SAW: Wristband-based Authentication for Desktop Computers

SHRIRANG MARE, Paul G. Allen School of Computer Science, University of Washington, USA
REZA RAWASSIZADEH, Department of Computer Science, University of Rochestor, USA
RONALD PETERSON, Department of Computer Science, Dartmouth College, USA
DAVID KOTZ, Department of Computer Science, Dartmouth College, USA

Token-based proximity authentication methods that authenticate users based on physical proximity are effortless, but lack explicit user intentionality, which may result in accidental logins. For example, a user may get logged in when she is near a computer or just passing by, even if she does not intend to use that computer. Lack of user intentionality in proximity-based methods makes them less suitable for multi-user shared computer environments, despite their desired usability benefits over passwords. We present an authentication method for desktops called Seamless Authentication using Wristbands (SAW), which addresses the lack of intentionality limitation of proximity-based methods. SAW uses a low-effort user input step for explicitly conveying user intentionality, while keeping the overall usability of the method better than password-based methods. In SAW, a user wears a wristband that acts as the user's identity token, and to authenticate to a desktop, the user provides a low-effort input by tapping a key on the keyboard multiple times or wiggling the mouse with the wristband hand. This input to the desktop conveys that someone wishes to log in to the desktop, and SAW verifies the user who wishes to log in by confirming the user's proximity and correlating the received keyboard or mouse inputs with the user's wrist movement, as measured by the wristband. In our feasibility user study (n=17), SAW proved quick to authenticate (within two seconds), with a low false-negative rate of 2.5% and worst-case false-positive rate of 1.8%. In our user perception study (n=16), a majority of the participants rated it as more usable than passwords.

CCS Concepts: • **Security and privacy** → **Authentication**; • **Human-centered computing** → *Ubiquitous and mobile computing*;

Additional Key Words and Phrases: Authentication, Wearable, Security, Privacy

ACM Reference Format:
Shrirang Mare, Reza Rawassizadeh, Ronald Peterson, and David Kotz. 2018. SAW: Wristband-based Authentication for Desktop Computers. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 125 (September 2018), 29 pages. https://doi.org/10.1145/3264935

1 INTRODUCTION

Authentication has become an integral part of computer usage, but it still remains an interruptive step in people's workflow. To authenticate to a computer, depending on the authentication method, users must exert mental effort (e.g., recall their password) and/or physical effort (e.g., type their password). These factors increase the cost of context switch for users – cost of switching attention from a primary task to the authentication step

Authors' addresses: Shrirang Mare, Paul G. Allen School of Computer Science, University of Washington, Seattle, WA, USA, shri@cs.uw.edu; Reza Rawassizadeh, Department of Computer Science, University of Rochestor, Rochestor, NY, USA, rezar@cs.ucr.edu; Ronald Peterson, Department of Computer Science, Dartmouth College, Hanover, NH, USA, rapjr@cs.dartmouth.edu; David Kotz, Department of Computer Science, Dartmouth College, Hanover, NH, USA, david.f.kotz@dartmouth.edu.

2474-9567/2018/9-ART125 $15.00
https://doi.org/10.1145/3264935

and back to the task – disrupting users' workflow [53]. The disruption due to a *single* authentication instance may be negligible, but it adds up, and depending on the authentication method and the nature of computer usage, the disruption can vary from barely noticeable to time consuming and frustrating [34]. For example, in some workplaces, such as hospital settings, where people typically log-in close to a hundred times daily at different workstations, often for just a few moments, the disruption is significant; according to an industry report, hospital staff spend about 30 to 45 minutes a day authenticating to computers and web applications [20]. As a result, frustrated users devise workarounds to make the authentication process more convenient to get their work done [1]. These workarounds include choosing simple passwords that are easy to remember and type, writing them down on a piece of paper – ironically, some vendors sell special stickers to write a username and password [25] – or choosing to disable authentication entirely, all of which leave the computer vulnerable [22, 50]. Rather than browbeating users to change their behavior, we need a secure authentication method that blends seamlessly into users' workflow. This work is largely motivated by the authentication challenges in multi-user shared-desktop settings in hospitals and other enterprise environments [21, 50]. Because shared desktops are prevalent in many enterprise environments, in this work we focus on authentication for desktop computers (henceforth, simply 'desktops'); the method could be easily adapted to laptops.

Proximity-based authentication methods (e.g., Apple Watch [4], Atama Sesame [6], ZIA [15]) are good candidates to build a usable, zero-effort, secure authentication solution, but they have two drawbacks: they do not work well in multi-user shared settings and they may cause unintended authentications. Typically, in a proximity-based authentication method, a user carries a wireless authentication token that is configured to authenticate the user to a target computer, and whenever the user is within a certain distance of the target computer – as determined by radio signal strength from the token – the user is automatically authenticated to the computer; some implementations require an additional condition that there should be some input to the target computer (presumably from the user) before the user is logged in, but there is no verification that the input was indeed provided by the user being authenticated. Multi-user shared settings are problematic for proximity-based methods because when there are multiple authorized users near a computer, all within the authentication distance threshold, these methods cannot determine which user (if any) should be authenticated. The second problem with proximity-based authentication methods is that a user may accidentally get authenticated without her consent, e.g., when she may just be passing by or present in the next room (or cubicle). Furthermore, adversaries can circumvent the distance threshold by using relay attacks [9, 44], and can get a user authenticated even if the user is far outside the set authentication distance threshold; relay attacks have been demonstrated for Bluetooth [30], RFID [17], and NFC [24].

The underlying cause for these shortcomings of proximity-based methods is that they do not require an explicit intent from a user before the user is authenticated, i.e., a user does not have to explicitly convey that she wishes to authenticate to the target computer. Instead, proximity-based methods *infer* user intentionality using proximity based on the assumption that if a user is 'near' a computer, the user wants to log in that computer (or at least it is okay to log in the user). The distance threshold for *nearness* depends on the underlying proximity protocol, and can range from 20 cm (for NFC) to 10 m (for Bluetooth), but using relay attacks, an adversary can extend this range by orders of magnitude [24, 30]. Thus, pure proximity-based authentication leaves a computer vulnerable when the computer is used outside a physically secure environment or when used in a multi-user shared settings. In this paper, we propose Seamless Authentication using Wristbands (SAW), an authentication method designed to address this shortcoming of proximity-based authentication methods, and we do so by adding a quick low-effort user input step that explicitly captures user intentionality for authentication.

In SAW, the user's wristband (e.g., fitness tracker, smartwatch) acts as the user's authentication token. To set up the wristband, the user (or the user's workplace IT staff) initializes the wristband with the user's identity and user-chosen PIN (e.g., by pairing [28] the wristband with a known trusted device that is set up to initialize SAW wristbands), and then by pairing the wristband with the target desktop so they can communicate securely in the

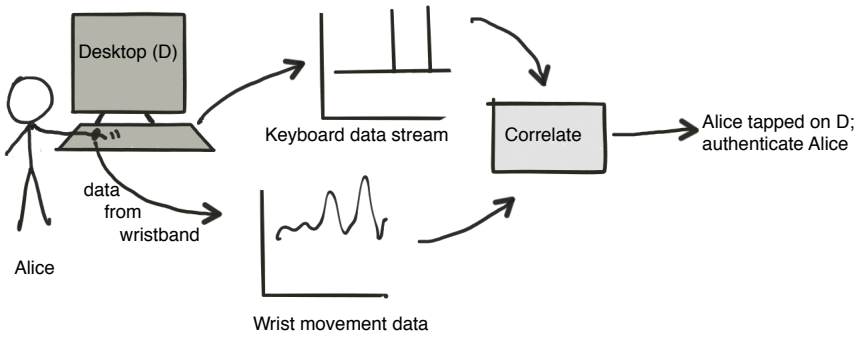

Fig. 1. User Alice taps on the keyboard for authentication. The desktop correlates the received keystroke (keyboard data stream) and the wrist movement (wristband data stream) to determine that Alice intends to authenticate, and authenticates Alice.

future. Subsequently, to authenticate to the target desktop, the user simply taps a key on the keyboard multiple times or wiggles the mouse with her *wristband hand*. Figure 1 illustrates how a user (Alice) authenticates to a desktop by tapping a key few times with her wristband hand. The tapping or mouse-wiggle interaction, called *intent action*, conveys to the target desktop that some user intends to authenticate and conveys to the user's wristband that the user intends to authenticate to some desktop. SAW correlates the timing of the keystroke (or mouse wiggle) on the target desktop with the movement of all nearby wristbands to identify the user (among all the nearby users) who performed the interaction and unlocks the target desktop for that user.

SAW uses a bilateral approach for authentication, an approach also used in prior work [33, 35–37]. In bilateral authentication, a user is verified by comparing measurements from two independent sources – correlating wristband movement with keyboard/mouse inputs in SAW's case. The prior work closest to SAW is ZEBRA [35], which uses a wristband for continuous authentication on a desktop. Compared to ZEBRA, SAW uses a different authentication interaction and correlation technique that enable SAW to securely authenticate a user within 2 s (ZEBRA takes 20 s); we discuss the differences in detail in Section 8.1.

Contribution. We propose SAW, a new proximity-based authentication method for desktop computers that captures user intentionality for authentication. To identify the user interacting with the desktop, SAW leverages the physical link established between the user's wristband and the keyboard/mouse when the user interacts with them. We identified and evaluated two potential low-effort authentication interactions (Mouse-wiggle and Tap-5x) that could be used for authentication in SAW. Of the two interactions, Tap-5x performed better in terms of time for authentication and accuracy. Using Tap-5x as the authentication interaction, we evaluate SAW's security under the standard Dolev-Yao adversary model [16]. We also evaluated SAW's feasibility and usability through two in-lab user studies (n=17,16). In the user studies, SAW had a low false-negative rate of 2.5% and a low worst-case false-positive rate of 1.8%.

2 BACKGROUND

The type of settings we envision where SAW would be most effective are where proximity alone is not sufficient to capture user intentionality to authenticate users.

2.1 User Intentionality for Authentication

User intentionality is an important principle of user authentication – a user should be authenticated to a computer only if she intends to authenticate to that computer [43]. In authentication methods where a user manually

provides authentication credentials (e.g., username and password, fingerprint), the act of providing the credentials conveys the user's intentionality, but in methods where credentials are wirelessly shared (e.g., proximity-based methods), user intentionality is inferred and doing so incorrectly leads to security errors. Adding a user input step to make user intentionality explicit in an otherwise zero-effort proximity-based authentication method while keeping the overall method usable and non-disruptive to the user's workflow is challenging.

