Designing Leakage-Resilient Password Entry on Head-Mounted Smart Wearable Glass Devices

Li, Yan; Cheng, Yao; Meng, Weizhi; Li, Yingjiu; Deng, Robert H.

Published in:
IEEE Transactions on Information Forensics and Security

Link to article, DOI:
10.1109/TIFS.2020.3013212

Publication date:
2020

Document Version
Peer reviewed version

Designing Leakage-Resilient Password Entry on Head-Mounted Smart Wearable Glass Devices

Yan Li, Yao Cheng, Weizhi Meng, *Senior Member, IEEE,* Yingjiu Li, and Robert H. Deng, *Fellow, IEEE*,

***Abstract*—With the boom of Augmented Reality (AR) and Virtual Reality (VR) applications, head-mounted smart wearable glass devices are becoming popular to help users access various services like E-mail freely. However, most existing password entry schemes on smart glasses rely on additional computers or mobile devices connected to smart glasses, which require users to switch between different systems and devices. This may greatly lower the practicability and usability of smart glasses. In this paper, we focus on this challenge and design three practical anti-eavesdropping password entry schemes on stand-alone smart glasses, named gTapper, gRotator and gTalker. The main idea is to break the correlation between the underlying password and the interaction observable to adversaries. In our IRB-approved user study, these schemes are found to be easy-to-use without additional hardware under various test conditions, where the participants can enter their passwords within moderate time, at high accuracy, and in various situations.**

***Index Terms*—Password Entry, Anti-Eavesdropping, Smart Glasses, Head-Mounted Device, Usability and Security.**

I. Introduction

DUE to the convenient and efficient capability of connecting individuals and cyberspace, head-mounted smart wearable devices are becoming prevalent, e.g., smart glasses. By wearing compact and lightweight smart glasses, users can access various services, such as personal email and online social network, and map services, in a hand-free manner at any place and at any time, through Augmented Reality (AR) or Virtual Reality (VR) [2], [3]. To protect these devices from unauthorized access, password-based user authentication has been pervasively used to validate users' identity. However, such authentication mechanism has intrinsic vulnerabilities, e.g., eavesdropping attackers can leak users' password by directly or indirectly observing password entry procedure via accessible channels. This kind of attack is particularly effective in practical scenarios, since smart glasses are usually used in public areas and outdoors that are vulnerable to password leakage.

A preliminary version of this paper appears in the Proceedings of AsiaCCS as a short paper, pp. 327-333, 2017 [1].

Y. Li is with the School of Cyber Engineering, Xidian University, China. E-mail: yan.li.2009@smu.edu.sg

Y. Cheng is with the School of Information Systems, Singapore Management University, Singapore. E-mail: ycheng@smu.edu.sg

W. Meng (Corresponding author) is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark. E-mail: weme@dtu.dk

Y. Li is with the Department of Computer and Information Science, University of Oregon, U.S.A. E-mail: yingjiul@uoregon.edu

R. H. Deng is with the School of Information Systems, Singapore Management University, Singapore. E-mail: robertdeng@smu.edu.sg

In order to thwart the threat of eavesdropping attacks, prior research focuses on improving anti-eavesdropping password entry on desktops, laptops, smartphones and tablets [4], [5], [6]. However, most of them suffered from both security and usability issues (i.e., some schemes may need up to 221 seconds per login attempt) [7], [6], [8]. To achieve better security and usability, it is necessary to use a protected environment to hide certain user interaction during the password entry [7], [9]. While this solution is not suitable to protect password entry on smart glasses due to the following reasons. Firstly, the traditional input equipment like keyboard and touch screen is not available on the smart glasses according to the compact and lightweight design. Secondly, smart glasses have a much smaller screen than the PC monitors and phone screens. Then, smart glasses have limited hardware support.

Motivations. Most existing password entry schemes on smart glasses require users to input their passwords by connecting the glasses with additional PCs or mobile devices, whereas the additional devices may not be always available or accessible in certain scenarios, especially in public places and outdoors. In this case, users may need to switch between smart glasses and mobile devices for password entry. Such interrupted user experience may lead to more inputting errors, and even raise users' stress and anxiety [10] when they perform important tasks like AR-based or VR-based payment [11]. Moreover, *Near-Eye-Display (NED)* screens have also been exploited on Google Glass, which is a tiny optical instrument to reflect and magnify the display to users' eyes [12]. It is found that the NED screen can help solve part of password leakage, but is still hard to fully protect the password entry [13]. As a result, how to design a secure and usable anti-eavesdropping password entry scheme for smart glasses remains a challenge.

Contributions. In this paper, we focus on this challenge and design three anti-eavesdropping password entry schemes for smart glasses, named *gTapper*, *gRotator* and *gTalker*. As the NED screen has been widely deployed in smart glasses, it plays an important role in our scheme design. There are several features of a typical NED screen. 1) In order to present clear and sharp display to users, the NED screen is head-mounted. It is fixed on the smart glass frame and placed physically close to the users' eyes [12]. 2) Thanks to its compact size and physical proximity to the users' eyes, the NED screen can privately display information to users without being observed by others. This characteristic can be used to deliver hidden information, breaking the correlation between the underlying password and the interaction observable to an adversary.

Due to these features, our schemes do not require any

additional hardware or external devices other than a touch pad, a gyroscope, and a microphone, which are commonly available on smart glasses. To enter password, our schemes require users to perform simple gestures on the touch pad, slightly rotate head, or speak numbers based on the hidden information displayed on the NED screen of smart glasses. In the evaluation, we implemented the proposed schemes on Google Glass, which is a popular commercial smart glasses, and conducted a user study with 57 participants. To simulate common users' daily usage of password entry, our user study considers various practical conditions in relation to *normal entry*, *time pressure* and *distraction*. The contributions can be summarized as follows.

- We focus on the security of smart glasses and design three password entry schemes to defend against eavesdropping attacks. Our goal is to ensure that no password information except password length might be leaked, no matter how powerful an attacker is and how many password entry sessions the attacker can observe during an eavesdropping attack.
- We implement our schemes on Google Glass to accommodate different users' preferences. The designed schemes do not need any additional hardware other than a touch pad, a gyroscope, and a microphone, which are commonly available on smart glasses. In this case, our schemes have a big potential to be implemented on commodity smart glasses in practice.
- In the evaluation, we conduct a user study to evaluate the scheme usability by considering multiple test conditions in practical scenarios. Experimental results with different time pressure and distraction levels indicate that our schemes can provide better security with good usability as compared to existing schemes.

The remaining parts of this paper are organized as follows. Section II summarizes related work on eavesdropping attacks, design principles, and entry schemes. In Section III, we introduce the basic functionalities of smart glasses, and discuss the issue of password leakage under eavesdropping attacks. Section IV describes our designed anti-eavesdropping password entry schemes on smart glasses: *gTapper*, *gRotator* and *gTalker*. Section V makes a user study to evaluate the scheme performance in the aspects of security and usability. Section VI discusses password length and points out limitations of our work. Finally, we conclude our work in Section VII.

II. Related work

In this section, we introduce related work regarding anti-eavesdropping password entry, including eavesdropping attacks, design principles, and entry schemes.

Eavesdropping attacks. This type of attacks against password entry can be categorized into *external eavesdropping* and *internal eavesdropping*. Depending on the exploitable attack channels, the former can be further classified into vision-based attack, motion-based attack, and acoustics-based attack. 1) Under the vision-based attack, an adversary can directly view or record videos about a victim's password entry and then infer the victim's password by analyzing various clues in the video [14], [15], [16]. 2) Under the motion-based attack, an adversary may attack remotely by accessing arm-mounted motion sensors equipped on a smart wearable device [17], [18], [19]. For instance, Liu et al. [17] explored such threat by showing the feasibility of inferring users' PINs and typed texts through sensor data on smart watch. Wang et al. [18] further demonstrated that motion data from wrist-worn devices can be used to distinguish mm-level distances in users' hand movements during password input. 3) Under the acoustics-based attack, an adversary can record audio signals about password entry and infer a victim's password by analyzing the ringtones and keystroke acoustics from the recorded audio [20].

On the other hand, internal eavesdropping attacks can be further classified into *unprivileged attacks* and *privileged attacks*. The first type of attacks can be launched by an adversary based on unprivileged access to password entry data. As a typical example, as the motion sensor data on smart devices could be accessed by any applications, an adversary may recover a victim's movements during the password entry process and then find the underlying password [21]. By contrast, the second type of attacks allows an adversary to reach the internal memory of a victim's device via malware, logic key logger, and network traffic monitor [22]. It is worth noting that our designed schemes can be used to mitigate the unprivileged attacks, while may still be subject to the privileged attacks that can be controlled by other security mechanisms and proper configuration of operating systems [23].

Design principles. Different design principles have been proposed for designing anti-eavesdropping password entry schemes. On one hand, Schaub et al. [24] analyzed how the virtual keyboard UI design on mobile devices affects the effectiveness of the vision-based eavesdropping attacks and suggested to alter the virtual keyboard UI appearance to lower the success rate of the vision-based eavesdropping attacks. However, the suggestions cannot perfectly prevent the eavesdropping attacks. Forget et al. [25] proposed a graphical password based scheme with cued-recall nature to help users memorize passwords and captured eye gaze as input methods, which claimed to be better than text password and typing method. Unfortunately, the proposed design is slow in login process (53.5 seconds) and is not suitable for smart glass with tiny screen and no camera tracking eyes.

