assessed in the Delft Gerontechnology Project. In *13th Triennial Congress of the International Ergonomics Association*, 1997.

[29] J. Sun, Xiaocong, Y. Chen, J. Zhang, Y. Zhang, and R. Zhang. VISIBLE: Video-Assisted Keystroke Inference from Tablet Backside Motion. In *NDSS*. ISOC, 2016.

[30] X. Tizon and Ö. Smedby. Segmentation with Gray-Scale Connectedness Can Separate Arteries and Veins in MRA. *Journal of Magnetic Resonance Imaging*, 15(4), 2002.

[31] Q. Yan, J. Han, Y. Li, J. Zhou, and R. H. Deng. Designing leakage-resilient password entry on touchscreen mobile devices. In *ASIACCS*. ACM, 2013.

[32] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao. Blind Recognition of Touched Keys on Mobile Devices. In *CCS*. ACM, 2014.

[33] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu. Fingerprint Attack Against Touch-enabled Devices. In *SPSM*. ACM, 2012.

[34] S. Zuffi, C. Brambilla, G. Beretta, and P. Scala. Human computer interaction: Legibility and contrast. In *ICIAP*. IEEE, 2007.