Preventing Shoulder Surfing using Randomized Augmented Reality Keyboards

Anindya Maiti, Murtuza Jadliwala *and* Chase Weber
Electrical Engineering and Computer Science Department
Wichita State University, USA
Email: a.maiti@ieee.org, murtuza.jadliwala@wichita.edu, me@chaseweber.com

***Abstract*—Shoulder surfing or adversarial eavesdropping to infer users' keystrokes on physical QWERTY keyboards continues to be a serious privacy threat. Despite this, practical and efficient countermeasures against such attacks are still lacking. In this paper, we propose *keyboard randomization* as a simple, yet effective, countermeasure against various types of keystroke inference attacks. Our proposal consists of several keyboard randomization strategies which randomizes or changes the position of keys on the keyboard. The randomized keyboard is then projected to the typing user by means of an augmented reality wearable device. As the randomized keyboard is visually superimposed over the actual physical keyboard, and is visible only to the typing user through the augmented reality device, it acts as an effective countermeasure against both side-channel and visual-channel based keystroke inference attacks. We implement our proposed solution on a commercially available augmented reality device and conduct preliminary evaluations to validate its performance and effectiveness.**

***Index Terms*—Eavesdropping, keystroke inference, random keyboard, augmented reality.**

I. Introduction

Physical QWERTY keyboards are the most widely adopted input interface for personal and portable computing systems. These keyboards have also been a constant target for various forms of "*shoulder surfing*" attacks, where the goal of an adversary is to obtain or infer users' keystrokes by directly, but surreptitiously, observing the typing user (and the keyboard) or eavesdropping on certain information directly related to the typing activity being performed. The first case, where the adversary has a covert visual access to the typing user, is a more common and easy-to-execute threat. Such threats are also the most difficult to protect against, especially by means of traditional cryptography-based or other information manipulation and hiding techniques. For instance, Roth et al. [1] proposed an oracle-based multi-round protocol for PIN entry by color coding keys into two shades (black and white). This scheme takes advantage of limitations in human cognitive capabilities to overcome shoulder surfing, however, Kwon et al. [2] recently showed that covert attention and perceptual grouping can improve information processing by humans, thus rendering Roth et al.'s approach ineffective.

Alternatively, there exists other forms of shoulder surfing attacks that, rather than relying on the direct visual channel, take advantage of indirect information channels (or side-channels) to infer users' keystrokes. For instance, Vuagnoux et al. [3] use electromagnetic emanations from external keyboards (both wired and wireless) to infer keystrokes, whereas, Berger et al. [4] have accomplished a similar feat by using acoustic emanations originating due to typing on these keyboards. As another variation of non-visual shoulder surfing, Marquardt et al. [5] utilized the vibrations sensed by a smartphone accelerometer (positioned in the proximity of the target keyboard) to infer a users' keystrokes on the keyboard. Maiti et al. [6] proposed a similar attack by taking advantage of motion information available from wrist-wearable devices such as smartwatches. More recently, Ali et al. [7] demonstrated the ability to infer keystrokes by observing the unique changes in the radio signal channel statistics caused during typing.

Interestingly, the success of all of the above attacks rely on one common assumption: the adversary has knowledge of the layout, and in some cases, even the exact model, of the keyboard used by the target user. This assumption, at least the former, is reasonable as most modern QWERTY keyboards have a standard layout of keys. Intuitively, this means that if the keyboard layout is changed from the default to something different, and if this new or changed layout is not known to the adversary, then at least the above side-channel or non-visual attacks will not succeed. In other words, a dynamic keyboard layout strategy is an appealing defense strategy against side-channel keystroke inference or shoulder surfing attacks. Such a strategy is also not far-fetched as a similar concept is currently being used in other types of commercial products, for instance, to enhance the security of electronic door access control systems [8]. Ryu et al. [9] also performed a usability evaluation of such randomized numeric keypads.

Despite the promise, there are two critical technical challenges with respect to implementing this solution for external or physical QWERTY keyboards. First, the layout of these keyboards cannot be easily modified; it is possible to modify the mapping between the physical keys (on the keyboard) to the actual character they represent, however such a keyboard will be extremely challenging to use as the users will have to memorize the mapping between the physical keys on the keyboard and the actual characters they represent. Second, even if somehow it was possible to dynamically change the physical layout of the keyboard, such a change would not protect against shoulder surfing attacks by an adversary that has covert visual access of the target keyboard (or the user

978-1-5090-4338-5/17/$31.00 ©2017 IEEE

typing on the keyboard).