On the other hand, Roth et al. [26] introduced an approach of using a cognitive trapdoor game to transform the knowledge of underlying password into obfuscated responses for password entry. Li et al. [6] pointed out several design principles, including time-variant responses, uncertainty, and balance. Later, Yan et al. [9] introduced the design principles against brute force attacks and generic statistical attacks. The three research studies above indicate that it is necessary to use certain secure channel between a user and the device during password entry to achieve provable security and high usability. Horcher et al. [27] suggested that the optimized size and design layout of virtual keyboards, and the consideration of security, usability and anticipatory behavior, can help prevent the eavesdropping attacks. This work follows these principles in designing our schemes.

Entry schemes. Many user authentication schemes have been proposed to resist against eavesdropping attacks. Ginzburg et al. [28] proposed an authentication scheme that can verify users via random challenges, but the scheme workload is high as users have to memorize a formula for authentication. Weinshal [5] designed a CAS scheme based on the cognitive capability of human beings. It requires a user to identify around 30 secret pictures and find out a path among 80 pictures randomly displayed on the screen in only one single round. According to the analysis in [5], CAS may expose a high usability cost, which may take up to 221 seconds per login attempt. Moreover, it requires 10 rounds of authentication attempts to mitigate brute-force attacks. Then, some eye-gaze based schemes were proposed for inputting passwords [29]. Unfortunately, such schemes require extra hardware-based eye-tracking tools that are often not available on existing smart glasses.

Li et al. [9] proposed a scheme, called CoverPad, to protect the password entry process on mobile phones and tablets with acceptable usability. It leverages a temporary secure channel between user and touch screen in order to transform a password during the password entry. This aims to prevent an adversary from inferring any information by monitoring a user's inputting behaviour. Due to the compact and lightweight design of smart glasses, it is difficult to apply the existing anti-eavesdropping password entry schemes to smart glasses. Li et al. [1] designed a series of anti-eavesdropping password entry schemes for smart glasses and presented preliminary results of the scheme performance. This work significantly extends the above work in four major aspects. Firstly, an important internal eavesdropping adversary model is introduced and analyzed. Secondly, the experiments involve distraction levels, multiple modalities, and six test cases that are important to evaluate the usability and practicability of our schemes in real-world scenarios. Thirdly, more comprehensive and thorough evaluation and analysis are provided. Fourthly, a more thorough literature review is presented.

III. Preliminaries

Smart glasses have attracted much attention from both academia and industry. In this section, we briefly introduce the basic design and functionalities of smart glasses, and then analyze the issue of password leakage under eavesdropping attacks.

A. Background

Smart glasses are emerging smart wearable devices and play an important role in both AR and VR applications. There are several commercialized products available in current market, such as Google Glass made by Google and HoloLens made by Microsoft [2], [3]. Generally, smart glasses can adopt different operating systems, like Android Wear in Google Glass and Windows Holographic in HoloLens. Users can install various applications and access a set of services, i.e., checking personal Emails, chatting with friends online, searching geographic information on personalized digital map, and so on. A typical smart glasses is usually equipped with a tiny head-mounted NED screen and multiple sensors to support user interaction and improve user experience, i.e., collecting the information about users and environments. The sensors may include a small touch pad, a gyroscope and a microphone, while the NED screen allows users to easily interact with the smart glasses. Due to the compact and lightweight design, traditional input equipments like keyboard and touch screen are not available on the smart glasses. This is because these traditional equipments are too heavy or too big to deploy on smart glasses.

In this work, we adopt Google Glass Explorer Edition (XE) 2 (with Android Glass OS) [12] as the main platform during our implementation and evaluation, which is one type of popular commercialized smart glasses products in current market. It is powered by a variation of common features and functionalities of smart glasses, and provides Android-based programmable API. More specifically, Google Glass is equipped with a 0.5-inch NED screen (0.75 inch in length and 0.375 inch in width) located in front of a user's right eye (see Figure 1). The tiny NED screen is designed to reflect and magnify the display view for users. The screen is placed physically close to the right eye for a clearer and sharper view (approximately 1 inch between the NED screen and the right eye) [12].

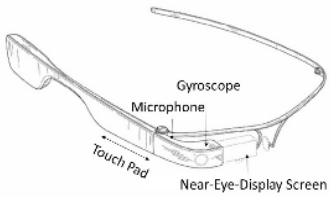


Fig. 1. The design of Google Glass.

As mentioned earlier, multiple sensors are embedded on the right side of Google Glass frame to facilitate users' operations, including a touch pad, a gyroscope, and a microphone. In particular, the touch pad with 3.25-inch long enables gesture-based user interactions on Google Glass, supporting simple gestures like finger tap and finger swipe [12]. As a kind of head-mounted device, Google Glass also tracks the user's head movement and reacts based on the data collected from the gyroscope. For example, users can select items from a menu list on Google Glass by rotating his/her head. In addition, as Google Glass supports speech recognition-based user interaction, users can send voice commands via the microphone directly.

B. Eavesdropping Attacks Against Password Entry

Up to date, traditional password-based user authentication is still widely adopted; however, this kind of authentication has intrinsic limitations, i.e., vulnerable to *eavesdropping attacks*, where an adversary may disclose or infer a victim's password by observing and analyzing the password entry process.

Because of the compact size of smart glasses, current commercialized smart glasses products, like Google Glass and HoloLens, always require users to enter their plaintext passwords through additional trusted PCs with keyboards or trusted mobile phones/tablets with touch-screens. For this purpose,

the PC and the mobile phone/tablet must be connected to the smart glasses during the whole password entry process. This makes the password entry on smart glasses especially suffers from password leakage, as users' passwords could be captured by various eavesdropping tools such as recording camera, malware, and key logger. Based on what kind of information can be accessed by an adversary, eavesdropping attacks on smart glasses can be categorized as *external* and *internal* eavesdropping.

External eavesdropping. Regarding the first type of eavesdropping attacks, an adversary can exploit channels outside smart glasses to infer victim's password. Based on the specified channels, such attacks can be further classified into *vision*-based attacks, *motion*-based attacks, and *acoustics*-based attacks.

- Under the vision-based attacks, an adversary is able to directly watch or video-tape a victim's password entry process, where all the finger movements, gestures, and head movements can be observed. Then attackers may infer the victim's password by analyzing the observed movements. It is worth noting that suck type of attacks does not necessarily need physical proximity between the adversary and the victim, because video surveillance systems are widely deployed in various public places and most of them are even connected to the Internet.
- Under the motion-based attacks, an adversary can estimate and track a victim's movements like finger movements and arm movements by means of additional equipments during the password entry. These estimated movements can facilitate an adversary to infer users' passwords. More importantly, attackers can remotely launch such attacks without any requirement on the physical proximity to the victim. For instance, smart watches, e.g., Samsung Galaxy Gear and Apple Watch [30] are particularly vulnerable to this kind of attacks, because they already provide built-in motion sensors such as gyroscopes and accelerometers.
- Under the acoustics-based attacks, an adversary can capture audio signals during the password entry process. Traditionally, this type of attacks relies on whether users' key pressing actions can be distinguished by tone patterns, i.e., on an old-fashioned phone. However, as speech recognition has become much popular and prevalent on mobile phones and smart glasses, an adversary could have more opportunities to conduct effective acoustics-based attacks when users play voice commands during the password entry.

Internal eavesdropping. As compared to the external eavesdropping, internal eavesdropping is more powerful, since an adversary can exploit internal channels and access internal states of smart glasses. Based on whether the specific privileges are declared, there are mainly two types of internal eavesdropping: *unprivileged* and *privileged*. The former is most likely to happen when the sensor data on the smart glasses can be accessed by an adversary without requiring any privileges directly. Attackers can estimate a victim's movements and infer the input passwords during the password entry by analyzing the available sensor data. For example, motion sensors like gyroscope and accelerometer are typical sensors, which are accessible by applications and background services without declaring specific permissions on Android [31]. Thus, it opens a hole for unprivileged malware to collect necessary motion sensor data and infer passwords.

IV. Design Overview

In this section, we introduce our design goals and three anti-eavesdropping password entry schemes: gTapper, gRotator and gTalker on smart glasses.

A. Design Goals

To design practical anti-eavesdropping of password entry on smart glasses, it is important that the schemes do not need to rely on any additional devices, which may not be always available in a real environment. Even if these devices are available, they may not resist against eavesdropping attacks. In addition to retaining most benefits of traditional passwords, our design goals can be explained from security, practicability, and usability perspectives.

- In order to protect password-based user authentication on smart glasses, a desirable scheme should minimize the threat of password leakage during the whole password entry process. In practice, smart glasses can be often used in various environments, especially public areas, an adversary has a big opportunity to steal passwords, including external eavesdropping and unprivileged internal eavesdropping as described in Section III-B. The adversary can infer the credentials by observing the entry process and analyzing the correlation between the observed information and underlying password. As a result, it is important to decouple the link in-between.
- Then, a desirable scheme should be pervasively accessible on smart glasses in practical settings. For this purpose, it is preferable no additional devices or external to be involved, as these devices or hardware may not be always available or accessible in a real scenario like outdoor and public areas. The scheme is expected to be constructed by using the built-in hardware and functionalities, which are commonly available on existing commodity smart glasses.
- A desirable scheme should provide good usability, i.e, preserving the benefits of legacy password such as nothing-to-carry and easy-to-use features [32]. Therefore, intuitive and simple operations are preferable for users during the password entry, especially in inconvenient environments, e.g., crowded places.