To express authentication intent, a user needs to specify two things: *1)* that the user intends to authenticate (to something), and *2)* the authentication target, the desktop the user wants to authenticate to. Naive approaches where a user presses a button on her wristband conveys the user's intent to authenticate, but does not indicate which desktop the user intends to use – an approach that may be sufficient in a single-user single-desktop setting where the user authenticates only to one desktop, but not in a multi-user shared-desktop environment where a user can authenticate to multiple desktops. Another approach is where each desktop is assigned a unique number, which is clearly displayed on each desktop's display, and the user enters on her token the displayed number for the desktop she intends to authenticate. Even if we assume that the token is easily accessible, say a wristband (or a smartwatch), expressing intentionality this way is cumbersome: a user has to locate the number of the desktop, raise her wrist, and input the number through the small interface.

2.2 Authentication in a Hospital Setting: Observations at a Local Hospital

To better understand the unique challenges in these environments, we reached out to a large local hospital, and engaged the hospital IT staff and clinicians through meetings and interviews. Below we share our key learnings, which further motivated this work and informed SAW's design. We had several meetings with the Chief Medical Information Officer (CIO), Information Security Manager, and Technical Compliance staff of a large local hospital. Through our interactions, we learned the challenges they face around authentication and the workarounds that hospital employees use, which echoed findings in prior work [21, 25]. To understand the authentication challenges and workflows from the end user (nurses and clinicians) perspective, the lead author observed the workflow of two nurses and two clinicians by shadowing them, and later conducted semi-structured interviews with them. (The observation and interview protocol was approved by our university's IRB.) When we visited the hospital, the team was actually in their third month of piloting a NFC-based authentication solution, and we were able to learn about their experience.

The workflow in the hospital varied a lot across departments. Some departments had more desktops, some departments used more laptops; some departments had one desktop/laptop per user, whereas in some departments users shared desktops and laptops. However, a common aspect of their workflow was a series of short computer interactions (sometimes just for a few minutes), spread throughout their shifts. For example, in Out Patient Departments, clinicians would visit patients in different examination rooms and would access the computer in that room (which was either a desktop, a laptop brought in and set up by a nurse, or a laptop carried by the clinician) while examining the patient, and after examination the clinician would go to another room to visit the next patient and would access a computer there. This meant that if a nurse or a clinician was to log out every time they were done with a *session* on their workstation, they would have to authenticate numerous times a day. When asked if she logs out of her workstation after her session, a clinician said no, justifying *"I would have to log in 100 times a day!"* Laptops were inconvenient to carry, but still nurses and clinicians preferred them because it meant they could log out (and log in) less, but because laptops were inconvenient to carry everywhere, they would often get left behind (and unattended) while the user went across the hall to do some task. We found two instances where laptops were left logged in and unattended.

The feedback from nurses and clinicians based on a three-month pilot of a NFC-based authentication solution was mixed. *"We had lot of challenges. If it worked as designed it won't be a big deal,"* the CIO said while referring to the system integration challenges they faced. An IT staff echoed the sentiment *"It just doesn't work as advertised"*

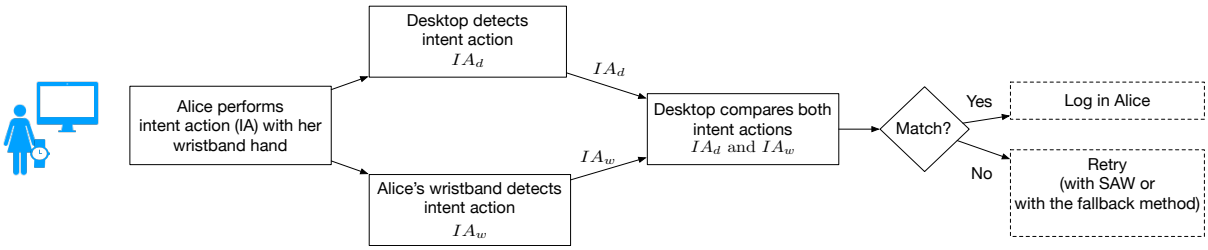

Fig. 2. An overview of SAW authentication.

while referring to instances when users could not authenticate using the new system. The NFC-based solution used a different radio frequency (RF) compared to what the hospital was using for their RF-based access-control systems. So the existing RFID cards could not be used, and new cards were bought. The pilot users now had (at least) two cards attached to their lanyard, which presented another problem: when a user would put their lanyard cards in front of the reader, sometimes the reader would not detect the card if was behind another card. We learned that several pilot users had reported this issue and the hospital was considering buying NFC cards that support both frequencies. Another concern the IT staff had with NFC cards was users misplacing or leaving their cards behind, and the cards getting stolen. So the staff had employed a tiered approach to authentication, where a user authenticates with their password and their NFC card, and after which for the next 5 hours (half a shift time) the user could authenticate with only the NFC card (i.e., without their password), if they wished. Other hospitals, as we learned, address this concern by requiring a 4-digit PIN along with each NFC card login. Given the difficulties with an NFC-based solution (even in settings that already use RFID cards), professionals' concerns around loss and misuse of NFC cards, and NFC's inherent vulnerability to range-extension attacks [18, 24], there remains a need for a secure usable authentication method.

3 SAW OVERVIEW

The intuition behind SAW is that a user's fine wrist movements that produce taps on the keyboard or move the mouse should strongly correlate with the keyboard taps or mouse movements, and based on the timing of the actions (wrist movement and taps/mouse movement) they should be uniquely linkable, allowing us to tie the user's wristband movement to the taps on the keyboard or movement of the mouse of a specific desktop. Further, we can link the wristband to the wearer's identity, using a variety of known methods (e.g., using PIN [4] or biometric [14]), thus linking the user to the keyboard tap for the purposes of authentication. This correlation should be unique and difficult to forge (mimic by an adversary) because it involves unpredictable fine human motor actions that are difficult to observe and imitate in real time; average human reaction time is about 215 ms [47], and in SAW the correlation time threshold is 50 ms.

Figure 2 shows an overview of how SAW works. A user performs a pre-defined intent action on the desktop (i.e., provides inputs on a keyboard and/or mouse). The intent action is detected by the desktop and the wristband independently. The desktop detects the action using keyboard/mouse inputs, the wristband detects the action using the data from the accelerometer and gyroscope sensors. The desktop then temporally compares the two intent actions (the intent action it detected and the intent action detected by the wristband) to determine whether they correlate, and if they do, the user is authenticated and logged in. Otherwise, the user has to try again, either with SAW or using the fallback authentication method (e.g., username and password).

The usability of SAW depends on the usability of the intent action. Our approach to keep the intent action quick and easy to perform was to build it around simple desktop interactions that are familiar to users. If we break down the steps involved in accessing a typical desktop, there are three steps: *a*) invoke the login screen on

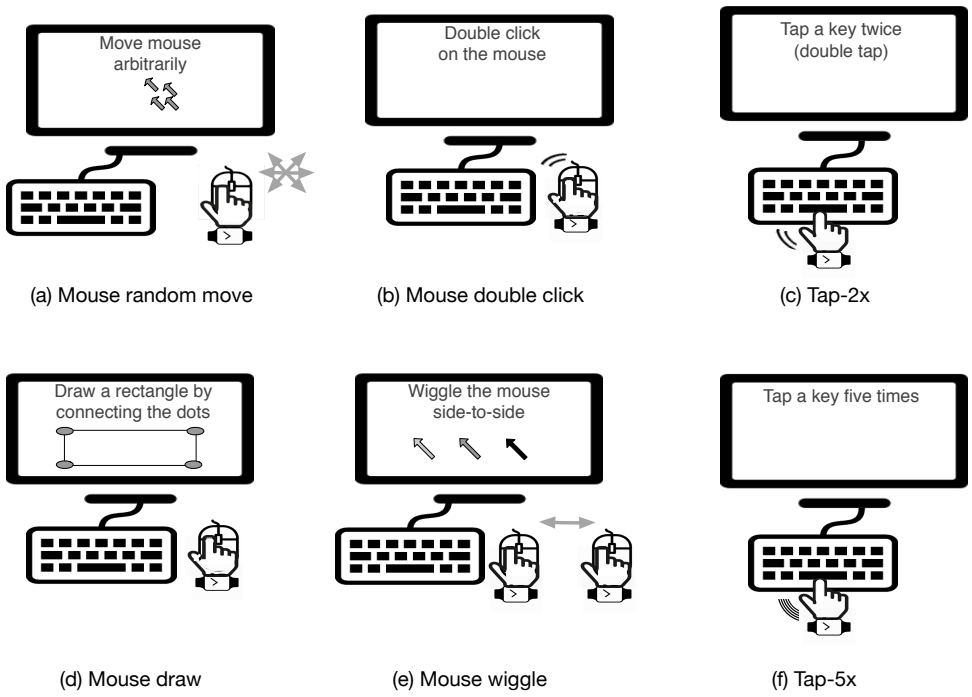

Fig. 3. Intent actions that we considered for SAW. All interactions are performed with the wristband hand.

a locked desktop, *b*) enter username and password, and *c*) start using the desktop. To invoke the login screen, users perform simple actions such as tapping a key once or multiple times, clicking the mouse, or moving the mouse. Since these actions are already part of users' workflow, intent actions built around these actions could blend easily in their workflow. So, based on these actions that users often perform, we considered six different intent actions to use in SAW. Figure 3 shows a sketch of how these six interactions were performed.

(1) *Mouse-random-move*: Move the mouse arbitrarily for a few seconds.
(2) *Mouse-double-click*: Click twice on the mouse button.
(3) *Tap-2x*: Tap a key twice (double tap).
(4) *Mouse-draw*: Draws a specified shape (e.g., rectangle, circle) on the display with the mouse.
(5) *Mouse-wiggle*: Wiggle the mouse side-to-side for a few seconds with a locked wrist.
(6) *Tap-5x*: Tap a key (any key) on the keyboard five or more times.