In this paper, we overcome the above technical challenges and propose a system for randomizing external keyboard layouts by making a novel and interesting use of augmented reality devices. One key advantage of our proposal is that it is able to overcome all forms of shoulder surfing attacks, including those possible through direct visual access of the target keyboard. Our proposal consists of several keyboard layout randomization strategies, each of which assigns a unique non-standard position to the keys on the keyboard which is unknown to the adversary. The randomized keyboard is then projected to the typing user by means of an augmented reality wearable device. As the randomized keyboard is visually superimposed over the actual physical keyboard, and is visible only to the typing user through the augmented reality device, it acts as an effective countermeasure against both side-channel and visual channel-based keystroke inference or shoulder surfing attacks. We implement our system on the commercially available EPSON Moverio BT-200 [10] augmented reality device and validate its performance and effectiveness by means of preliminary empirical usage data from a small number of test subjects.

II. Related Work

Protection against shoulder surfing attacks have received significant attention in the literature, with several different solution directions proposed and analyzed. For instance, Kumar et al. [11] proposed EyePassword, where orientation of the user's pupils were used for password entry. The authors further showed that such an approach requires only marginal additional time over using a keyboard and that the error rates due to this approach is similar to those of using a keyboard. In order to thwart shoulder surfing attacks against traditional alphanumeric passwords, graphical passwords was also proposed as an alternative where the user was expected to select a predetermined image or set of images in a particular order [12], [13]. Human subject studies showed that such graphical passwords were perceived to provide reasonable protection against visual shoulder-surfing attacks [14], however it was later showed that those conclusions were not completely valid [15], [16]. More recently, Yan et al. [17] proposed CoverPad, a leakage-resilient password entry system for touchscreen mobile devices, where a user is expected to cover the touchscreen (by hand) to securely read a hidden message that contains information on removing the correlation between the actual password (or PIN) and the one entered by the user. One common theme in most (if not all) past research efforts in this direction is that they focus only on preventing shoulder surfing attacks against authentication information such as passwords or PINs. Our proposed design and system protects all kinds of textual inputs, including, but not limited to, passwords and authentication information, against both visual and side-channel shoulder surfing attacks.

III. Adversary Model

We consider the scenario of a target user typing on an external or physical QWERTY keyboard and an adversary who intends to carry out a shoulder surfing attack on the user in order to infer his/her keystrokes. The attacker may carry out the shoulder surfing attack using various channels. He may have a covert visual access of the physical keyboard and the user's typing activity. This could be achieved by the adversary surreptitiously watching the target user's keyboard as he is typing or by gaining access (either legally or in an unauthorized fashion) to a video feed of the user's keyboard and typing activity by means of a camera or a surveillance device. We assume that the information being typed is protected from visual eavesdropping of the display screen (or monitor). This is a reasonable assumption as most applications obfuscate confidential on-screen information or text such as passwords and PINs by symbols or special characters (e.g. asterisk). Alternatively, the monitor could also be protected using a privacy screen. It should be noted that these measures do not protect against an adversary eavesdropping on the keyboard and user's keystrokes. If a visual channel is unavailable to the adversary, he may attempt to accomplish the keystroke inference attack using other forms of information side-channels, such as, electromagnetic emanations from the keyboard [3], vibrations [5] or acoustic [4] signals captured during the typing activity, by observing the changes in the radio signal channel statistics [7] or by capturing the motion information of the typing hand [6]. As discussed later, our protection mechanism involves the use of commercial off-the-shelf augmented reality glasses such as EPSON BT-200. We assume that the display of this augmented reality device is visible only to the target user, and that this device is secured from the adversary.