B. Design of Password Entry Schemes

Most existing smart glasses have typical built-in hardware and sensors, including an NED screen, a touch pad, a gyroscope, and a microphone. As mentioned in Section III-A, because of the tiny size and physical proximity to a user's eye, it is difficult for an adversary, even a hidden camera, to access the channel between the NED screen and the users without

causing awareness. Thus, our designed schemes can use the NED screen to display information privately to a user without being noticed. In addition, our schemes adopt a set of simple and typical interaction operations on smart glasses, including finger gestures, head movements and human voice by means of the touch pad, gyroscope, and microphone. Overall, we take advantage of these hardware and interaction channels to design our schemes on smart glasses.

Regarding our design, we assume a server and a user agree on a n-length password $pwd = (p_1, p_2, ..., p_n)$. For each $i \in \{1, 2, ..., n\}$ in an authentication process, a hidden random keypad $\Gamma_i(\cdot)$ is privately displayed to the user through the NED screen on smart glasses during the process of password entry. The hidden random keypad $\Gamma_i(\cdot)$ defines a random mapping $\Omega \rightarrow \Phi$ where Ω is the set of all elements contained in the password alphabet and Φ is the set of all candidate user operations via a user-device interaction channel. Note that in each round i, a new random mapping $\Gamma_i(\cdot)$ (i.e., a new hidden keypad) is drawn from the universal set of the candidate mappings $\Omega \rightarrow \Phi$ following uniform distribution. Thus, given the hidden keypad $\Gamma_i(\cdot)$, users have to perform corresponding operations $op_i = \Gamma_i(p_i)$ via the interaction channel in order to select the correct underlying password element p_i in pwd.

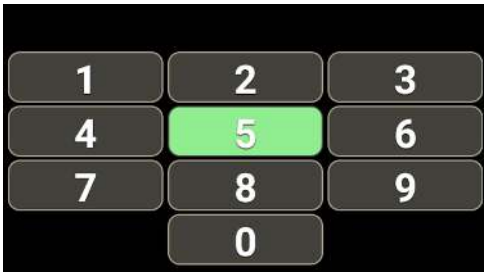
The observable response operation op_i by the user for the same password element is uniformly randomized due to the hidden random keypad $\Gamma_i(\cdot)$. Therefore, as long as $\Gamma_i(\cdot)$ is not disclosed, an adversary cannot infer any useful information from op_i to discover the underlying password element p_i, even through external eavesdropping and unprivileged internal eavesdropping attacks. As the hidden keypad is privately delivered to the user via the NED screen, it is difficult for adversaries to compromise this delivery channel. The detailed security analysis will be discussed later in this section.

During the password entry process, different users may have various response operations and hidden random mappings. According to the specific interaction channels, we design and implement three anti-eavesdropping password entry schemes, named as gTapper, gRotator, and gTalker.

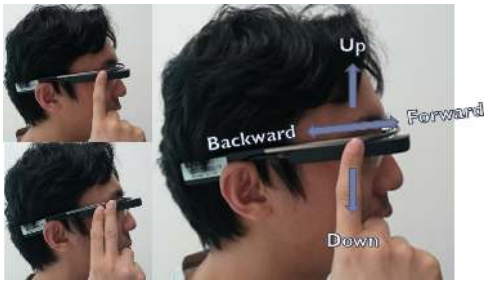
C. gTapper

This scheme of gTapper is designed based on the small touch-pad that is widely available on most smart glasses [12]. The pad accepts users' finger gestures as input signals, including tapping, pressing, and swiping with different fingers towards various directions.

Typically, this scheme adopts the password alphabet Ω to be comprised of all single-digit numbers from 0 to 9. To implement gTapper, the hidden keypad contains 10 numbers as shown in Figure 2(a). In each round i, gTapper randomly selects a number $s_i \in \{0, 1, 2, ..., 9\}$ and sets the focus on that number s_i, i.e., Figure 2(a) shows that the number of 5 is focused in the hidden keypad. It is worth noting that users can locate the keys easily and swiftly, as the key locations would not change. To change the number focus in either descending or ascending order, users have to use one finger to swipe forward or backward on the touch pad, as shown in Figure 2(b). To summarize, users can use one finger to shift the number focus to $(s_i - 1)$ mod 10 or $(s_i + 1)$ mod 10, by swiping forward once or by swiping backward once. In this scheme, we only consider two intuitive gestures to shift focus like swiping forward/backward in order to reduce users' mental preparation and workload [33], which is also examined and observed in our pilot study.



(a) Demonstration of gTapper.



(b) The top left figure shows one-finger operation. The bottom left figure shows two-finger operations. Right figure shows swiping gestures in the four directions.

Fig. 2. Demonstration of gTapper and gestures

To enter a password element $p_i \in \{0, 1, 2, ..., 9\}$ in round i, a user has to shift the number focus to p_i on the keypad from the initially focused number s_i by swiping forward or backward for op_i times, where $op_i = (s_i - p_i)$ mod 10 or $op_i = (p_i - s_i)$ mod 10 respectively. Then the user can enter the selected number p_i with a one-finger tap on the touch pad. Taking Figure 2(a) as an example, if given an underlying password element 7 and a hidden keypad in which the focused number is set to 5, a user can select and enter this password element 7 on the hidden keypad, by swiping backward with one finger on the touch-pad for $2 = 7 - 5$ times.

Security Analysis of gTapper. During the user's password entry, as attackers can directly or indirectly observe the user's operations in each round i through external eavesdropping attacks and unprivileged internal eavesdropping attacks, they can know user operations on gTapper including swiping forward/backward on the touch pad and the number of swiping operations. Based on the observation in round i, attackers can know the number and the directions of shifts from the initially focused number to the i-th element of the password. However, since the hidden keypad is protected, it is hard for attackers to know what the initially focused number is and therefore cannot infer the i-th element of the password.

Proof: Given the user operation op_i in any round i, the initially focused number s_i, and any two elements p_x and p_y in a w-sized password alphabet (password alphabet $\{0, 1, 2, ..., 9\}$ with $w = 10$), let $Pr(op_i|p_x)$ and $Pr(op_i|p_y)$ be the probabilities for the operation op_i when the underlying password

elements are p_x and p_y, respectively. Thus, if the observed user operation is swiping forward, we have $Pr(op_i|p_x) = Pr(op_i = s_i - p_x \bmod w) = Pr(s_i = p_x + op_i \bmod w) = Pr(s_i = C) = 1/w = Pr(op_i|p_y)$ for any i, x, and y, while if the observed user operation is swiping backward, we have $Pr(op_i|p_x) = Pr(op_i = p_x - s_i \bmod w) = Pr(s_i = p_x - op_i \bmod w) = Pr(s_i = C) = 1/w = Pr(op_i|p_y)$ for any i, x, and y, where C can be any integer randomly drawn from $\{0, 1, 2, ..., 9\}$. The sequence of user operations observed by an adversary is equivalent to a random sequence. The adversary cannot distinguish the i-th element in the underlying password between any two elements in the password alphabet. □

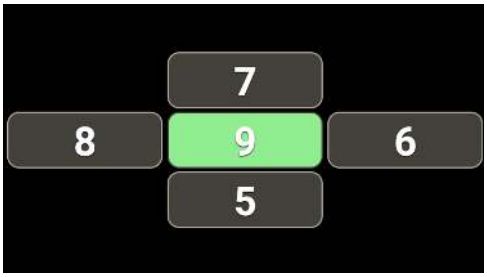
D. gRotator

The design of gRotator relies on a gyroscope, which is widely available on existing smart glasses for detecting and tracking users' head motions, like head rotations. In this case, gRotator allows users to select and enter password elements via head rotation movements.

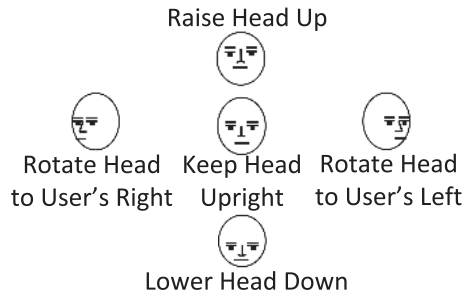
Similar to the first scheme, gRotator adopts the alphabet of password Ω to have all single-digit numbers from 0 to 9. The hidden keypad of gRotator is comprised of two number screens: a small number screen C_s and a big number screen C_b, where $C_s \subset \Omega$, $C_b \subset \Omega$, $C_s \cup C_b = \Omega$, and $c_s < c_b$ for any $c_s \in C_s$ and $c_b \in C_b$. In our implementation, we set the two number screens as $C_s = \{0, 1, 2, 3, 4\}$ and $C_b = \{5, 6, 7, 8, 9\}$. At any time, only one number screen would be displayed for the sake of the limited size of NED screen and the inaccurate control of head movements by human beings.

Figure 3(a) shows the big number screen in our implementation. In each round i, five numbers and their positions would be randomly shuffled. One number screen would be randomly displayed as the initial screen under the uniform distribution. To change the number screen, users have to swipe forward with one finger on the touch-pad, if the i-th underlying password element is not included in the displayed number screen. To select a number, users have to rotate his or her head according to the number position on the screen. As shown in Figure 3(b), users may need to select a number located at top, at bottom, on the left, on the right, or in the center by raising head, lowering head, rotating head towards left, rotating head towards right, or heading upright, respectively. Taking Figure 3(a) as an example, a user needs to raise his or her head up, as the number of 7 is located at the top of the displayed screen.