To identify potential intent actions, we conducted two small exploratory pilot studies where we asked five volunteers to perform these actions while we recorded the wrist movement and desktop inputs. Our goal was to study the wristband movement data before, during, and after these actions are performed, to identify which of these intent actions can be reliably detected from the wristband movement data. In our initial experiments, we considered Mouse-random-move, Mouse-double-click, and Tap-2x, but all of these actions proved difficult to identify from wrist movement data. The duration for Mouse-double-click and Tap-2x was too short, which made it difficult to isolate that action in the wristband movement data, especially when the wrist moves immediately before and after the action. Mouse-random-move was relatively long (about 2 seconds or more), but throughout the action most participants' wrists were relatively stationary and the movement of the mouse and the wrist

was not significant enough, which makes correlation hard. In the second pilot study, we experimented with Mouse-draw, Mouse-wiggle, and Tap-5x. Mouse-draw required participants to move the mouse over a large shape drawn on the display so as to cause some wrist movement, and the data seemed promising for correlation, but this action was slow compared to Mouse-wiggle and Tap-5x, and also required visual attention, so we did not study this action further. Mouse-wiggle and Tap-5x were easy to identify in the wristband movement data and they did not require any visual attention, which makes it easy to perform these actions. Because Mouse-wiggle and Tap-5x proved most promising in our initial analysis, we use them as authentication interactions in SAW.

3.1 Design Goals

As a potential solution, SAW has the following usability (U*) and security (S*) goals (with Section numbers where we evaluate these goals):

- U1 *Accurate:* SAW should be accurate in identifying the user when she intends to authenticate the desktop (Section 6.1).
- U2 *No-need-to-memorize:* SAW should not require the user to memorize any secret (Section 6.2.1).
- U3 *Quick:* SAW should be quick to unlock a desktop; we aim for authentication with SAW to be within one to two seconds (Section 6.2.2).
- U4 *User-agnostic:* SAW should not depend on how the user performs the authentication interaction. If the user's authentication interaction behavior changes, SAW should still be able to authenticate the user (Section 6.2.3).
- U5 *Low-effort and Easy-to-perform:* SAW should require no (or minimal) physical effort and should be easy to perform (Section 6.2.4).
- S1 *Requires-explicit-intent:* SAW should require a user to explicitly express her intent for authentication (Section 6.3).
- S2 *Resilient-to-physical-observation:* SAW should not leak any secret information to an adversary that is capable of physically observing the user during authentication. The adversary can video or audio record the user during authentication (Section 6.3).
- S3 *Resilient-to-accidental-logins*: SAW should be resilient to accidental logins, i.e., a user should not be authenticated without her consent (Section 6.3.1).
- S4 *Resilient-to-mimic-attack*: The way SAW authenticates a user makes it susceptible to mimicking attacks, so SAW should be resilient against mimicking attacks (Section 6.3.2).

3.2 System Assumptions

We make following assumptions in SAW to get the data required for authentication.

- A1 *Wristband:* The user wears a wristband that has accelerometer and gyroscope sensors and a radio (e.g., Bluetooth) to communicate to the target desktop.
- A2 *Paired:* The wristband is already paired with the target desktop, a one-time activity, using a secure pairing method [28, 31].
- A3 *Wristband wear detection:* The wristband can detect when it is worn and when it is taken off from the wrist. This can be achieved for example, using an optical sensor as used in Apple Watch [4] or a sensor attached to the clasp that triggers when the watch is worn or taken off. The wristband is securely associated to its wearer (to prevent any unintended sharing) by detecting when it is taken off and requiring the wearer, when she dons the wristband, to reconfirm her identity using a PIN (as on the Apple Watch [4]), using a biometric (like bioimpedance [14]), or any other convenient and secure method.
- A4 *No hardware or software tampering:* The adversary does not tamper with the hardware or software of the target desktop and the user's wristband. Our goal for SAW is to be as secure as password. If an adversary can tamper with the desktop, the adversary can also easily capture the user's password.

3.3 Adversary and Threat Model

For SAW, we consider a Dolev-Yao adversary who has physical access to the target desktop and can observe (and even record) the user when the user is in radio proximity of the target desktop. (Under the Dolev-Yao model [16], the adversary has complete control over the communication channel between the desktop and wristband, but the adversary is computationally bounded, i.e., cannot break cryptographic primitives.) The adversary may be a curious family member, friend, colleague, or even an authorized user of the target desktop. The goal of the adversary is to log in as the user, but we assume the adversary cannot spoof the user's wristband. Thus, the adversary attempts to log in as the user by performing the intent action on the target desktop in a way that would fool SAW. For a given desktop input, whether SAW authenticates the adversary as the user depends on the user's wristband movement data when the intent action was performed. So depending on 'where the user is' (within radio proximity of the desktop or away) and 'what the user is doing' (sitting, walking, authenticating on a nearby desktop) when the adversary attempts his attack, we consider three types of scenario.

(1) *Accidental login:* In this attack, the adversary waits for an opportunity when the user is in radio proximity of the target desktop. The user could be sitting, walking, writing, or working on another desktop, but the user is not attempting to log in to another computer. The adversary attempts to authenticate (as the user) to the target desktop. (The attacker may employ range-extending techniques to falsely indicate the user's presence near the target desktop.)

(2) *Mimic attack without wireless jamming:* In this attack, the adversary waits for an opportunity when the user is in radio proximity and is attempting to authenticate to another nearby desktop. This attacker monitors the user's attempt to authenticate to a nearby desktop using a side-channel (e.g., visual, audio) and attempts to authenticate (as the user) to the target desktop by mimicking the user's intent action. To match the timing of the user's intent action, the adversary may also anticipate when the user's wrist will move in a pattern similar to the intent action.

(3) *Mimic attack with wireless jamming:* This attack is similar to the previous one, but in this attack, the adversary can cause wireless interference and selectively jam any communication to and from the user's wristband. We consider this as a separate attack because the ability to jam wireless communication affects the adversary's chance of success.

4 SAW: METHOD

To use a wristband with SAW for authentication on a target desktop, the wristband should be paired with the target desktop (Assumption A2, Section 3.2), and activated with the user's identity. Wearing a paired and activated wristband, a user approaches the target desktop and performs an intent action (i.e., Tap-5x or Mouse-wiggle) to log in. Below we discuss the protocol and method to identify the user (among multiple nearby users) who should be authenticated and logged in to the target desktop.

4.1 Intent Action and Correlation Events

An intent action generates two data signals, one in the desktop in the form of keyboard or mouse inputs, and another in the wristband in the form of wrist-movement data. Intuitively, since there is one source for both the signals, they should correlate. Existing applications that leverage human action to authenticate or pair two devices compare the same type of signals, e.g., accelerometer data [37] or Wi-Fi measurements [36]. In SAW, the two signals are fundamentally different, which makes it difficult to compare them directly.

The intent actions we chose involve specific events that can be measured in both the signals, and using the timing of each events we show that we can link the two signals and identify the user (the wristband) that performed the intent action. We call these specific events *correlation events*, and the moments in the intent action related to these correlation events as *correlation points*. Figure 4 shows correlation points (red circular dots) along

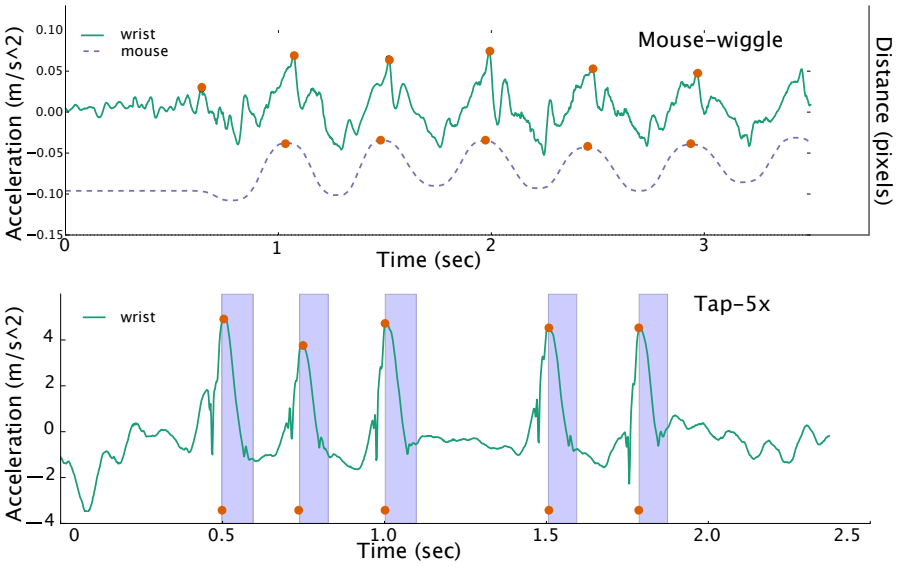

Fig. 4. Sample Mouse-wiggle and Tap-5x interactions with correlation points marked in red circular dots. Solid line is acceleration data (along one axis) from the wristband; dotted line is mouse pointer displacement (in pixels) alone x-axis; and the shaded region indicates the duration of a tap (keystroke), from KeyDown and KeyUp.

accelerometer data stream and desktop input stream from sample Mouse-wiggle and Tap-5x actions. In Figure 4, the figure at the top shows a Mouse-Wiggle action in which a participant wiggled the mouse side-to-side. The dotted line shows the mouse displacement (in pixels) along the direction of the wiggle (x-axis of the mouse), and the wrist data is linear acceleration along the x-axis of the wristband, which happened to align with the x-axis of the mouse. The figure on the bottom shows Tap-5x action, where a participant pressed the spacebar key five times. The highlighted region shows the duration of the keypress (i.e., KeyDown to KeyUp event) for each tap.

4.2 Protocol

There are six steps in the SAW protocol, as described below and shown in Figure 5. The figure also shows the wristband activation step, where Alice activates the wristband when she wears it at the start of her workday. After activating the wristband, the wristband serves as her authentication token, henceforth named as Alice's Wristband.

Bootstrapping Secure Communication. During the wristband activation step, Alice pairs her wristband with the target desktop, and as part of the pairing process, the desktop and the wristband share their public keys K_d and K_w, respectively, with each other. (In an enterprise setting, wristbands might be paired with desktops using a centralized system.) Using these keys, the desktop and the wristband establish a secure communication when they are in radio proximity. SAW uses established protocols like SlyFi [19] to prevent replay and MITM attacks, and for source authenticity (i.e., verifying the source of a message). With SlyFi, when Alice arrives within the radio proximity of the desktop, Alice's wristband and the target desktop establish a session key (k_w^s) using the public keys they had shared during the pairing process. This session key, which is shared only between the target desktop and Alice's wristband, is used to secure their communication, and the desktop thereby uses this key to bind any communication with Alice to Alice's identity, for the duration of this session. The following protocol describes how user verification works in SAW.