IV. Proposed Defense Model

Consider the scenario where a user wants to type a sensitive piece of information on an external QWERTY keyboard in the presence of an eavesdropping adversary, as shown in Figure 1. To obscure keystrokes from the eavesdropping adversary, we propose the use of randomized keyboard layouts in cohort with an augmented reality device. In our proposal, the user privately sees a randomized (using strategies explained later) keyboard layout augmentation over the actual keyboard, where keys are positioned differently from the default QWERTY layout, by means of an augmented reality device or glasses (shown in Figure 1). The augmentation is done such that the randomized keys are superimposed over the existing keys of the physical keyboard. This can be achieved with the help of marker (Figure 3) or character recognition [18] of individual keys on the physical keyboard. Also, the augmented reality device establishes a temporary secure wireless link with the computer (with which the keyboard is attached to) so as to communicate the key mapping between the randomized augmented layout and the underlying QWERTY layout. This secure link can be established using widely available wireless technologies, such

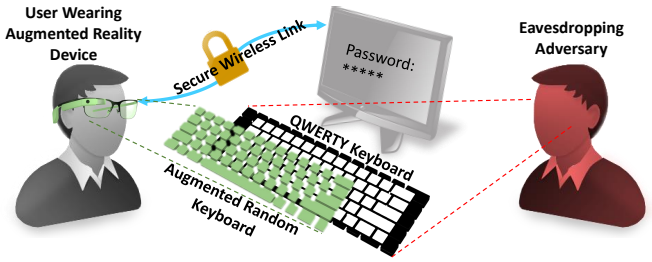


Fig. 1. The proposed defense model, where the user wearing the augmented reality device sees and types on the randomized augmented keyboard. The eavesdropping adversary can observe only the default QWERTY layout of the physical keyboard.

as Bluetooth, and made secure using symmetric encryption schemes, such as AES.

Whenever the user presses a key based on observation of the augmented layout, the computer uses the mapping between the randomized and QWERTY layouts to substitute the character typed on the physical keyboard with the correspondingly placed key in the augmented layout. The adversary, however, can only eavesdrop on the physical keyboard having the default QWERTY layout. As the adversary does not see (or is unable to eavesdrop on) the augmented layout and does not have access to the mapping, it cannot infer the character actually registered by the computer system.

A. Randomization Strategies

To prevent keystroke inference attacks, an important task in the proposed system is to ensure that the layout of the augmented characters is unpredictably different from the default QWERTY layout. Moreover, as an adversary can gain semantic knowledge from multiple observations and re-train his attack framework, changing the augmented keyboard layout just once (or in a very predictable or insignificant fashion) will not be an effective defense. To prevent an adversary from knowing the keyboard layout in use at any given time, the change in layout should be *randomized*. Accordingly, in our proposed system, every time the user wants to type sensitive text, a newly randomized keyboard layout is augmented over the physical keyboard. The new mapping of the randomized layout to the underlying physical keys is also updated accordingly on the computer side by means of the secure communication link. In this paper, we focus on randomization of just the twenty-six alphabets (Figure 2), however it could be easily extended to all keys. Below, we list a few representative (by no means an exhaustive list) randomization strategies that can be used to change the keyboard layout:

(i) Individual Key Randomization (IKR): This strategy randomly assigns positions to each alphabet or letter on the augmented keyboard layout, without any relation to its actual position on the QWERTY layout. An instance of IKR is shown in Figure 4.

(ii) Row Shifting (RS): In this strategy, the alphabets in each row of the QWERTY layout (rows in Figure 2) are circularly left or right shifted by a random number of keys on the augmented layout. In other words, each alphabet on the augmented layout is found on the same row as in the QWERTY layout, however its position is shifted left or right by a random number.

(iii) Column Shifting (CS): In this strategy, the alphabets in each column of the QWERTY layout (columns in Figure 2) are circularly top or bottom shifted by a random number of keys on the augmented layout. In other words, in CS each alphabet on the augmented layout is found on the same column as in the QWERTY layout, however its position is shifted top or bottom by a random number. As the column (correspondingly, row in RS) of each alphabet and the order of alphabets in each column (correspondingly, row in RS) is maintained in CS, intuitively it appears that it may be comparatively easier for a user to search for an alphabet on the CS and RS layouts. We want to validate if this is true in practice, and thus the reason for choosing these two layouts in addition to IKR.

While several additional randomization strategies can be envisioned, for conciseness we limit the current discussion to just the above three strategies.

B. Security Analysis

As the keyboard layout is randomized, the best an adversary (assumed to know the randomization strategy used by it's target) can do is guess the mapping between the randomized and QWERTY layouts. We use the successful guessing probability to indicate the level of security assurance each randomization strategy provides in the presence of an eavesdropping adversary. For a particular randomization strategy, the lower this probability is, the higher the security assurance provided by it.