In order to determine and track users' head movements, we can estimate the movements based on the motion data captured by the gyroscope, including angular speeds on three orthogonal axes (i.e., axis X, axis Y, and axis Z) from the motion sensor coordinate system [2]. With the estimation of head rotations, users can easily input password elements by performing corresponding head rotations. Figure 4 depicts the typical motion sensor coordinate system on the NED screen. Generally, axis X is horizontal that points to the right; axis Y is vertical that points to the up; and axis Z points toward a user's face. In terms of the angular speed, we can estimate users' head rotation using a dead-reckoning algorithm [34]. Let $R_{t_i} = (r_{x,t_i}, r_{y,t_i}, r_{z,t_i})$ be the angular speed generated by the



(a) Demonstration of gRotator.



(b) Head movements in gRotator

Fig. 3. Demonstration of gRotator and head movements

gyroscope at time t_i. The rotation angle along each axis can be calculated by the trapezoidal rule for integral approximation as follows.

$$\theta_{s,t_i} = (r_{s,t_{i-1}} + r_{s,t_i}) \cdot (t_i - t_{i-1})/2 \quad (1)$$

where $s \in \{x, y, z\}$. Note that $(\theta_{x,t_i}, \theta_{y,t_i}, \theta_{z,t_i})$ is also called as Cardan angles in 3D coordinate system [35]. For simplicity, we use angle θ_{x,t_i} and angle θ_{y,t_i} to determine the up/down directions and left/right directions of head movements starting from an initial head pose, which is defined as the user's frontal face head pose. This pose can be calibrated and set at the moment when a user initially launches gRotator or when a user taps on the touch-pad with two fingers together.

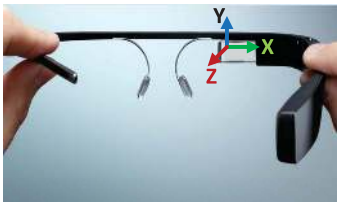


Fig. 4. A typical motion sensor coordinate system on smart glasses.

In practice, users may easily cause abrupt changes to affect the estimation of head rotation directions, due to the inaccurate control of head poses [36]. For this issue, we apply thresholds ξ_v and ξ_h for up/down direction and left/right direction, respectively. Thus, the estimation of head rotation direction H_{t_i} at time t_i can be computed as below.

$$H_{t_i} = \begin{cases} \text{up} & \theta_{x,t_i} \le (-1) \cdot \xi_v \ and \ |\theta_{y,t_i}| < \xi_h \\ \text{down} & \theta_{x,t_i} \ge \xi_v \ and \ |\theta_{y,t_i}| < \xi_h \\ \text{left} & \theta_{y,t_i} \ge \xi_h \ and \ |\theta_{x,t_i}| < \xi_v \\ \text{right} & \theta_{y,t_i} \le (-1) \cdot \xi_h \ and \ |\theta_{x,t_i}| < \xi_v \\ \text{upright} & |\theta_{x,t_i}| < \xi_v \ and \ |\theta_{y,t_i}| < \xi_h \end{cases} \quad (2)$$

where $\xi_v \geq 0$ and $\xi_h \geq 0$. It is worth noting that θ_{x,t_i} and θ_{y,t_i} would be negative values if the rotation directions of axis X and axis Y are counter-clockwise [2]. According to [33], the best performance for determining the head rotation directions can be achieved at $\xi_v = 15°$ and $\xi_h = 25°$, which is also examined in our pilot study.

Security Analysis of gRotator. As attackers can observe user operations like swiping on the touch pad and head rotations, they may know in each round whether the user changes the number screen displayed initially and know the exact positions of the underlying password elements located in the displayed number screen. However, as long as the hidden keypad, including the two number screens, is not disclosed, the adversary would not know which number screen is chosen by the user nor the mapping between the 5 numbers and the positions in the displayed screen. Therefore, the adversary cannot infer any element of the underlying password.

Proof: Given the user operation op_i in any round i and any two elements p_x and p_y in 10-sized password alphabet $\{0, 1, 2, ..., 9\}$, let $Pr(op_i|p_x)$ and $Pr(op_i|p_y)$ be the probabilities for the operation op_i when the underlying password elements are p_x and p_y, respectively. Based on the design of gRotator, one of the two number screens C_s and C_b is randomly drawn from a uniform distribution and displayed initially (i.e., with a probability of $\frac{1}{2}$). Each number screen contains 5 numbers whose positions are randomly shuffled. Thus we have $Pr(op_i|p_x) = \frac{1}{2} \cdot Pr(p_x \in direction\ of\ op_i) = \frac{1}{2} \cdot P_4^4/P_5^5 = \frac{1}{2} \cdot 4!/5! = \frac{1}{10} = Pr(op_i|p_y)$ for any i, x, and y. Thus an adversary gains no advantage for distinguishing the i-th element in the underlying password between any two elements in the password alphabet by observing users' operations. □

E. gTalker

The design of gTalker depends on a speech recognition-enabled built-in microphone, which can take a user's speech as input. With the microphone, smart glasses can recognize and react to the speech content.

For implementation, gTalker adopts the alphabet of password as $\Omega = \{0, 1, 2, ..., 9\}$. Figure 5 shows the hidden keypad's layout of gTalker, where every *white number* p is followed by an underlined *red number* s. The hidden keypad consists of two keypads, one original keypad with all white numbers and one transformed keypad with all underlined red numbers. In each round i, white numbers would remain their positions while underlined red numbers would shuffle their positions randomly. For each white number $p_k = k$, let s_{ik} denote the corresponding underlined red number in round i, where $k \in \Omega$ and $s_{ik} \in \Omega$. For $\forall j, k \in \Omega$ and $j \neq k$, $s_{ij} \neq s_{ik}$ holds.

To enter an underlying password element k, users have to firstly identify the position of a white number p_k, and then speak out the underlined red number s_{ik}. For authentication, gTalker should receive the right p_k and recognize the correct number s_{ik} said by the user. It is worth noting that the mapping relationship between the original keypad and the transformed keypad would not the same in each round. For example, as shown in Figure 5, if given an underlying password element 7 and the hidden keypad, a user has to select and enter the password element 7 and speak out 6 for authentication (note that the white number is 7 while the followed red number is 6 in the hidden keypad).

Fig. 5. Demonstration of gTalker.

To recognize a user's input, gTalker uses an offline speech recognition function available in Android API [37], which is developed based on deep neural networks with hidden Markov models (DNN-HMM) [38], [39]. This speech recognition function is selected for gTalker, due to its speaker-independence and low word error rate (WER) [38].

Security Analysis of gTalker. For gTalker, an adversary may know the number spoken by the user in each round i. However, as long as the transformed keypad is not disclosed, the adversary does not know the random mapping between the original keypad and the transformed keypad. Therefore, the adversary cannot infer the i-th element of the underlying password in each round i.

Proof: Given the number s_{ik} spoken by the user in any round i and any two elements p_x and p_y in a w-sized password alphabet ($w = 10$ in our implementation), let $Pr(s_{ik}|p_x)$ and $Pr(s_{ik}|p_y)$ denote the probabilities of observing a number s_{ik} when the underlying password elements are p_x and p_y, respectively. Because the original keypad remains unchanged while the transformed keypad randomly shuffles in each round, we have $Pr(s_{ik}|p_x) = P_{w-1}^{w-1}/P_w^w = (w-1)!/w! = 1/w = Pr(s_{ik}|p_y)$ for all i, x, and y. Thus an attacker gains no advantage for distinguishing the i-th element in the underlying password between any two elements in the password alphabet by observing the number spoken by the user. □

V. Data Collection and Evaluation

A. Data Collection and Methodology

To evaluate the scheme performance, we got an IRB approval and recruited up to 57 participants from our university via recruiting emails, including 29 males and 28 females aged between 19 and 28. It is worth noting that a numerical identifier is assigned to each participant to protect their privacy. In the study, each participant has to spend around 60 minutes in a quiet room and is paid with 10 dollars as compensation. In particular, the study contains three major parts, which are developed based on a within-subjects design [40]. After completing each part, participants can have a short break for 1-3 minutes before they move to the next part. Prior to the main user study, we conducted a pilot study among 10 internal users in order to validate the design of the user study procedures and select proper experimental settings and parameters. The details of each part in the main user study are described as below.

- In the first part, we briefly explain the purpose of our study to each participant, and introduce how to use the designed schemes, including gTapper, gRotator, and gTalker on Google Glass. More specifically, we provide each participant with Google Glass and teach them how to use in an interactive step-by-step manner to make sure that every participant understands how to perform the experiments.
- In the second part, each participant is requested to use gTapper, gRotator, and gTalker as three *test groups*. To avoid the learning effect on the scheme performance, all schemes are assigned to each participant in a random sequence. In each test group, participants have to remember a password randomly generated at the beginning and the same password would be used in the same test group. The password is a 6-digit PIN, which has been commonly employed in most real-world scenarios, e.g., online banking services. If a participant forgets the assigned password, they can use a 'show the password' function by swiping up with one finger on the touch pad.
- In the last part, each participant is asked to give feedback via an online questionnaire with a 5-point Likert scale. The online questionnaire contains 15 questions and collects users' knowledge background and previous experience of smart devices, perception of the security of user authentication, and perception and attitudes towards our designed password entry schemes.