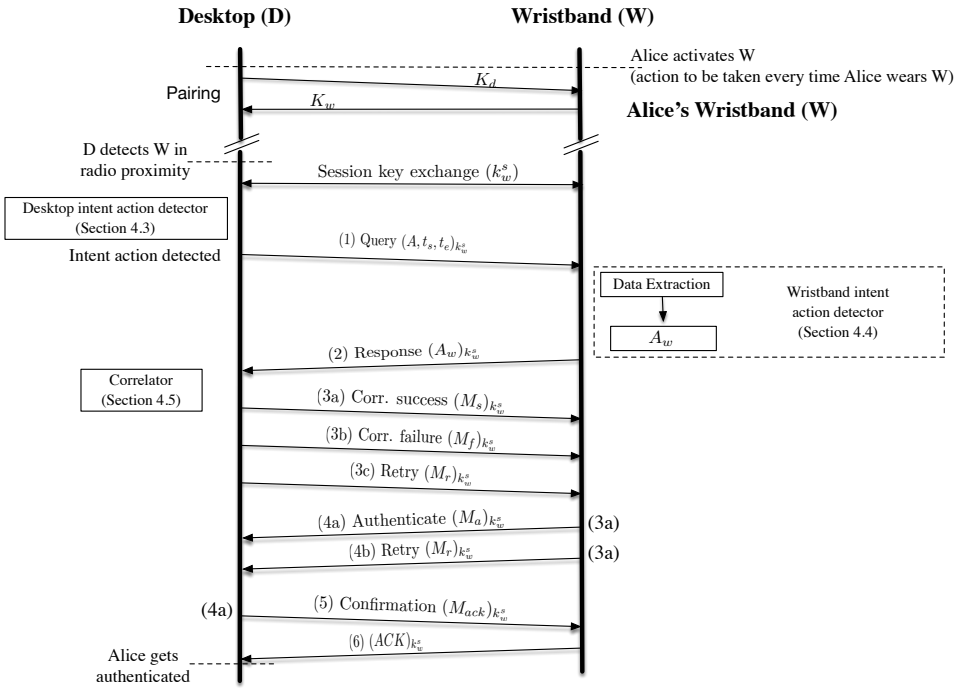

Fig. 5. SAW protocol. Numbers correspond to the steps in the protocol; the messages with the same number (but different letter) represent different responses the desktop/wristband can send when it receives a message, depending on their state and received message (e.g., when the wristband receives 3a, it sends either 4a or 4b depending on whether the user should be authenticated).

(0) Initial step: The authentication process is initiated when a locked desktop D detects keyboard or mouse input that resembles the intent action.

(1) Query step: Desktop D sends a query message to wristbands in radio proximity with start and end times (t_s, t_e) of the intent action relative to message transmission time, where $t_s < t_e < 0$. The receiving wristband extracts wrist movement data in the time window $(t_{rx} + t_s, t_{rx} + t_e)$, where t_{rx} is the message receive time. We consider wristbands only within 10 meters (standard Bluetooth range), determined using radio range, as candidate wristbands; this simple proximity threshold (filter) is an optimization in the protocol, and could be further constrained or relaxed depending on particular application use cases.

(2) Candidate response step: Among the nearby wristbands, each wristband determines if it is a *candidate* for this request (Section 4.4). If a wristband is a candidate, it detects the intent action in the extracted motion data and computes a set of correlation points A_w corresponding to the intent action (Section 4.5), which is then sent to the desktop for correlation with the desktop's intent action sequence A_d; if the wristband is not a candidate, it sends A_w as an empty set.

(3) Correlation step: Desktop D correlates all the wristband sequences (A_w) it receives with its own intent action sequence (A_d) to find the best-matching sequences.

(3a) If D finds only one match with high confidence, D sends a successful correlation message M_s to that wristband indicating that the user can be authenticated.

(3b) If the correlation fails for a sequence received from a wristband, D sends a failure message M_f to that wristband indicating failed authentication.

(3c) Disambiguation step: If, on the other hand, there are two (or more) wristbands that correlate with the intent action, D asks those users to repeat the intent action by sending a retry message M_r. Alternatively, D can fall back to a default authentication method like username and password.

(4) Wristband confirmation step: In this step the wristband confirms to the desktop that the user should be authenticated. This step ensures that the user is not authenticated to multiple desktops at the same time. In a many-to-one use case (many users, one desktop), a wristband can be a candidate for only one desktop. In a many-to-many use case, a wristband might be a candidate for two (or more) desktops; such a wristband sends its motion data to all such desktops, and waits for their response.

(4a) If the candidate wristband receives a message M_s from only one desktop, it responds with an okay-to-authenticate message M_a, which includes Alice's identity.

(4b) If the candidate wristband receives M_s from multiple desktops, the candidate wristband denies authentication to all desktops with a retry message M_r, indicating that the desktops should ask the user to authenticate again, as in the step 3c.

(4c) If the candidate wristband does not receive M_s, but receives M_f or M_r, it alerts the user, indicating that an authentication attempt was made and it failed, and she should try again.

(5) Desktop confirmation step: After D receives M_a from a wristband and it is ready to authenticate, it sends a message M_{ack} requesting a final confirmation from the wristband to authenticate the user.

(6) User authentication: The wristband sends an *ACK* confirming that the user can be authenticated. The wristband can also be configured to alert the user that she has been logged in to desktop D, or require a confirmation from the user before sending the final *ACK* to the desktop.

We assume the communication between the wristband and the desktop is reliable, i.e., the underlying MAC or other protocol layer in the communication stack handles message failures and guarantees message delivery. If the communication breaks between the wristband and the desktop, the authentication protocol aborts and the user is not authenticated by SAW. We discuss how this protocol helps achieve SAW's security goals in Section 6.3.

4.3 Desktop Intent Action Detector

When a locked desktop receives keyboard or mouse input, it determines whether the input is an intent action, i.e., Tap-5x or Mouse-wiggle. When a desktop is unlocked (i.e., some user logged in), it does not look for intent actions, and hence, does not initiate the authentication protocol even if an intent action is deliberately provided. Detecting an intent action on desktop is straightforward: Tap-5x is a sequence of five (or more) keystrokes on the same key in quick succession; Mouse-wiggle is a rapid side-to-side displacement of the mouse pointer along the x-axis with small displacement (if any) along the y-axis.

On receiving an intent action, the target desktop initiates the authentication protocol by sending a query message M_q to all the nearby wristbands. (Desktops that support SAW keep track of the wristbands that are in radio proximity.) The query message $M_q = (A, t_s, t_e)$ includes the type of action (A, Tap-5x or Mouse-wiggle), and the start and end times (t_s, t_e) of the action relative to when the query message is transmitted; the receiving wristband extracts the motion sensor data corresponding to this action by adding the relative start and end times to the message receive time. If T_s, T_e are the start and end time of the intent action and T_{tx} is the time when the desktop transmits the query message, $t_s = T_s - T_{tx}$ and $t_e = T_e - T_{tx}$.

4.4 Wristband Intent Action Detector

When a wristband receives a query message from a nearby desktop, it extracts motion data, determines whether the user is a candidate for authentication, and if the user is, the wristband proceeds to compute the correlation points that may correspond to the intent action.

Data Extraction. A wristband in SAW is always sensing motion data and keeps a buffer of the past 10 seconds, similar to fitness trackers and smartwatches. Based on the received message $M_q = (A, t_s, t_e)$ at time T_{rx}, the wristband extracts accelerometer and gyroscope data between the time window $T_{rx} + t_s - \epsilon$ and $T_{rx} + t_e$, where ϵ is to account for possible communication delays and clock skews (in SAW, we use $\epsilon = 0.5\ s$). The extracted motion sensor data should include the wristband movement during the intent action, but it does not have to precisely match the duration of the intent action. In other words, the extracted data can contain wristband movement data before or after the intent action, but it should encompass the intent action. The ϵ parameter ensures that the extracted data does include the start of the intent action, so that the desktop can determine whether the user is a candidate for authentication.

Candidate Detection. The candidate detection step determines whether the user intends to authenticate to a desktop, *any* desktop. There can be multiple authorized users near a target desktop, and this step acts as the first filter to eliminate users with wrist movement significantly different compared to a movement from a wrist that performs an intent action. Being a candidate user does not mean that this user is the one who provided the intent action on the target desktop. It implies that the user's wrist movement is *similar* to an intent-action wrist movement, and it is likely that the user performed an intent action. Whether the user's intent action is the *same* as the intent action on the target desktop determines whether the candidate user is the one who should be authenticated and logged in, and this is determined in the correlation step (Section 4.5).

SAW uses an activity classifier to identify wrist movement that looks similar to a wrist movement corresponding to intent actions; the classifier is trained to recognize 'walking', 'stationary', 'writing', 'other physical activity', and specific intent actions, 'Tap-5x' and 'Mouse-Wiggle'. The wristband computes the classification feature vector from the extracted sensor data and feeds it to the classifier. If the data is classified as one of the intent actions, the wristband (and its user) is considered as a candidate for authentication, and the wristband extracts correlation points from the sensor data. If the data is classified as a different activity, it implies that the user did not express an intent to authenticate, and should not be authenticated. To build the intent detection classifier, we use a Random Forest Classifier [12] and standard activity recognition features [27]. To train the classifier, we used Tap-5x and Mouse-Wiggle samples from our experiments, and for 'walking', 'stationary', 'writing', and 'other physical activity' we used wrist movement data collected from two volunteers who performed these activities while wearing a wristband.

Correlation Event Detection. For Tap-5x, the correlation events are key-press down (`KeyDown`) or key release (`KeyUp`) events, and for Mouse-Wiggle, the correlation events are the 180 degree changes in the mouse trajectory. These events generate small but sudden changes in wrist movement or rotation, which appear as peaks or troughs in the accelerometer and gyroscope sensor data. Thus, the wristband represents an intent action as a sequence of timestamps of peaks and/or troughs that likely correspond to the correlation events.