In IKR, the probability that an adversary correctly guesses the mapping of a particular alphabet is $\frac{1}{26}$, i.e., uniformly distributed. Moreover, the probability that the adversary guesses the entire mapping correctly is $\frac{1}{26!} = 2.4 \times 10^{-27}$, which is negligibly small. However, in case of RS and CS, the adversary can improve it's guessing, based on the relative positioning of key within a row and column, respectively. Knowing that keys within a shifted row remain in (circular) order, for a row shifted keyboard (RS), the adversary only needs to guess the random length of shifting. The probability that an adversary correctly guesses the length of a row's shifting is $\frac{1}{10}$, $\frac{1}{9}$, and $\frac{1}{7}$, for rows 1, 2, 3, respectively (as labeled in Figure 2). Therefore, the probability that the adversary guesses the mapping for all 26 alphabets correctly is $\frac{1}{10} \times \frac{1}{9} \times \frac{1}{7} = 1.5 \times 10^{-3}$.

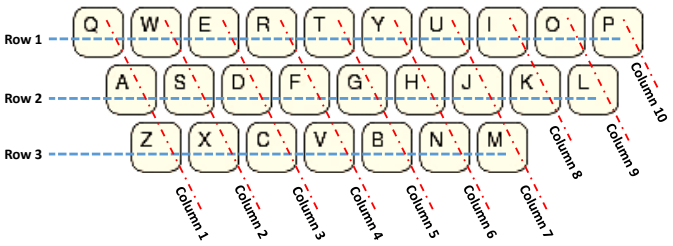


Fig. 2. Assumed rows and columns for RS and CS strategies.

Fig. 3. A QWERTY keyboard with alphabetic markers glued on top of the corresponding alphabet keys.

Fig. 4. An instance of an augmented keyboard with IKR strategy as observed by the typer on the EPSON Moverio BT-200.

Similarly, for CS, the probability that the adversary correctly guesses the length of random shifting is $\frac{1}{3}$ for columns 1 to 7, $\frac{1}{2}$ for columns 8 and 9, and 1 for column 10 (as labeled in Figure 2). Therefore, the probability that the adversary guesses the mapping for all 26 alphabets correctly is $(\frac{1}{3})^7 \times (\frac{1}{2})^2 \times (1)^1 = 1.1 \times 10^{-4}$. Thus, given the adversary knows the strategy being used, IKR is probabilistically the most secure while RS is the least secure randomization strategy, which is also intuitive. However, in practice the adversary will not know the randomization strategy currently in use, thus making these strategies even more secure. The security of the system could be further improved by re-randomizing or reshuffling the keyboard at regular intervals by using a particular randomization technique (and parameters). However, if the keyboard layout is changed too often, the usability may suffer drastically, because the keyboard will change even before users get habituated to the current one. This trade-off between security and usability is what we intend to study by means of experiments involving human participants.

V. Evaluation

To validate the feasibility of the proposed system, we implement a proof-of-concept prototype and perform preliminary experimentation to evaluate system efficiency and performance parameters such as task completion times and typing accuracy. Next, we first present our prototype and experimental setup followed by results from our evaluation.

A. Study Design

We perform some preliminary evaluation of our proof-of-concept implementation with the help of data collected from human participants who use our prototype for typing. Below, we specify our experimental setup, tasks performed by the participants, and the empirical parameters used in the evaluation.

Experimental Setup: Figure 5 depicts the setup used in our evaluation. We recruited thirteen participants; all of them were familiar with typing on a QWERTY keyboard. The participants were seated in front of a keyboard, with a display screen in the background. We chose to use the Anker A7726121 Bluetooth keyboard because of its generic design. The keyboard was connected to the computer and the alphabet keys were covered with corresponding alphabetic markers (Figure 3). As a result, the keyboard was usable even as a regular QWERTY keyboard. Participants wore the EPSON BT-200 augmented reality device during the experiment. The EPSON BT-200 is equipped with a front facing camera with a resolution of 640×480 pixels, which enables augmented reality applications. The BT-200 also features the Android 4.1 platform, and our implementation of the augmented randomized keyboard was installed as an application. Our implementation of the augmented randomized keyboard uses the ARToolKit library [19]. We would like to stress, however, that in practice a specialized and expensive AR hardware, such as, the EPSON BT-200, is not required. We have also implemented an alternate smartphone application of our proposed augmented randomized keyboard which can be installed by users on their AR-friendly smartphones and used in conjunction with an affordable augmented reality viewer such as Google Cardboard.