More specifically, in each test group, we adopted six *test conditions* to evaluate the impact of *time pressure* and *distraction* during the password entry. These test conditions are used to simulate a common and practical usage scenario when users input their passwords. In practice, users may need to log into a system/service emergently and complete the password entry process within a time limit (time pressure-related conditions), or users are interrupted and respond to tasks in emergency or higher priority during the password entry (distraction-related conditions). For example, a user may suspend the password entry and rotate his/her head to talk to his/her colleague when it is necessary. The detailed test conditions are discussed next.

B. Test Conditions

To simulate a practical scenario, we employed a timer and some secondary tasks in particular tests. The *timer* is designed to give participants time pressure by showing how much time left for the existing test. In a test, the timer was implemented with a text field and displayed a countdown number in second, at the top left corner on the NED screen. The *secondary tasks* aim to simulate conditions related to unexpected distraction. In the study, we adopted *multiple modality* presented secondary tasks, which are often used in research fields, e.g., experimental psychology [41], [42]. The goal of using multiple modality presented secondary tasks is to investigate the influence of modality switch in dual-task experiments [41].

With these secondary tasks, the primary task can be evaluated based on the presence of different modality presentation in the secondary tasks. The important modalities include a linguistic modality and a spatial modality, which conduct linguistic/spatial processes [41], [42]. We select these two modalities in our secondary tasks, because they are common in real-world scenarios, such as immediately responding to chatting requests from others during the password entry. In our tests, a participant is either required to speak out a displayed number in a secondary task with the linguistic modality, or required to rotate his/her head according to a displayed direction in a secondary task with the spatial modality.

In the study, we consider the following two conditions to test the impact of time pressure: normal condition and timed condition.

- **Normal condition:** participants are required to minimize their failure rate in a fixed number of login attempts without enforced time limitation. By considering a common scenario, we set the number of login attempts as 3.
- **Timed condition:** participants are required to reach as many successful logins as possible within a fixed time period. In the tests, by considering the usability, we set the time limit for gTapper and gRotator as one minute but increase the time limit for gTalker to two minutes. The time limits are carefully selected based on the users' feedback and data quality requirements in our pilot study in order to avoid the users' uncomfortable experiences and insufficient data in the experiments.

For secondary tasks, we define the following distraction levels that were used in the distraction-related test conditions.

- **No distraction:** participants only need to perform the login task.
- **Distraction:** when a participant performs the login task, a secondary task may appear with a 1/3 probability each time asking the participant to complete a single element entry. In particular, either a speaking number task or a head rotation task would be selected as the secondary task with a 1/2 probability. This aims to simulate a practical unexpected distraction during the password entry.
- **Heavy distraction:** when a participant performs the login task, a secondary task would appear each time asking the participant to finish a single element entry. Similar to the above Distraction condition, either a speaking number task or a head rotation task would be selected as the secondary task with a 1/2 probability.

To summarize, we have 6 test conditions in each test group through combining both time conditions and distraction levels: normal condition with no distraction, normal condition with distraction, normal condition with heavy distraction, timed condition with no distraction, timed condition with distraction, and timed condition with heavy distraction. For simplicity, in the rest of this paper, we denote the normal condition with no distraction and the timed condition with no distraction as *normal condition* and *timed condition*, respectively. Further, to minimize the learning effect, each test group would encounter these tests randomly. In this work, we have the following eight focused questions for investigation in the experiments. The questions investigate how the different conditions and distractions impact the performance of the proposed schemes. Since the conditions and distractions in the experiments are

chosen to simulate users' usage in real-world scenarios, the answers to the questions help understand the usability and practicability of the proposed schemes.

- **Q1:** Compared to *normal condition*, is the *login time* significantly shorter under *timed condition*?
- **Q2:** Compared to *normal condition*, is the *login accuracy* significantly lower under *timed condition*?
- **Q3:** Compared to *normal condition*, is the *login time* significantly longer if *distraction* is present?
- **Q4:** Compared to *normal condition*, is the *login accuracy* significantly lower if *distraction* is present?
- **Q5:** Compared to *normal condition*, is the *login time* significantly longer if *heavy distraction* is present?
- **Q6:** Compared to *normal condition*, is the *login accuracy* significantly lower if *heavy distraction* is present?
- **Q7:** Compared to *normal condition with (heavy) distraction*, is the *login time* significantly shorter under *timed condition with (heavy) distraction*?
- **Q8:** Compared to *normal condition with (heavy) distraction*, is the *login accuracy* significantly lower if *timed condition with (heavy) distraction* is present?

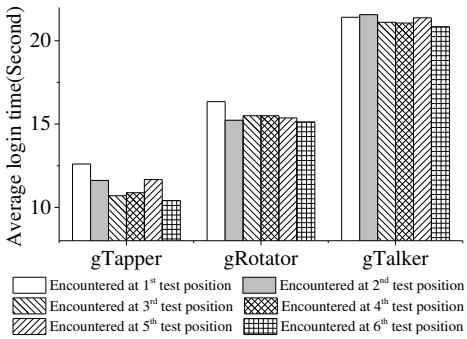
C. Experimental Results

In this work, we employ two important metrics to evaluate the scheme performance: *average login time* and *login success rate*. The former is used to evaluate the speed of a login process while the latter is used to evaluate the accuracy of login attempts. We further apply statistical analysis to measure the significance of our experimental results. In particular, we run an omnibus test [43] across the test conditions for each designed scheme, and used Kruskal-Wallis test [44] to investigate the significant differences as compared to other test conditions.

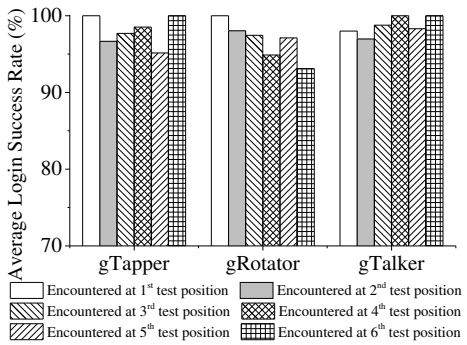
1) Learning Curve: In our schemes, we use simple and intuitive operations to reduce the difficulty level of learning [45]. To evaluate the performance, Figure 6 compares the participants' performance under the normal condition where the 6 tests may appear at different positions in each test group. It is found that participants spent more login time for the tests appearing at the first position than the other positions; however, the results are not significantly different, i.e., it is 2.2 seconds, 1.1 seconds, and 1 second for gTapper, gRotator, and gTalker, respectively. In addition, it is found that changing the test positions would not significantly affect the login success rates among different schemes, in which the rate ranged mainly between 93.1% and 100%. On the whole, the results are not significantly changed with different test positions. There are two possible reasons: 1) our training processes are effective by providing each participant with a detailed tutorial in an interactive step-by-step manner; and 2) our designed schemes are easy to learn due to their simple and intuitive operations.

2) Performance under Normal Condition: As participants only need to complete the login tasks without any time pressure or secondary tasks, we use the test results under normal condition as the baseline in our experiments.

Figure 7(a) shows the average time for a successful login under normal condition for each scheme. It is seen that the average login time for gTapper is 11.2 seconds, which was generally shorter than the other two schemes with 15.6 seconds

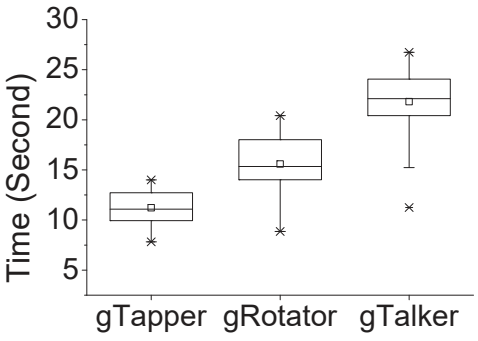


(a) Average login time of the tests in normal condition encountered at different test positions

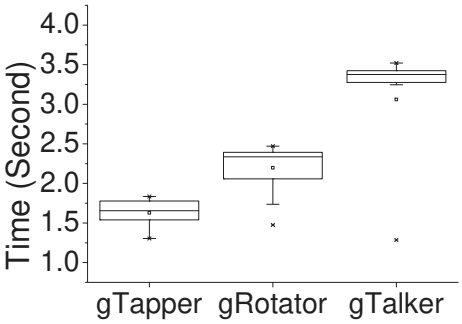


(b) Login success rates of the tests in normal condition encountered at different test positions

Fig. 6. Learning curves for gTapper, gRotator, and gTalker



(a) Distribution of login time



(b) Distribution of single element entry time

Fig. 7. Average login time and single element entry time

and 21.8 seconds for gRotator and gTalker, respectively. To investigate the differences in login time, Figure 7(b) shows the distribution of *single element entry time* for each scheme. Similarly, gTapper only required 1.63 seconds regarding the single element entry, which is much shorter than gRotator and gTalker each with 2.20 seconds and 3.05 seconds. Regarding the login success rate, gTapper, gRotator, and gTalker can reach a rate of 98.3%, 98.2%, and 98.2% under the normal condition, respectively. These results indicate that participants can make a tap task shorter than rotating head or speaking a number on Google Glass.

TABLE I
SIGNIFICANCE TEST ON AVERAGE LOGIN TIME IN GTAPPER.