Depending on the wristband orientation and how the user performs the intent action, the peaks (or troughs) may appear as more prominent in one axis than others. So we find correlation points along each individual axis (x, y, z) in accelerometer (a) and gyroscope (g) signals; in total, we get twelve sequences, from six axes and a sequence of peaks (p) and troughs (t) for each axis. Thus the set A_w consists of twelve sequences representing the intent action:

$$A_w = \{S^p_{ax}, S^t_{ax}, \ldots, S^p_{gz}, S^t_{gz}\}$$

4.5 Intent Action Correlation

The desktop's intent action set A_d contains two sequences (`KeyUp` and `KeyDown` events for Tap-5x, and peaks and troughs for Mouse-Wiggle) that represent the intent action; the wristband's intention action set A_w contains twelve sequences. We do a pair-wise matching of sequences in both sets, compute the correlation score for a pair, and use the highest correlation score to determine whether A_d and A_w represent the same intent action, and whether the wristband user should be authenticated.

For a given pair of sequences of timestamps R (from the desktop) and S (from the wrist), this correlator module generates a correlation score to quantify how well the two sequences correlate. If the two sequences are of same length, then aligning and matching them is straightforward. The wrist signal, however, is noisy and may have more or fewer peaks (or troughs) than what we expect ($|R| \neq |S|$), in which case we want to match *corresponding* timestamps – timestamps that are closest to each other – in both sequences while penalizing missing or extra timestamps in the S sequence. Fortunately, the problem of sequence matching is well studied in bioinformatics. We adapt the Needleman-Wunsch algorithm, which is used to align protein sequences [40]. Given two sequences, the algorithm produces two aligned sequences with the maximum similarity score, for a given scoring matrix. For example, for sequences ABCDEF and ABCGF the algorithm would output ABCDEF and ABC−GF as the two aligned sequences. A similarity score for the sequences ABCDEF and ABC−GF is determined by comparing letters at each position: if two letters are the same, it is a *match* and a positive match value is added to the similarity score; if two letters are different (E and G at fifth position), it is a *mismatch* and a mismatch penalty is added to the similarity score; if there is an insertion or deletion (D and – at fourth position), a gap penalty is added to the similarity score. The match value, mismatch penalty, and gap penalty are defined by a predefined scoring matrix. The Needleman-Wunsch algorithm determines the best alignment using dynamic programming.

In SAW we use a variation of this algorithm to align two sequences of timestamps R and S. Two timestamps, t_1 and t_2 are considered a match if $|t_1 - t_2| \leq \tau$, where τ is the *matching threshold*; if the timestamp difference is greater than τ, they are considered mismatched. In the scoring matrix, a match is 1, a gap is -0.5, and a mis-match is -1. The algorithm computes the similarity score for the aligned sequences as the sum of all matches (number of matches × match score), all gaps (number of gaps in both sequences × gap penalty), and all mis-matches (number of mis-matches × mis-match penalty). We normalize this score by dividing it by the maximum possible similarity score for sequence R with any other sequence S. We use the normalized similarity score as the correlation score, c, where $c = 1$ indicates perfect correlation between the user's wrist movement and the keyboard or mouse inputs. If the correlation score is greater than the correlation threshold, τ_c ($c \geq \tau_c$), we consider the correlation good enough to authenticate the user.

5 DATA COLLECTION

We conducted two in-lab studies to evaluate SAW. The first study collected data to test the feasibility of the idea, and the second study sought to understand usability and user perception about SAW. Both the studies were approved by our university's IRB. (As noted above, we also conducted two pilot studies to explore different intent actions, before choosing Tap-5x and Mouse-Wiggle.)

5.1 Study 1: Feasibility Study

We recruited 17 participants for the feasibility study: ten male and seven female; eleven graduate students, four undergraduate students, and two college employees. We asked participants to wear a Shimmer device [49] – the SAW wristband for the purpose of our study. We sampled the accelerometer and gyroscope sensors in the Shimmer at 512 Hz, and streamed the sensor data to a desktop in real time, where it was logged to a file. To capture keyboard inputs to the desktop, we wrote a Python script that intercepts keyboard and mouse inputs from the desktop OS and logs them; we used Apple iMac desktops in both the user studies.

Wearing the Shimmer, each participant performed 20 interactions for each of the two intent actions (Mouse-wiggle and Tap-5x), with a brief pause between each repetition. All actions were performed with the wristband hand; participants could choose either hand to wear the wristband. For Mouse-wiggle, we instructed the participants to 'move the mouse side to side for a few seconds', and for Tap-5x, we instructed the participants to 'Tap a key, any key, on the keyboard five times'. We did not demonstrate or give them any specific instruction on how to perform taps or wiggle the mouse; we wanted to capture participants' natural interactions. We did, however, instruct them to assume that the desktop display is off and they have to wake up the display with their interactions; we envision SAW being used in this way. We collected two types of data for the Tap-5x and Mouse-Wiggle interactions performed by the participants: *a*) the participant's wrist movement, captured by the Shimmer, and *b*) the keyboard and mouse inputs received by the desktop, intercepted by the script we wrote.

Separately, we recruited two volunteers to wear Shimmer on their wrist and collected data for 10 min while sitting still on a chair ("sitting" activity); for 10 min while walking ("walking" activity); for 30 min while using a computer ("PC use" activity); and for 50 min while performing different activities (moving around, writing, reading, eating, etc.; "other" activity). We collected only 10 min data for sitting and walking activity, because that was enough to characterize those simple activities.

5.2 Study 2: User Perception Study

We recruited 16 participants for our second user study: ten male and six female; eight undergraduate students and eight graduate students. The goal of this study was to learn about users' perception of the usability and security of SAW. Participants were asked to fill out a survey about their demographic information and their current preferred authentication method. After the initial survey, participants were asked to perform 30 authentication attempts with three different authentication methods (discussed below). We then conducted a brief (15 min) semi-structured interview with the participants to gain insight into what they liked and disliked about each method. We took notes summarizing participant responses and writing direct quotes. Finally, the participants were asked to fill out a modified systems usability scale (SUS) survey for each of the three authentications methods. We modified the SUS survey [13] by changing 'system' to 'method', and removing the question 'I found the various functions in this system were well integrated', because it was not applicable to our authentication methods.

Participants are known to be biased in user studies and they try to give answers they expect the researcher desires [8]. To reduce this bias we chose participants who were not aware of this work, and had not participated in the feasibility study; we did not inform the participants that were evaluating a method that we developed. Instead, we told participants that we were evaluating two potential authentication methods that other researchers have proposed: 1) a mouse-based method (a fictitious method based on mouse movement dynamics, a behavioral biometric method), and 2) a tap-based method (SAW). We explained to the participants how both methods work. In the mouse-based method two objects are shown on the screen and the user drags one object over another; the user is authenticated based on how she moves the mouse – her unique mouse movement signature. In the tap-based method the user has to wear a wristband (which acts as the user's identity) and the user taps a key five times; the user is authenticated based on the timings of the taps that the wristband determines from the wrist motion and sends it to the desktop. We told participants that we are collecting data to study the feasibility of these methods and to get their feedback on their perception of the usability and security of these two methods.

In addition to the two authentication methods, we also asked participants to perform authentication attempts with a username and password, as a baseline for comparison. We simulated the authentication environment through a browser, where the participant was shown a webpage asking him/her to do a task and then click the login button to authenticate; the webpage had written instructions for the task. For username and password method, the task was to enter a username and password (chosen at the start of the experiment); for the mouse-based method, the task was to drag one rectangle (specified on the webpage and always the same rectangle)

over the second rectangle (both rectangles always appeared in same location on the webpage); and for the tap-based method, the task was to tap a key five times (any key; most users chose the space bar). We informed the participants that we would be timing each authentication attempt. During the experiment, the participants were asked to perform 30 attempts, 10 for for each method, in a randomized sequence (the sequence was same for each participant).

6 EVALUATION

SAW's goals are to be provide users with a quick and easy way to express their intent for authentication. Based on the data from the user studies, we evaluate how accurately SAW captures user intent (Section 6.1), its usability (Section 6.2), and that SAW indeed prevents accidental logins and raises the bar for adversaries to mimic and spoof the user (Section 6.3).

6.1 Feasibility

To evaluate SAW, we fed it with a number of test cases generated from the feasibility study (Section 5.1); a test case is a combination of wristband motion data (W) and desktop input data (D). Some of the test cases are positive, where W and D from the *same* action perform by the same participant, and some are negative, where W and D from *different* actions performed by same or different participant. SAW evaluates all the test cases and labels them as positive or negative. A perfect system would correctly label all positive test cases as positive and all negative test cases as negative. We use false-negative rate and false-positive rate as our metrics for evaluation. False-negative rate (FNR) is the fraction of positive test cases that SAW misidentified as negative. FNR tells us how effective a method is at authenticating users, in other words, how frequently the method makes an error and denies access to a user; we want FNR close to zero. False-positive rate (FPR) is the fraction of negative test cases that SAW misidentified as positive. FPR tells us how effective a method is at stopping adversaries, in other words, how frequently the method makes an error and accidentally grants an unauthorized individual access to the computer; we want this number close to zero. We discuss FNR results here and FPR results in Section 6.3.

SAW detects user intent in two phases. First it detects a user's intent to authenticate, then it determines whether the user should be authenticated. We discuss each, next.

6.1.1 Accuracy of Intent Detection on the Wristband. SAW uses an activity classification model to identify a specific intent action based on the accelerometer and the gyroscope data. During the authentication protocol, the desktop informs the wristband which intent action it received, and the wristband uses the classifier trained to identify that intent action. To train a model, we need *positive* samples (user performing the intent action) as well as *negative* samples (user not performing the intent action, i.e., doing other activity). For positive samples, we used wristband (Shimmer) data when participants performed the intent action. For the negative samples, we chose wristband data from the activity dataset (Section 5.1).

To test the classifier we used leave-one-out validation: for each participant, the classifier is trained with data from participants other than the one being tested. This ensures that the model is not specific to the user's pattern of performing the intent action. For each participant, we trained the model and then tested it on the participant's intent-action wristband data. The average false negative rate (FNR), i.e., failure to identify an intent action, across all participants, was 0.002 ± 0.003 (mean $\pm$ standard deviation) for Tap-5x and 0.009 ± 0.005 for Mouse-Wiggle. This means that in about 1,000 authentication attempts, SAW failed to detect a user's Tap-5x intent to authenticate twice, and the user had to try again.