Task: The participants were directed with audio-visual instructions on what to type on the keyboard. In the first part of the experiment, each participant typed all twenty six alphabets of English language in random order. In the second part of the experiment, each participant typed five familiar words: first name, last name, hometown, address street, and area of work. In the third part of the experiment, each participant typed an experimental password of their choice. For the second and third parts, ground truth was collected beforehand, in order to calculate typing accuracy. Participants repeated all three parts of the experiment four times; in default QWERTY (without the augmented randomized keyboard turned on), IKR, CS, and RS. The default QWERTY typing serves as a base line to

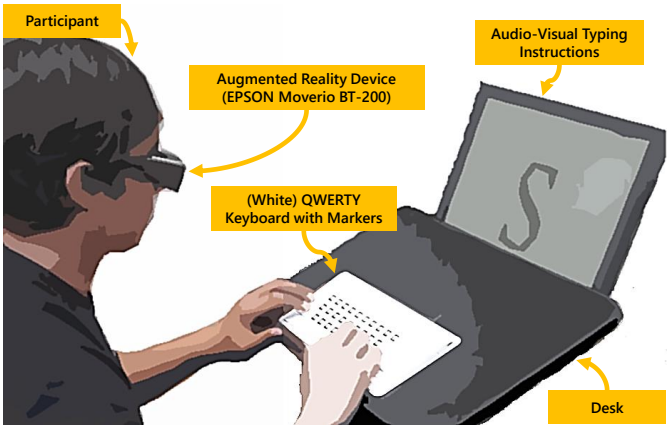


Fig. 5. A participant typing on the randomized augmented keyboard.

compare results obtained in the other three scenarios, where participants type using the augmented randomized keyboard. The order of the four typing scenarios was counterbalanced across participants [20], so as to minimize the chances of order effects. For consistency, the same instances of randomized keyboards (each for IKR, CS, and RS) are used by all participants. Participants were also given practice sessions before each part of the experiment, in order to allow them to get familiarized with the keyboard being used.

Empirical Parameters: In order to evaluate our implementation, we measure two usage-related parameters for each participant. For evaluating efficiency, we measure the participants' *typing speed* both on the standard QWERTY layout and on the proposed randomized layouts. Typing speed is measured as the average typing time (in seconds) per character for all the 100 typed characters. We use the computer's clock to log these time intervals. For evaluating performance, we measure the participants' *typing accuracy* for both the standard QWERTY and the randomized layouts. Typing accuracy is measured by enumerating the number of errors during typing by comparing each character instructed to be typed with the character actually typed by the participant. In addition to usage-related parameters, we also measure the users' perceived workload by using a standard metric such as NASA Task Load Index (NASA-TLX) [21].

B. Results

We outline results and observations from our experiments below.