Mode1	Mode2	Kruskal-Wallis test	
normal	timed	K=17.19	p-value$<$0.001★
	distraction	K=21.21	p-value$<$0.001★
	heavy distraction	K=36.53	p-value$<$0.001★
timed	timed + distraction	K=14.54	p-value$<$0.001★
	timed + heavy distraction	K=41.15	p-value$<$0.001★
distraction	distraction + timed	K=35.65	p-value$<$0.001★
	heavy distraction	K=4.59	p-value=0.032★
heavy distraction	heavy distraction + timed	K=23.32	p-value$<$0.001★
timed + distraction	timed + heavy distraction	K=18.37	p-value$<$0.001★

TABLE II
SIGNIFICANCE TEST ON AVERAGE LOGIN TIME IN GROTATOR.

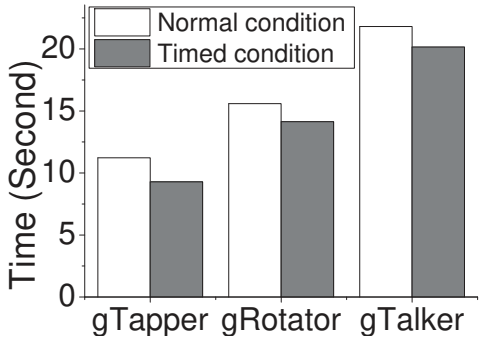
Mode1	Mode2	Kruskal-Wallis test	
normal	timed	K=2.06	p-value=0.151
	distraction	K=12.39	p-value$<$0.001★
	heavy distraction	K=27.25	p-value$<$0.001★
timed	timed + distraction	K=3.46	p-value=0.063
	timed + heavy distraction	K=25.07	p-value$<$0.001★
distraction	distraction + timed	K=9.94	p-value=0.002★
	heavy distraction	K=2.12	p-value=0.146
heavy distraction	heavy distraction + timed	K=0.63	p-value=0.426
timed + distraction	timed + heavy distraction	K=15.78	p-value$<$0.001★

TABLE III
SIGNIFICANCE TEST ON AVERAGE LOGIN TIME IN GTALKER.

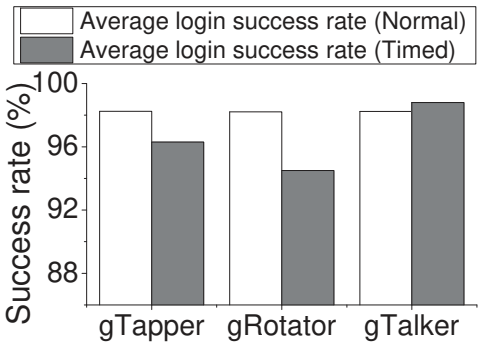
Mode1	Mode2	Kruskal-Wallis test	
normal	timed	K=5.10	p-value=0.024★
	distraction	K=5.64	p-value=0.018★
	heavy distraction	K=11.64	p-value$<$0.001★
timed	timed + distraction	K=6.72	p-value=0.010★
	timed + heavy distraction	K=13.00	p-value$<$0.001★
distraction	distraction + timed	K=2.63	p-value=0.105
	heavy distraction	K=1.39	p-value=0.238
heavy distraction	heavy distraction + timed	K=5.26	p-value=0.022★
timed + distraction	timed + heavy distraction	K=0.42	p-value=0.517

3) Impact of Time Pressure: Figure 8 presents participants' performance under time pressure without distractions. Generally, participants were found to enter passwords faster under time pressure that under normal condition, i.e., the average time for a successful login under timed condition for gTapper, gRotator and gTalker is 9.3 seconds, 14.1 seconds and 20.1 seconds. Similarly, participants can reduce the average time for a single element entry, i.e., 1.36 seconds for gTapper, 1.98 seconds for gRotator, and 2.84 seconds for gTalker as shown in Figure 9. Based on the statistical tests, we found that there is a significant difference in login time, where $p < 0.001$ for gTapper and $p = 0.024$ for gTalker (details can refer to Table I, Table II, and Table III). Regarding **Q1**, our results indicate that participants spend significantly less login time under the timed condition as compared to the normal condition.

Regarding the average login success rates, Figure 8(b) shows that participants under timed condition can achieve 96.3% and 94.5% for gTapper and gRotator, respectively, which are slightly lower than the normal condition. By contrast, participants achieve a rate of 98.8% for gTalker under timed condition, but it is very close to that in a normal condition. In addition, our statistical tests indicate that the results on login accuracy under timed condition and normal condition are not significant, i.e., the participants can still achieve high login accuracy even under the timed condition. This is because these tests may not be sufficiently difficult to distinguish the impacts of test conditions due to ceiling effect [46]. Regarding **Q2**, we found that the results on login accuracy are not significantly different under timed condition as compared to normal condition.



(a) Average login time



(b) Average login success rate

Fig. 8. Impact of time pressure

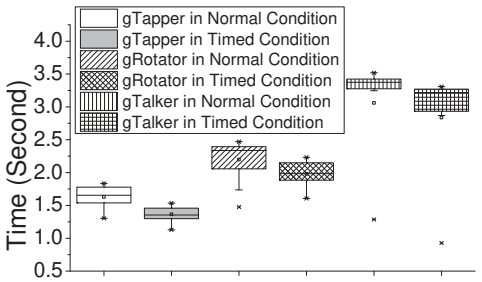
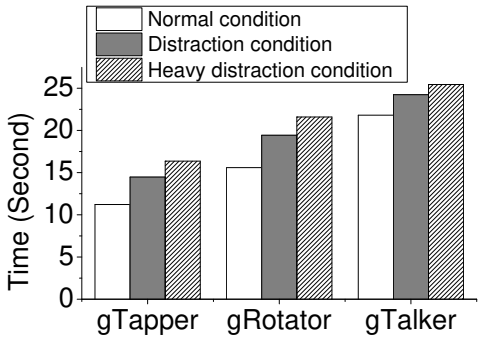


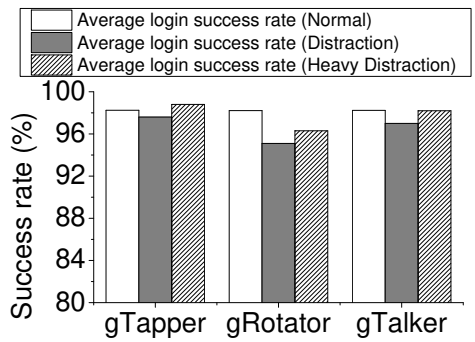
Fig. 9. Distribution of single element entry time under time pressure.

4) Impact of Distractions: Figure 10 shows the impact of distractions for average time login and average login success rate. It is observed that by raising the distraction level from no distraction to heavy distraction, participants should spend longer time on the login process, as well as the average time of single element entry. The statistical results on the impact of distractions can refer to Table I, II, and III for all three schemes (i.e., $p < 0.001$ for gTapper in both distraction level and heavy distraction level, $p < 0.001$ for gTapper in both distraction level and heavy distraction level, $p < 0.001$ for gRotator in both distraction level and heavy distraction level, and $p = 0.018$ and $p < 0.001$ for gTapper in distraction level and heavy distraction level, respectively). Thus, for **Q3** and **Q5**, we found the login time to be significantly longer with distraction as compared to a normal condition.

Figure 10(b) then shows that the average login success rates for gTapper, gRotator and gTalker were ranged between 97.6% and 98.8%, between 95.1% and 98.2%, and between 97% and 98.2%. Our statistical results also indicate no significant difference in the average login success rates at different distraction levels for all the three test groups; that is, participants can achieve high login accuracy even with distractions. Regarding **Q4** and **Q6**, it is found that participants can reach good login accuracy even if under distraction or heavy distraction as compared to a normal condition.



(a) Average login time



(b) Average login success rate

Fig. 10. Impact of distraction

Figure 12 presents the time distribution participants spent on secondary tasks. It is observed that participants have to take longer time for speaking number tasks than rotating head tasks, i.e., the average time of speaking numbers and rotating head is 4.6 seconds and 0.7 second, respectively. However,

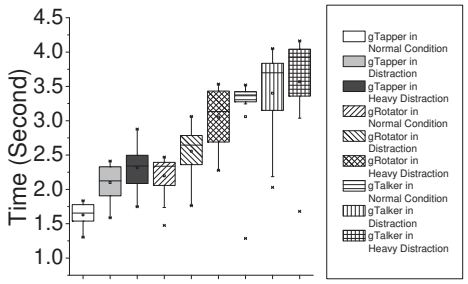


Fig. 11. Distribution of single element entry time under distraction.

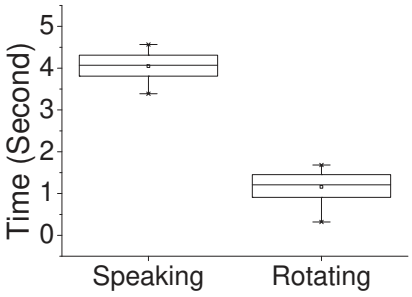


Fig. 12. Distribution of time for the secondary tasks

participants can achieve an average success rate of 92% and 100% for speaking number tasks and rotating head tasks, respectively. These results show that the secondary tasks could work effectively to distract participants during the password entry.

5) Impact of Combined Conditions: We examine the scheme performance under the combined conditions, which can involve both time pressure and distractions simultaneously. Compared to the tests without time pressure, the average login time under combined conditions decreases by 2.7 seconds on average. Our statistical results show that the difference in the average login time is significant in most cases (i.e., $p < 0.001$ for gTapper in both distraction level and heavy distraction level, $p = 0.002$ for gRotator in distraction level, and $p = 0.022$ for gTalker in heavy distraction level). However, the difference in the average login success rates is not significant. Therefore, regarding **Q7**, we found that the login time is significantly shorter under timed condition with (heavy) distraction, whereas regarding **Q8**, we found that the login accuracy is not significantly lower under timed condition with (heavy) distraction compared to normal condition with (heavy) distraction. It is worth noting that the time pressure can speed up the password entry process without greatly affect the login accuracy.