6.1.2 Accuracy of Authentication. We fed SAW with all positive test cases from our feasibility study and measured its accuracy in authenticating a user, i.e., identifying user intention on the wristband *and* successfully correlating the wristband and desktop data. If the correlation score (c) for a test case was greater than the threshold (τ_c),

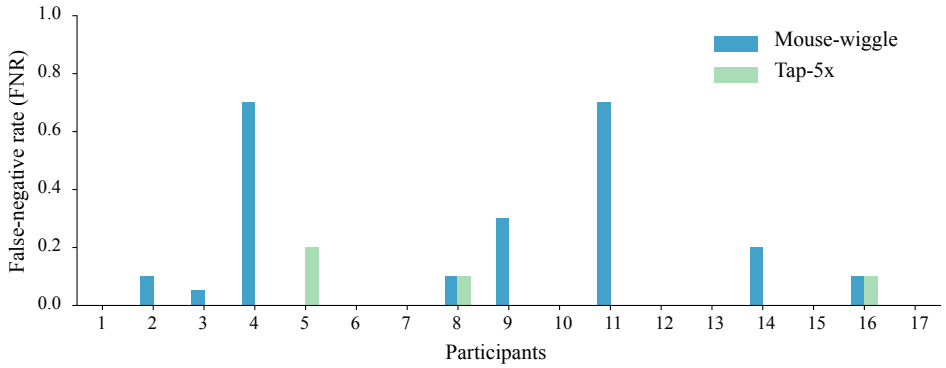

Fig. 6. False-negative rate (FNR) for study participants with Mouse-wiggle and Tap-5x interactions.

SAW considered that as a successful correlation and authenticated the user. We evaluated a range of threshold values by computing the false positive rate (FPR) and false negative rate (FNR); for our dataset, SAW performed best (in terms of both FPR and FNR) with τ_c at 0.7. (A lower threshold reduces FPR but increases FNR, and a higher threshold reduces FNR but increases FPR.) All the values we report are for $\tau_c = 0.7$.

The mean FNR, across all participants, was 0.025 (± 0.061) for Tap-5x and was 0.13 (± 0.22) for Mouse-wiggle. For comparison, password FNR is about 0.079 (± 0.091) for laptop and desktop authentication [34]. Figure 6 shows the Tap-5x and Mouse-wiggle FNR for individual participants. Mouse-wiggle FNR was high for two participants in particular, P4 and P11. Inspecting their data, we found that these participants did not wiggle the mouse side-to-side: P4 moved the mouse arbitrary and P11 did not wiggle the mouse at all. The Mouse-wiggle intent action does require the user to move the mouse a certain way. With Tap-5x, however, none of our participants had any similar issues getting their taps recognized; P5 did show a relatively high FNR (0.2) and that is because she tapped gently with a relatively immobile wrist. We emphasize that in our experiments we did not instruct participants on how to perform Mouse-wiggle or Tap-5x, because we wanted to evaluate how well SAW performs for participants without any training and with participants' natural interaction styles. With Tap-5x, SAW's FNR is much lower than that of passwords [34]; as participants learn these intent actions, SAW would perform even better.

In our feasibility experiments, the Tap-5x authentication interaction proved more quick and accurate than the Mouse-wiggle action, so we chose Tap-5x as the default authentication action for SAW, and discuss results below with respect to Tap-5x.

6.2 Usability

Below we evaluate the usability goals we presented in Section 3.1.

6.2.1 No Need to Memorize in SAW. By design, SAW is memoryless – there is no secret for the user to remember. The user does have to remember to wear the wristband, but once worn the wristband stays with the user. The goal is minimize the number of password authentications in a day using SAW, but SAW uses password as a backup authentication method and to activate the wristband, so the user does have to remember the password. But the user does not have to recall the password as frequently, and there is no need to memorize any new secret.

6.2.2 SAW is Quick. Authentication time in SAW is the sum of communication latency, computation time, and the time required to perform the intent action. Before an authentication attempt, the desktop and the wristband

Table 1. Time (in seconds) participants took to complete 30 iterations of each of the three authentication methods; mean (standard deviation).

Participant	Password	Mouse-based	SAW
1	9.7 (7.5)	3.4 (0.8)	3.5 (0.4)
2	9.0 (3.1)	7.7 (4.2)	4.4 (1.2)
3	8.0 (1.2)	7.4 (1.3)	4.5 (0.8)
4	10.8 (3.1)	5.3 (0.9)	4.3 (0.4)
5	10.2 (2.3)	4.4 (1.3)	3.5 (1.1)
6	7.7 (1.0)	5.0 (1.6)	3.5 (0.5)
7	10.2 (5.2)	4.0 (0.5)	3.4 (0.4)
8	9.8 (1.8)	4.6 (0.6)	3.2 (0.2)

already have established a secure connection, and thereafter communication latency is negligible. In our tests, computation (performed on a laptop with 8 GB RAM and 2.6 GHz Intel core i7 processor) took at most 500 ms. Based on the data from the feasibility study, participants took on average of 1.5 s and 4 s to perform Tap-5x and Mouse-wiggle, respectively. Thus, overall, SAW takes about 2 s with Tap-5x and 4.5 s with Mouse-wiggle. In the user perception study, however, we found that participants took about 4 seconds to complete Tap-5x, which is still much less compared to the time they took to type their username and password. Table 1 shows the average time participants took to complete 30 iterations of the three authentications. Due to an experimental error, we failed to record data for the other eight participants.

6.2.3 SAW is User Agnostic. A user-agnostic method does not depend on a user's unique characteristics, e.g., in the context of computer use, how she types or how she moves a mouse. User-agnostic methods offer two advantages: 1) they do not need user-specific training, and 2) they are resilient to changes in user behavior. We evaluated SAW in a user-agnostic fashion by using leave-one-out validation, that is, when testing SAW for a participant, we did not use that participant's data to train any component in SAW. Thus, SAW is user agnostic.

6.2.4 SAW is Low Effort and Easy to Perform. The SUS scores for the passwords, the fictitious mouse-based authentication method, and the tap-based method (a.k.a. SAW) are shown in Figure 7. Since we used nine questions in our SUS survey, we report the usability scores out of a total of 90. A higher SUS score indicates that a system is more usable. Although a majority (11 out of 16) of the participants rated SAW to be more usable than passwords (median for passwords 70 vs. SAW 75), we see a high variance in the data (means ± std for passwords 69 ± 10 vs. SAW 71 ± 13). Eleven out of sixteen participants rated SAW more usable than passwords, four rated SAW as less usable than passwords, and one participant felt both methods were equally usable.

When we probed participants about their ratings, most of the participants who favored SAW said they liked it because it was quick and easy to perform taps.

"Tapping the space bar was the easiest." (P10)

"Tap-based method was the fastest for me." (P16)

Some participants felt SAW was non-intrusive and that it does not require any mental effort.

"It's very non intrusive. I don't have to think about authentication." (P4)

"It is quick and simple. There is no need to watch the screen or keyboard." (P14)

"I liked the tap-based method because I tap a key 2-4 times to wake-up my computer." (P15)

A few participants felt that counting taps till five was an effort, and two participants suggested reducing the number of taps to three from five.

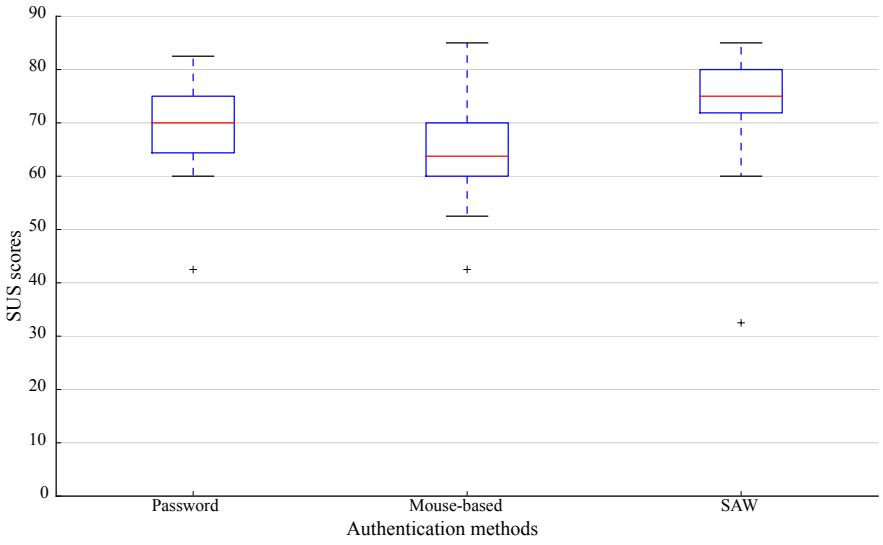

Fig. 7. System usability scale (SUS) scores for password-based authentication, mouse-based authentication, and SAW.

"I lost track of how many times I tapped." (P11)

Four participants rated SAW less usable compared to passwords. When inquired, one participant (P6) said he simply did not like that he would have to wear a wristband. One participant said he liked the convenience of tapping but he was not confident in the security of the method, and similarly two other participants said tapping was easier for them, but rated SAW less than passwords in the survey, expressing concerns about the security of the method. We thus suspect some participant's concerns about the security of the method were reflected in the usability survey. This experience suggests that for security-critical systems (or methods) the SUS survey should be administered carefully, delineating the participants' security concerns and asking them to focus on the usability aspects of the system.

Although several participants confessed that performing the mouse-drag movement was simpler, they rated mouse lower than passwords (and lower than SAW), because they were more comfortable and quick typing than using a mouse.

"Using the mouse felt like a task to me." (P2)

We also asked participants to rate how secure they thought SAW and mouse-based method would be compared to their current method (which was password for everyone). Twelve out of sixteen participants felt SAW was almost as secure or more than passwords. One participant explained that she felt SAW was more secure because of the physical device (wristband) that stays with her.

"Having the wristband makes it feel more secure" (P1)

Four participants rated SAW as less secure than passwords; when we inquired, they expressed lack of confidence in the method.

In summary, majority of our participants liked SAW for its quickness and simplicity, but some participants had concerns about its security and the burden to wear something. We expect people who already wear a fitness band or a smartwatch would be willing to use SAW. Getting an objective usability comparison from users between a

new unfamiliar method and a well established, understood, and regularly used method – passwords – is difficult. We believe the result that participants rated SAW to be as usable as passwords speaks in favor of SAW.

6.3 Security

By design SAW *Requires-explicit-intent* (Goal S1) and there is no visible secret information exchanged between the user and the desktop, so SAW is *Resilient-to-physical-observation* (Goal S2), unlike passwords, which can be easily stolen by an observer.