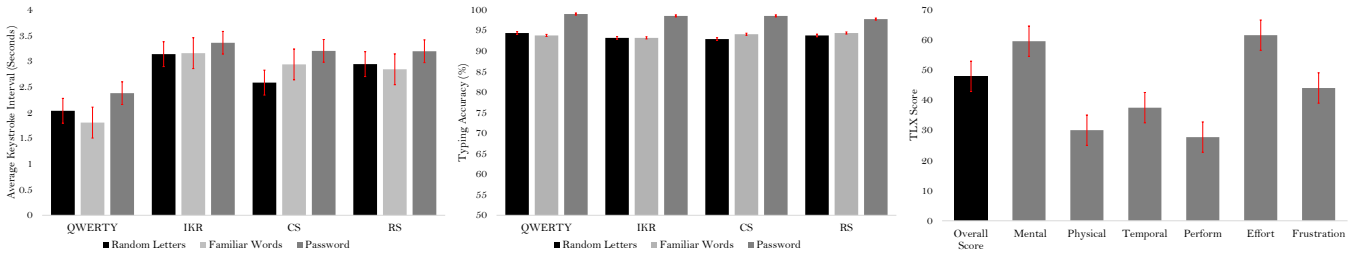
Typing Speed: The average time taken by all thirteen participants to type a key on the default QWERTY keyboard (with augmentation turned off) was 2.03, 1.80, and 2.37 seconds for random letters, familiar words, and password, respectively. Readers should note that this measurement includes the time taken by participants to hear/see the alphabet to type, search of the corresponding alphabet on the keyboard, and then key it. When the randomized keyboard augmentation was turned on with the IKR randomization strategy, the average time taken by the thirteen participants to type a key increased to 3.13, 3.15 and 3.36 seconds, respectively. Following a similar trait, in cases of CS and RS randomization strategies, the mean time taken by the thirteen participants to type a key increased (with respect to the QWERTY layout) to 2.58, 2.93 and 3.20 seconds, and 2.94, 2.84 and 3.19 seconds, respectively. Averaged results from each typing scenario are presented in Figure 6a. These results suggest that there is a notable increase in task completion time with the use of randomized augmented keyboards. As mentioned by some of the participants who are habitual with touch-typing, significant time was used up in searching for particular alphabets on the randomized (IKR) keyboard. A noteworthy observation is that the typing speed is slightly higher on keyboards randomized with RS and CS strategies, compared to IKR. Intuitively, this is due to the fact that a subset of keys stay *relatively* in the same position as on the QWERTY layout. Therefore, it somewhat eases the process of key search.

Typing Accuracy: The average typing accuracy for all thirteen participants in typing a key on the default QWERTY keyboard (with augmentation turned off) was 94.37%, 93.78%, and 99% for random letters, familiar words, and password, respectively. When the randomized keyboard augmentation was turned on with the IKR randomization strategy, the average accuracy for all thirteen participants in typing a key dropped marginally to 93.19%, 93.19%, 98.53%, respectively. However, typing accuracies in CS (92.89%, 94.08%, 98.53%) and RS (93.78%, 94.37%, 97.76%) randomization strategies were similar to the QWERTY keyboard, if not better. Averaged results from each typing scenario are presented in Figure 6b. After the experiment was completed, one of the participants expressed concerns about the lag in rendering of the keys, especially noticeable when the user moves his/her head. The delay in rendering may have confused the participants, and lead to longer task completion times and/or more errors in typing. Therefore, results suggest that if some of the issues with our proof-of-concept prototype are resolved, typing accuracy can be comparable to typing on default QWERTY keyboards. Readers may notice that password typing took the longest and was also more accurately typed than the random letters and familiar words. This occurrence is primarily because the participants had to carefully recall and type the experimental password (chosen at the beginning of the study), which most likely is not one of the passwords they use in real-life.

Perceived Task Load: 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 this scale in our experiments to capture the task load imposed on participants in using the augmented random keyboard. Figure 6c shows the average overall score as well as the six individual subscales. Using augmented random keyboard was perceived by participants to be mentally demanding and complex (59.61 - Mental). Participants also felt that the task required significant effort to accomplish (61.61 - Effort). Participants were also not entirely satisfied with the performance of our implementation (27.76 - Perform). However, the physical activity required and time pressure felt due to the pace at which the tasks were being completed are notably low (30.07 - Physical, 37.53 - Temporal). Participants felt moderately content, relaxed, and complacent during the task (44.07 - Frustration).

VI. Discussion

Generalization to Other Keyboards: One advantage of our proposed design is that it can be easily generalized and deployed across different types of keyboards/keypads. The use of character recognition, instead of the exemplary marker recognition used in our prototype, will enable such a generalized design. One application of such a generalized design can be found in systems such as ATM machines. Numeric keypads on ATMs, due to their open or unrestricted locations, are the most prone to shoulder surfing attacks. The proposed system could be used in this scenario, where a users' augmented reality device could communicate with the ATM by means



(a) Average time (b) Average typing accuracy (c) Workload assessment using NASA-TLX

Fig. 6. Experimental results involving thirteen participants.

of a secured wireless channel to exchange a per-transaction randomized layout. This layout can then be augmented over the actual numeric keypad of the ATM machine and made visible only to the user by means of his/her augmented reality device.