6) Memory Interference: As mentioned earlier, we generate a random password for each participant to test each scheme. All participants are allowed to use a "Show Password" function by swiping up with one finger on the touch-pad, if they forget the assigned passwords. Figure 13 shows the average number of "Show Password" triggered by the participants under normal condition. Note that each test requires a participant to complete three login attempts. Our experimental results show that no "Show Password" was used by the participants during the test of gTapper. For gRotator and

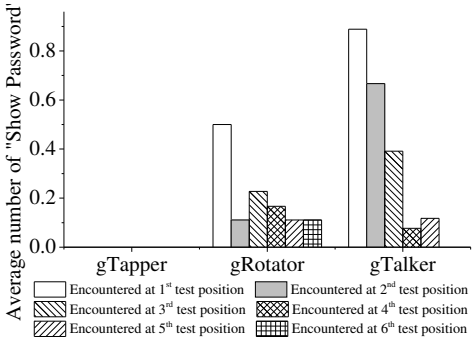


Fig. 13. Average number of "Show Password" used during the tests in normal condition encountered at different test positions

gTalker, participants used the "Show Password" more often in the tests appearing at first position than other positions. This is because participants may gradually get more familiar with the assigned password. For the tests appearing at the last position, the average number of using "Show Password" decreases to only 0.1 times for gRotator and 0 times for gTalker. Our results indicate that our schemes do not incur significant interference on password recall.

7) Users' Perception: In this study, we collect users' perception and feedback through online questionnaires. As shown in Figure 14, most participants found that our schemes are easy to learn and could be more secure than the existing password entry methods used on Google Glass. In particular, gTapper is the most popular among the three schemes, i.e., all of the participants are willing to use gTapper, while 56.1% and 58.5% of participants are willing to use gRotator and gTalker, respectively.

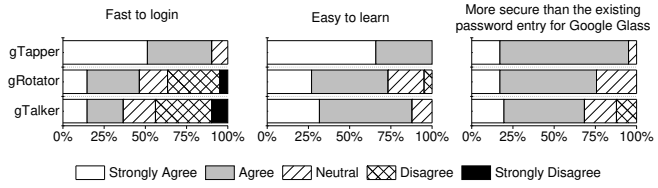


Fig. 14. Perception of participants.

8) Comparison with Existing Password Entry Method on Smart Glasses in Practice: In this part, we compare our password entry schemes with the existing real-world password entry methods available on commodity smart glasses, such as Google Glass and HoloLens. Most current password entry methods require users to input their passwords in plaintext via a standard keyboard on PCs or a touch screen on mobile phones, which are connected to their smart glasses, as explained in Section III-B. It is worth emphasizing that our schemes aim to preserve the benefits of traditional password entry schemes on smart glasses, and further improve their security and practicability.

Table IV shows a scheme comparison based on the security-deployability-usability metrics given by Bonneau et al. [32]. In particular, the metrics related to security focus on anti-eavesdropping of password, which correspond to the rows from "Resilient-to-Physical-Observation" to "Unlinkable". The metrics of deployability highlight the practicability of password entry schemes, which correspond to the rows from "Accessible" to "Non-Proprietary". The metrics of usability pay attention to the usability costs, which correspond to the rows from "Nothing-to-Carry" to "Easy-Recovery-from-Loss".

The comparison indicates that our designed schemes are able to improve the security level by offering the benefits of *Resilient-to-Physical-Observation*, *No-Trusted-Third-Party* and partial benefit of *Resilient-to-Internal-Observation*, because our schemes are secure against external eavesdropping attacks and unprivileged internal eavesdropping attacks without relying on any trusted third party. For deployability, our schemes can provide the benefit of *Negligible-Cost-per-User*, as they use common built-in hardware on smart glasses only, whereas the other methods need additional PCs or mobile phones to connect with smart glasses. For usability, our schemes can preserve most benefits of existing entry methods, and provide the benefit of *Nothing-to-Carry*, which is only partially offered by most other methods.

VI. Discussion

A. Security and Usability by Extending Password Alphabet and Length

Our current schemes adopt a 6-digit password, while we can raise the security level of the schemes by using a longer password and a richer password for certain scenarios with higher security requirements, such as online banking.

On one hand, applying a longer password is a direct way to raise the security level as additional rounds of password element entry associated with more hidden random keypads are involved (refer to Section IV-B). However, simply enlarging password length may introduce higher time cost of user authentication process and more memory efforts by users [9], which could eventually affect the usability of the schemes.

On the other hand, it is not difficult to achieve higher security level by enlarging a password space with a richer password alphabet [24]. For a richer password alphabet, our schemes can be easily adjusted. For example, by given a password alphabet with 36 elements, including digits 0 to 9 and letters a to z, gTapper's screen can be adjusted to accommodate 36 grids for the password alphabet. By displaying more grids, it is necessary to accelerate focus cursor moving by allowing long-distance shifts. For gRotator, we can modify the screen layout to 18 elements, so that a user may select an element by rotating his/her head to a correct position among 18 positions. For gTalker, both original keypad and transformed keypad can be adjusted to include all 36 elements in the password alphabet. To enter an element on the original keypad, a user has to speak out the corresponding letter or digital number on the transformed keypad (see the implementation details of gTalker in Section IV-E). The richer password alphabet could introduce more elements to be displayed and chosen by users and result in usability cost during user operations, but the usability of our schemes can be further mitigated and optimized by making clever tradeoffs between password alphabet and password strength, i.e., a richer password alphabet only requires a shorter password length to achieve the same password strength. Due to the complexity, we leave it as one of our further directions.

TABLE IV
COMPARISON BETWEEN THE PROPOSED SCHEMES AND THE EXISTING PASSWORD ENTRY METHOD ON SMART GLASSES USING SECURITY-DEPLOYABILITY-USABILITY METRICS [32] WHERE ▲ INDICATES THE BENEFIT IS OFFERED, △ INDICATES THE BENEFIT IS PARTIALLY OFFERED, WHILE *blank* CELL INDICATES THE BENEFIT IS NOT OFFERED

Metrics	Our schemes	Existing password entry on smart glasses
Resilient-to-Physical-Observation	▲	
Resilient-to-Targeted-Impersonation	△	△
Resilient-to-Internal-Observation	△	
Resilient-to-Theft	▲	▲
No-Trusted-Third-Party	▲	
Requiring-Explicit-Consent	▲	▲
Unlinkable	▲	▲
Accessible	▲	▲
Negligible-Cost-per-User	▲	△
Mature		▲
Non-Proprietary	▲	▲
Nothing-to-Carry	▲	△
Easy-to-Learn	▲	▲
Efficient-to-Use	△	▲
Infrequent-Errors	△	△
Easy-Recovery-from-Loss	▲	▲

B. Limitations

Ecological validity. This is an open challenge to most research in this area [8], [29], [6]. Our participants were mainly from universities who may be usually more active to use smart wearable products such as smart glasses. There is always a need to consider other population and include even larger sample size.

Password alphabet. How to include full password alphabet is a challenge, including digits, case sensitive letters and symbols. Due to the compact design of NED screen on smart glasses, it is very difficult to display too many characters on such small screen at the same time. One potential solution is to use multiple screens; however, users may need to frequently change among multiple screens in order to identify their password elements. This may harm the usability of the password entry schemes. Fortunately, emerging techniques like augmented reality and virtual reality could allow users to access on screens with wider views and more user-friendly interactions. For example, eye-gaze based interaction may be available for users to input passwords on future smart glasses, based on the eye gaze and movements [47]. With the eye-gaze based interaction, a user may quickly identify and select target elements on the NED screen.

Speech recognition. This is a necessary functionality for gTalker, and its performance is mainly decided by the accuracy and the time consumption for recognizing a number said by a user. This is similar to most systems that employ speech recognition [39]. Compared to traditional typing based input, speech recognition-based input does not have any advantages in accuracy or speed if the input is not a sentence but a single character like password element [48]. This is because saying a word is usually slower than a single tap. As our implementation adopts a general speech recognition function provided by Google's Android API, the input of our schemes only involves 10 single-digit numbers. To solve this issue, our implementation includes all homophones for the 10 single-digit numbers, and can improve the recognition rate to 93% in the study. In future, it is an interesting topic to investigate how to train a speech recognition model to achieve even higher accuracy and shorter login time.

Head movement estimation. How to accurately estimate head movements in gRotator using inertial sensors is still a challenge on smart glasses. The main factors include the cumulative errors of dead-reckoning based estimation algorithms and the accuracy of inertial sensors. All dead-reckoning based algorithms are subject to cumulative errors, because they estimate a current position based on a pre-determined position [34]. The accuracy of inertial sensors on existing smart glasses is still not high. Fortunately, the accuracy of inertial sensors would improve fast with better hardware. This trend will lead to better performance of gRotator.