6.3.1 Resilience to Accidental Logins. In this attack, we consider the case that a (victim) user does not wish to authenticate to any desktop, but she gets logged in to the (adversary's) target desktop. When the victim user is in the radio proximity of the target desktop, the adversary attempts to log in as the victim user. When the adversary provides the intent action, SAW will attempt to authenticate the victim user with the desktop, and send a query to the victim user's wristband. Whether the user will be authenticated depends on the victim user's wrist movement, which in turns depend on what she is doing when the intent action was performed; she may be walking, sitting, using another desktop, or doing some other activity. To evaluate precisely this type of scenario we collected activity data in our experiment. All the test cases for this scenario are negative test cases, generated with D (desktop inputs) from intent actions performed by participants (as the adversary's input) and W (wristband data) from a randomly chosen sample for an activity from the activity dataset (as the victim user's wrist movement). In total, we generated 3,000 negative test cases.

We first evaluated these negative test cases with the wristband intent detector and if the intent detector labeled any test case as positive, we evaluated those with the correlator. The intent detector's FPR for sitting, walking, and other activity was zero, 0.013, and 0.151, respectively. Overall, the intent detector correctly identified 85% of the test cases as negative, i.e., it correctly identified that the victim user did not intend to authenticate in 85% of the cases. About 15% of the wristband samples during walking and other activities were misclassified by the intent detector as valid intent actions, but these 15% test cases were identified as negative by the correlator, producing zero FPR for this attack in our experiments.

6.3.2 Resilience to Mimicking Attack. A majority of the accidental login attack test cases failed (from the adversary's perspective) due to the intent detector, which emphasize SAW's *Requires-explicit-intent* design goal. This design goal raises the bar for an adversary, because now the adversary not only has to account for the user's presence but also for the user's wrist movements. In a mimicking attack, the victim user is near the (adversary's) target desktop and is attempting to log in to a different desktop, while the adversary times his intent action to match and mimic with the victim user's intent action.

Before we evaluate this scenario, let us see how SAW works in a multi-user multi-desktop setting. Consider a scenario with two users, Alice and Bob, who want to log in to desktops D1 and D2, respectively. Both Alice and Bob are authorized users of both the desktops. When they attempt to log in at the same time, D1 sends out a query to nearby wristbands and receives data from Alice and Bob's wristband, and attempts to identify which of the two attempted to log in; and D2 does the same thing. As long as the time delay between when Alice and Bob performed their authentication interaction is more than 50 msec, D1 can easily distinguish them. But if Alice and Bob performed authentication interactions at the same time (or within 50 msec) – an unlikely case – there is a possibility that D1's correlation scores for Alice and Bob are above the threshold and D1 cannot decide which of the two is its current user. In this dilemma (an "authentication collision") D1 simply refuses to authenticate either user, and asks its current user (Alice) to try again. Alternatively, authentication collision can happen for the wristband: D2 and D1 may fail to authenticate Bob, but Alice might pass the correlation threshold on D1 and D2, in which case, Alice's wristband must decide which of the two desktops Alice is using – the wristband can only authenticate to one desktop at a time. In this situation, Alice's wristband rejects authentication from D1 and D2,

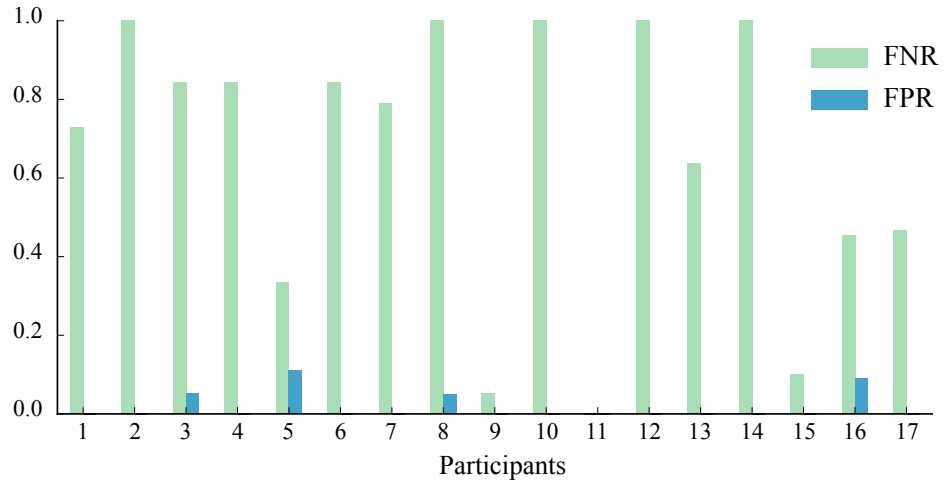

Fig. 8. False-positive rate (FPR) and False-negative rate (FNR) for the mimic attack case, where an adversary attempts to spoof the user by mimicking her actions; FPR should be low in attack scenarios.

and D1 will ask its current user (Alice) to authenticate again. We expect such collisions in authentication attempts are unlikely in real life, so having to retry again in such situations is not a significant trade-off for usability.

The mimicking attack scenario is similar to the above case, except that Bob is an adversary who wears no wristband and is not an authorized user of D2 (the target desktop). Using the interaction data from our user study, we generate test cases for this scenario as follows. For the negative test case, we use W (wristband data) from a participant, say Alice (in the role of the victim user) and D (desktop data) from another participant, say Bob (in the role of the adversary), and we align the timestamps so both the data streams begin at the same time, as if Bob timed his desktop inputs to match perfectly with Alice's; and for the positive test case, we use W and D from Alice. We repeat this experiment for each participant in the role of victim user and a different participant (selected at random) in the role of adversary.

Figure 8 shows the average FPR and FNR for each participant as Alice. SAW achieves a low FPR of 0.018 ($\pm$ 0.036) across all participants for the mimicking attack. The FNR rate, however, is high (more than 0.5), and this is because resetting the timestamps in the test cases caused collisions in authentication for the victim user – she was being authenticated on two desktops, the target desktop and her own desktop – and the victim's wristband denied authentication to her on both desktops in that attempt. For an attack scenario, we want a low false-positive rate (which SAW achieves) and preferably a low false-negative rate, but a high false-negative rate is acceptable because such attack scenarios are unlikely in practice – in real settings we expect at least a few seconds difference between the two individual's login attempts.

By aligning the timestamps we evaluate SAW's performance for an adversary who is a perfect mimicker. This is a stress test for SAW because in practice it is difficult for an adversary to imitate others in real time. Prior work on mimic attacks on gait and keystroke biometrics suggest that humans can imitate others with limited success, but under certain conditions (e.g., not in real time, mimicker and target both have similar physical characteristics, the mimic task is simple) and with significant training [29, 38]. In SAW the adversary needs to react within 50 ms, and given that the average human reaction time is about 215 ms [26, 47, 55], mimicking SAW's intent action in real time is difficult.

If an adversary does manage to mimic a victim user, then the user's wristband intent action would correlate with two desktops – the desktop the user is attempting to use and the target desktop on which the adversary is mimicking the user – resulting in an authentication collision, and the wristband will deny authentication on both the desktops. An adversary who can cause wireless interference can block communication between the victim user's wristband and her desktop at the right moment to avoid authentication collision and log in to the target desktop as the victim user. In that case, the victim user would be notified on her wristband that she logged in to a desktop (Step 5 in the protocol, Section 4.2), allowing her to track down the desktop and take action, reducing the attack window.

6.3.3 Protocol Analysis. SAW uses SlyFi [19], a secure communication protocol, for desktop-wristband communication and relies on SlyFi's security guarantees against message replay, message spoofing, and man-in-the-middle attacks. As describe in Section 4.2, the desktop and the wristband bootstrap their session by sharing session keys. Because the desktop uses a different session key for each wristband, the desktop can verify the message source, which prevents multi-session attacks where a message or a part of the message from one session is used in another session.

An adversary may selectively jam one or more messages in the protocol to gain advantage in spoofing the user, but the SAW protocol is secure against message failures in the protocol. If the desktop does not receive a valid message from the wristband in each step of the protocol, the desktop aborts the authentication protocol, and the user has to try again. Furthermore, because the desktop does not log in a user until it receives a confirmation ACK from the user's wristband (Step 6 in the protocol), the wristband is always aware where (on which desktop) the user is (or could be) logged in. This confirmation step prevents the case where an adversary manages to spoof the user but blocks messages to the wristband to keep the wristband notifying the user that she has been logged in.

In SAW's threat model, an adversary can be an authorized user of the desktop, i.e., he can have a SAW wristband and can participate in the authentication protocol. By participating in an on-going authentication protocol, such an adversary learns the type and time of the intent action performed on the desktop, but this information does not give the adversary any advantage to spoof the user in this particular on-going authentication attempt, because the intent action has already been performed and the adversary cannot change the desktop input at this point. (SAW uses three inputs to verify a user: proximity range (to verify that the user is within range), desktop input, and wristband motion data. To spoof a user, the adversary can only control proximity range and the desktop input.) The adversary, however, could use the intent type and time information to learn about the user's preference for intent action and pattern, which may help the adversary better mimic the user in future spoofing attempts. That said, this information could also be gained by physically observing the user while she authenticates, so the protocol does not leak any additional information about the user.

6.4 Feedback from Hospital Staff

To get feedback on SAW from users in a hospital setting, we conducted two short (about 30 min each) informal focus groups at a local hospital; one with a group of five clinicians, and another with a group of five IT staff members and one clinician. Scheduling interviews with hospital staff was challenging due to their busy schedules, and conducting usability studies in the hospital was logistically infeasible. So we resorted to seeking feedback through focus-group style meetings, during which we demonstrated SAW with a prototype, explained how authentication works in SAW, and asked for feedback and questions.

All participants really liked that SAW used a wrist wearable as the authentication token, primarily because they saw a wrist wearable as a better design than keycards that users forget all the time. Once worn, wristbands are likely to stay with (on) the user, unlike keycards or other tokens that people have to carry in their pockets or purses, which are likely to be left behind. In SAW, the authentication token stays *"tethered to their [users] wrist,"*

as one of IT staff put it. Our design choice to explore authentication based on a wrist wearable was in part due to similar feedback (around forgetting cards) that we received during our early conversations with the hospital staff.