Hardware Limitations: The hardware and software of the augmented reality device plays a crucial role in the design and implementation of the proposed system. For example, the camera resolution of the EPSON BT-200 is extremely low (640×480 pixels), which makes marker recognition error-prone and difficult, especially if the user is at a distance from the keyboard (and the markers). We were also restricted by the limitations of the processor on the EPSON BT-200 which resulted in a noticeable lag in rendering when the user moved his/her head. We are hopeful that these limitations will be resolved with advances in augmented reality device technology.

Usability: As evident from our preliminary evaluation, typing on a randomized augmented reality keyboard requires some extra time and effort from the user. As part of future work, we plan to conduct a comprehensive usability study with the help of a significant number of participants, natural typing experiments, and standard usability metrics, such as SUS [22].

VII. Conclusion

We proposed a novel technique to overcome various forms of shoulder surfing attacks against a user typing on an external physical QWERTY keyboard. Our proposal augments a randomized key layout, unknown to the adversary, over the actual QWERTY keyboard, which only the typing user can see by means of an augmented reality device. Our preliminary experimentation involving three different randomization strategies showed that keyboard randomization and augmentation does increase the time required by users to complete their typing tasks. In certain cases, it also introduced additional errors during typing. Despite its promise, these issues along with the usability of the proposed system requires further investigation.

VIII. Acknowledgment

Research reported in this publication was supported by the Division of Computer and Network Systems (CNS) of the National Science Foundation (NSF) under award number 1523960.

References

[1] V. Roth, K. Richter, and R. Freidinger, "A PIN-entry Method Resilient Against Shoulder Surfing," in *ACM CCS 2004*.
[2] T. Kwon, S. Shin, and S. Na, "Covert Attentional Shoulder Surfing: Human Adversaries Are More Powerful than Expected," *IEEE Transactions on SMC: Systems*, vol. 44, no. 6, 2014.
[3] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *USENIX Security 2009*.
[4] Y. Berger, A. Wool, and A. Yeredor, "Dictionary Attacks using Keyboard Acoustic Emanations," in *ACM CCS 2006*.
[5] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding Vibrations from Nearby Keyboards using Mobile Phone Accelerometers," in *ACM CCS 2011*.
[6] A. Maiti, O. Armbruster, M. Jadliwala, and J. He, "Smartwatch-Based Keystroke Inference Attacks and Context-Aware Protection Mechanisms," in *ACM ASIACCS 2016*.
[7] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke Recognition using WiFi Signals," in *ACM MobiCom 2015*.
[8] Software House, "Scramble Keypad SP-100," www.swhouse.com/products.
[9] 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*, vol. 5, no. 2, pp. 65–75, 2010.
[10] EPSON, "Moverio BT-200," www.epson.com/MoverioBT200.
[11] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing Shoulder-Surfing by Using Gaze-Based Password Entry," in *ACM SOUPS 2007*.
[12] Passfaces, "Two Factor Authentication - Graphical Passwords," www.realuser.com.
[13] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The Design and Analysis of Graphical Passwords," in *USENIX Security 1999*.
[14] F. Tari, A. Ozok, and S. H. Holden, "A Comparison of Perceived and Real Shoulder-Surfing Risks Between Alphanumeric and Graphical Passwords," in *ACM SOUPS 2006*.
[15] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu, "Fingerprint Attack Against Touch-enabled Devices," in *ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, 2012.
[16] A. H. Lashkari, A. Abdul Manaf, M. Masrom, and S. M. Daud, *DICTAP 2011*. Springer, ch. Security Evaluation for Graphical Password.
[17] Q. Yan, J. Han, Y. Li, J. Zhou, and R. H. Deng, "Designing Leakage-resilient Password Entry on Touchscreen Mobile Devices," in *ACM ASIACCS 2013*.
[18] D. F. Abawi, J. Bienwald, and R. Dorner, "Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit," in *ACM ISMAR 2004*.
[19] H. Kato, "Inside ARToolKit," in *1st IEEE International Workshop on Augmented Reality Toolkit*, 2007.
[20] R. A. Bailey, *Design of comparative experiments*. Cambridge University Press, 2008, vol. 25.
[21] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," *Advances in Psychology*, vol. 52, pp. 139–183, 1988.
[22] J. Brooke, "SUS - A Quick and Dirty Usability Scale," *Usability Evaluation in Industry*, 1996.