VII. CONCLUSION

At present, most existing anti-eavesdropping password entry schemes on smart glasses are heavily depending on additional PCs or mobile devices connected to smart glasses, while the required devices may not be always available in a real scenario, e.g., public areas and outdoors. This requires users to switch between different systems and devices, which may cause interrupted experience and significantly degrade the practicability and usability of smart glasses. In this paper, we focus on this challenge and propose three anti-eavesdropping password entry schemes for smart glasses: named gTapper, gRotator and gTalker. These schemes can protect the password entry by breaking the correlation between the underlying password and the interaction observable to adversaries. In addition, our schemes do not need extra hardware beyond what is commonly available on existing smart glasses. To evaluate the scheme performance, we conducted an IRB-approved users study with 57 participants, and found that our designed schemes are easy to use in various real-world scenarios.

ACKNOWLEDGMENT

We would like to thank all participants for their hard work and valuable support in the user study. The work was

supported in part by NSFC under Grant 61802289. Robert Deng was supported in part by AXA Research Fund. Weizhi Meng was supported in part by H2020-SU-ICT-03-2018: CyberSec4Europe.

REFERENCES

[1] Y. Li, Y. Cheng, Y. Li, and R. H. Deng, "What you see is not what you get: Leakage-resilient password entry schemes for smart glasses," in *AsiaCCS'17*. ACM, 2017, pp. 327–333.

[2] Google, "Google Glass," 2017, https://developers.google.com/glass/distribute/glass-at-work.

[3] Microsoft, "Microsoft Hololens," 2017, https://www.microsoft.com/microsoft-hololens/en-us.

[4] T. Matsumoto and H. Imai, "Human identification through insecure channel," in *EUROCRYPT'91*, 1991, pp. 409–421.

[5] D. Weinshall, "Cognitive authentication schemes safe against spyware," in *S&P 2006*, 2006, pp. 6–pp.

[6] S. Li and H.-Y. Shum, "Secure human-computer identification (interface) systems against peeping attacks: SecHCI," *Cryptology ePrint Archive: Report 2005/268*, 2005.

[7] B. Coskun and C. Herley, "Can "something you know" be saved?" in *Information Security*, 2008, pp. 421–440.

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

[9] Q. Yan, J. Han, Y. Li, and R. Deng, "On limitations of designing usable leakage-resilient password systems: Attacks, principles and usability," in *NDSS 2012*, 2012.

[10] P. D. Adamczyk and B. P. Bailey, "If not now, when?: the effects of interruption at different moments within task execution," in *CHI 2004*, 2004, pp. 271–278.

[11] M. Korolov, 2016, http://www.hypergridbusiness.com/2016/08/alibaba-working-on-vr-payments/.

[12] J. Dolcourt, 2013, http://www.cnet.com/news/everything-you-need-to-know-about-google-glass-faq.

[13] D. K. Yadav, B. Ionascu, S. V. K. Ongole, A. Roy, and N. Memon, "Design and analysis of shoulder surfing resistant pin based authentication mechanisms on google glass," in *FC 2015*, 2015, pp. 281–297.

[14] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, "Beware, your hands reveal your secrets!" in *CCS 2014*, 2014.

[15] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *CCS*, 2014.

[16] J. Sun, X. Jin, Y. Chen, J. Zhang, R. Zhang, and Y. Zhang, "Visible: Video-assisted keystroke inference from tablet backside motion," in *NDSS 2016*, 2016.

[17] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "When good becomes evil: Keystroke inference with smartwatch," in *CCS 2015*. ACM, 2015, pp. 1273–1285.

[18] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu, "Friend or foe?: Your wearable devices reveal your personal pin," in *ASIACCS 2016*, 2016.

[19] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "Mole: Motion leaks through smartwatch sensors," in *MobiCom 2015*, 2015.

[20] R. Schlegel, K. Zhang, X.-y. Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: A stealthy and context-aware sound trojan for smartphones." in *NDSS*, 2011.

[21] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: password inference using accelerometers on smartphones," in *ACM HotMobile 2012*, 2012.

[22] O. Begemann, 2012, http://oleb.net/blog/2012/10/remote-view-controllers-in-ios-6/.

[23] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: validating ssl certificates in non-browser software," in *CCS 2012*. ACM, 2012, pp. 38–49.

[24] F. Schaub, R. Deyhle, and M. Weber, "Password entry usability and shoulder surfing susceptibility on different smartphone platforms," in *MUM 20112*, 2012.

[25] A. Forget, S. Chiasson, and R. Biddle, "Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords," in *CHI'10*. ACM, 2010, pp. 1107–1110.

[26] V. Roth, K. Richter, and R. Freidinger, "A pin-entry method resilient against shoulder surfing," in *CCS 2004*, 2004.

[27] A.-M. Horcher, "One size does not fit mobile: Designing usable security input on mobile devices," in *SOUPS'18*. ACM, 2018, p. 5.

[28] L. Ginzburg, P. Sitar, and G. K. Flanagin, "User authentication system and method," May 25 2010, uS Patent 7,725,712.

[29] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *SOUPS 2007*. ACM, 2007, pp. 13–19.

[30] W. Shanklin, 2016, http://newatlas.com/apple-watch-1-vs-samsung-gear-s3-comparison/45289/.

[31] Google, 2016, https://developer.android.com/develop/index.html.

[32] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *S&P 2012*, 2012.

[33] Z. J. Li, C. Yang, and X. Y. Wu, "The improvement and evaluation of an analog digital television systems availability and users experience," in *Applied Mechanics and Materials*, vol. 496, 2014, pp. 2027–2033.

[34] I. Kamal, "WFR, a dead reckoning robot–a practical application to understand the thoery," 2008.

[35] S. Tupling and M. Pierrynowski, "Use of cardan angles to locate rigid bodies in three-dimensional space," *MBEC*, vol. 25, no. 5, pp. 527–532, 1987.

[36] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *TPAMI*, vol. 31, no. 4, pp. 607–626, 2009.

[37] S. Overflow, 2016, http://stackoverflow.com/questions/17616994/offline-speech-recognition-in-android-jellybean.

[38] M. Stenman, "Automatic speech recognition an evaluation of google speech," p. 37, 2015.

[39] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[40] G. Charness, U. Gneezy, and M. A. Kuhn, "Experimental methods: Between-subject and within-subject design," *JEBO*, 2012.

[41] W. J. Horrey and C. D. Wickens, "Driving and side task performance: The effects of display clutter, separation, and modality," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46, no. 4, pp. 611–624, 2004.

[42] P. Goolkasian, P. W. Foos, and M. Eaton, "Modality effects in sentence recall," *The Journal of general psychology*, vol. 136, no. 2, pp. 205–224, 2009.

[43] J. A. Doornik and H. Hansen, "An omnibus test for univariate and multivariate normality," *Oxford Bulletin of Economics and Statistics*, vol. 70, no. s1, pp. 927–939, 2008.

[44] H. Bhattacharyya, "Kruskal–wallis test," *Encyclopedia of Statistical Sciences*, 1983.

[45] J. D. Gould and C. Lewis, "Designing for usability: key principles and what designers think," *Communications of the ACM*, vol. 28, no. 3, pp. 300–311, 1985.

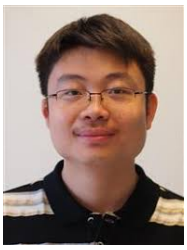
[46] A. Aron, E. Coups, and E. N. Aron, *Statistics for The Behavioral and Social Sciences: Pearson New International Edition: A Brief Course*. Pearson Higher Ed, 2013.

[47] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[48] A. G. Hauptmann and A. I. Rudnicky, "A comparison of speech and typed input," in *SNLW*, 1990.

Yan Li is currently with the School of Cyber Engineering, Xidian University, China. He received his Ph.D. degree in information systems from Singapore Management University in 2014. His research interests include usable security, password-based user authentication, biometric-based user authentication, privacy and security in social networks, security-related data analytics, deep learning.

Yao Cheng is currently a senior researcher at Huawei International in Singapore. She received her Ph.D. degree in Computer Science and Technology from University of Chinese Academy of Sciences. Her research interests include security and privacy in deep learning systems, blockchain technology applications, Android framework vulnerability analysis, mobile application security analysis, and mobile malware detection.



Weizhi Meng is currently an Assistant Professor in the Cyber Security Section, Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong (CityU), Hong Kong. Prior to joining DTU, he worked as a research scientist in Infocomm Security (ICS) Department, Institute for Infocomm Research, A*Star, Singapore. He won the Outstanding Academic Performance Award during his doctoral study, and is a recipient of the Hong Kong Institution of Engineers (HKIE) Outstanding Paper Award for Young Engineers/Researchers in both 2014 and 2017. His primary research interests are cyber security and intelligent technology in security, including intrusion detection, smartphone security, biometric authentication, HCI security, trust computing, and blockchain in security. He is a senior member of IEEE.

Yingjiu Li is currently a Ripple Professor in the Department of Computer and Information Science at the University of Oregon. His research interests include IoT Security and Privacy, Mobile and System Security, Applied Cryptography and Cloud Security, and Data Application Security and Privacy. He has published over 140 technical papers in international conferences and journals, and served in the program committees for over 130 international conferences and workshops, including top-tier cybersecurity conferences.

Robert H. Deng is AXA Chair Professor of Cybersecurity, Director of the Secure Mobile Centre, and Deputy Dean for Faculty & Research, School of Information Systems, Singapore Management University (SMU). His research interests are in the areas of data security and privacy, network security, and system security. He received the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. He serves/served on many editorial boards and conference committees, including the editorial boards of ACM Transactions on Privacy and Security, IEEE Security & Privacy, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is a Fellow of IEEE and Fellow of Academy of Engineering Singapore.