One concern clinicians had about the wearable design was *"is it washable?"* According to them, wristbands should be water and alcohol resistant, basically resistant to all the chemicals that clinicians hands are frequently exposed to. One surgical clinician reported *"my rubber wristband eroded because of all the washing [with chemicals]."* Thus, the wristband may need to be made water/alcohol proof or removed before washing and in certain procedures (such as surgery or any other area requiring scrubbing), but we note that it is common for clinicians and nurses to wear a wristband (or watch), at least in the U.S. In countries (e.g., U.K., Australia) where hospitals use a 'bare below the elbows' policy and do not allow any jewelry below elbows, SAW may not be appropriate. That said, we anticipate SAW can be useful in a wide range of enterprise settings, not just hospitals.

Participants, especially clinicians, liked that the authentication was fast and did not need any password or secret PIN. The IT staff was keen to leverage the token for automatic deauthentication, i.e., automatically logging out users when they leave a workstation, which was one of their main security challenges; SAW can easily support automatic deauthentication based on user's step count after log in. Clinicians asked how SAW would authenticate when they forget their wristband, and they found the strategy of using passwords as backup authentication acceptable. A requirement that one clinician raised was the ability to have *read-only* access to a workstation from far away. His particular use case was to be able to watch a graph or a particular screen as he walked by, and he found *"having to login/logout just for that is annoying."* Instead of logging out a user, putting the workstation in read-only mode (with appropriate measures to protect patient privacy) could be a solution to address this particular use case.

7 DISCUSSION AND LIMITATIONS

Here we discuss important issues related to the feasibility of deploying SAW in a real system, some limitations of SAW, and areas of future work.

Identity and Securing Wristband to User. SAW does not identify a wristband wearer or provide resilience to theft. However, fitness wristbands and smartwatches are considered personal wearable devices usually intended to be worn more or less continuously and are less likely to be shared with others and hence are more likely to be kept under close control by their owners. Furthermore, smartwatch manufacturers could securely link smartwatches to users (to prevent theft and unintended sharing) by adding simple methods to detect when the user takes off the smartwatch and requiring her to enter a PIN when she puts the smartwatch on (as in the Apple Watch for example) or using a biometric to identify the user when she puts on the smartwatch [14].

When using motion data from a user's wristband, there are potential privacy concerns because the user's activity or what she typed can be inferred using the motion data [54]. Such privacy concerns, however, can be mitigated by processing the motion data locally on the wristband [46] and sharing only the processed data, as SAW does.

Energy Consumption on Wristband. The energy consumption of the accelerometer, gyroscope and radio on the wristband will determine the recharging interval, with longer times being preferable. Using hardware and software optimizations, a wristband running SAW could last days if not weeks on a single charge. Gyroscopes use more power (e.g., the STMicroelectronics L3GD20H draws 5mA), but the ADXL362 (and many other low power accelerometers) has an extremely low power motion-activated wake-up mode (270 nanoAmps) that could be used to keep both sensors and the radio off until motion above a preset threshold is detected. Low-energy radios such as Bluetooth Low Energy can run for months on a button-sized battery and the radio could be used both as a test for rough proximity to the computer terminal and to wake the sensors when proximity is achieved. Together, methods such as these should provide a battery life for a wristband of weeks, and as a background task on a smartwatch similarly consume little power.

Limitations. One of the main limitations of SAW is that Bluetooth, which is the most ubiquitous wireless protocol for personal area networks, was not designed for complex network topologies. As a result, pairing multiple desktops with a single wristband, or more generally, multiple devices with multiple wristbands, can be challenging. This limitation is not fundamental; there are existing ultra-low power wireless protocols (such as ANT+ and TI's SimpleLink) that support many-to-many connectivity in high-density environments. Bluetooth Low Energy, however, does not require pairing, and indeed, supports some broadcast operations (e.g., those used for iBeacon and Eddystone protocols). The Bluetooth SIG Smart Mesh Working Group is currently working to develop low-energy Bluetooth smart-mesh networking support [10]. These features should migrate into commercial smartwatches in the near future. Smartwatches support Wi-Fi as well, which may meet the needs of SAW since radio interactions with the terminal may only occur over a small portion of each day.

8 RELATED WORK

We present a comparative evaluation of SAW and other authentication methods using the UDS evaluation framework [11]. In the UDS framework, a set of usability and security properties (called 'benefits') are defined, and each authentication scheme is rated as either offering or not offering the benefit; if a scheme *almost* offers the benefit, but not quite, it is indicated with the *Quasi-* prefix. For our evaluation, we use the following thirteen benefits (seven usability and six security; denoted with U and S), drawn from the UDS framework and our design goals; for UDS benefits, we use the UDS definition.

- U1 *Memorywise-Effortless*: Users of the scheme do not have to remember any secret at all.
- U2 *Nothing-to-Carry*: Users do not have to carry any additional physical object to use the scheme. We grant *Quasi-Nothing-to-Carry* if the object is something that they would carry everywhere all the time anyways, e.g., wristband, smartwatch, or a phone.
- U3 *Physically-Effortless*: The authentication process does not require physical (as opposed to cognitive) user effort beyond, say, pressing a button. Schemes that do not offer this benefit include those that require typing, scribbling or performing a set of motions. We grant *Quasi-Physically-Effortless* if the user's effort is limited to speaking, on the basis that even illiterate people find that natural to do.
- U4 *Easy-to-Learn*: Users who do not know the scheme can figure it out and learn it without too much trouble, and then easily recall how to use it.
- U5 *Efficient-to-Use*: The time users must spend for each authentication is acceptably short.
- U6 *Easy-Recovery-from-Loss*: Users can conveniently regain the ability to authenticate if the token is lost or the credentials forgotten.
- U7 *User-Agnostic*: The scheme does not depend on users' individual characteristics. Biometric and behavioral schemes do not offer this benefit.
- S1 *Resilient-to-Physical-Observation*: An attacker cannot impersonate users after observing them authenticate one or more times. Attacks include shoulder surfing, filming the keyboard, recording keystroke sounds, or thermal imaging of keypad.
- S2 *Resilient-to-Leaks-from-Other-Verifiers*: Nothing that a verifier could possibly leak can help an attacker impersonate the user to another verifier. This penalizes schemes where insider fraud at one provider, or a successful attack on one back-end, endangers the user's accounts at other sites.
- S3 *Resilient-to-Phishing*: An attacker who simulates a valid verifier cannot collect credentials that can later be used to impersonate the user to the actual verifier.
- S4 *Resilient-to-Theft*: If the scheme uses a physical object for authentication, the object cannot be used for authentication by another person who gains possession of it. We still grant *Quasi-Resilient-to-Theft* if the protection is achieved with the modest strength of a PIN.

S5 *Requiring-Explicit-Intent*: The authentication process cannot be started without the explicit consent of the user.

S6 *Resilient-to-Relay-Attacks*: An attacker cannot break the scheme using simple range-extending or relay attacks.

Benefits *User-Agnostic* and *Resilient-to-Relay-Attacks* are based on SAW's design goals (Section 3.1) and the other benefits are based on the UDS framework. Below we discuss related work and rate them for these benefits; Table 2 shows a summary of our comparative evaluation.

Table 2. Comparative evaluation of SAW and other authentication schemes.

		Usability							Security					
Scheme	References	*Memorywise-Effortless*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Easy-Recovery-from-Loss*	*User-Agnostic*	*Resilient-to-Physical-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*Require-Explicit-Intent*	*Resilient-to-Relay-Attacks*
SAW		●	○	○	●	●	○	●	●	●	●	○	●	●
ZEBRA	[35]	●	○	≡	●	≡	○	●	●	●	●	○	●	●
Passwords		≡	●	≡	●	●	●	●	≡	≡	≡	≡	●	●
Proximity-based														
Using NFC or Bluetooth	[6]	●	○	●	●	●	○	●	●	●	●	○	≡	≡
Biometric														
Fingerprint	[48]	●	●	●	●	●	≡	≡	●	≡	≡	≡	●	●
Voice	[2]	●	●	○	●	●	≡	≡	≡	≡	≡	≡	≡	≡
Face	[3]	●	●	●	●	●	≡	≡	≡	≡	≡	≡	○	●

● = offers the benefit; ○ = almost offers the benefit; *no circle* = does not offer the benefit.
|||| = better than SAW; ≡ = worse than SAW; *no pattern* = equivalent to SAW.

8.1 ZEBRA

SAW builds on the *bilateral authentication* concept we introduced in our prior work on continuous authentication, ZEBRA [35]. In bilateral authentication the user's interaction with a desktop is observed by two parties – the desktop and the user's wristband – who then compare their observations to verify that the interaction to the desktop indeed came from the wristband's wearer. Although both SAW and ZEBRA authenticate by correlating user's wrist movements with desktop inputs, the key differences in the choice of the authentication interaction and

correlation techniques makes one (SAW) suitable for initial authentication and the other (ZEBRA) for continuous authentication. ZEBRA authenticates a user by comparing a sequence of interactions, where interactions are classified into three types: *Typing* (user typing on the keyboard), *Scrolling* (user scrolling the mouse), and *MKKM* (user switching between mouse and keyboard). The desktop converts the user's input (to keyboard and mouse) into a sequence of these three interactions – for example, TTMS, for Typing–Typing–MKKM–Scrolling. (In a sequence, each interaction is limited at most 1 s long duration; a longer interaction is split into small duration interactions of same type.) The user's wristband – based on the user's wrist movement – produces a corresponding sequence of interactions that it predicts the user performed, say TTMT (in this case, the wristband inferred the last interaction incorrectly). ZEBRA compares the two sequences and if the majority of the interactions match, the user is authenticated. This is a coarse-grained correlation, because all typing interactions are considered the same (similarly, all scrolling actions are considered the same and all MKKM interactions are considered the same); and matching of two sequences is done by majority, without any penalty for mis-matched interactions in the sequences. One drawback of this coarse-grained correlation is the possibility of a spoofing using a relay attack from an opportunistic human observer [23], because ZEBRA cannot distinguish between two typing interactions. Another drawback of ZEBRA is the time required for an authentication decision: as the paper reports, ZEBRA performs best with sequences of 20 or more interactions, which means it can take up to 20 s to authenticate the user. Thus, ZEBRA is not suitable for initial authentication, i.e., to unlock a computer, which is why it relies on an existing method to unlock the desktop.

Unlike ZEBRA, SAW authenticates a user based on just *one interaction*, tapping or mouse-wiggle interaction (called Tap-5x or Mouse-wiggle, respectively). SAW performs correlation using key moments in the interaction, which enables it do fine-grained correlation and distinguish between two different instances of the same interaction (e.g., Tap-5x performed at different times by same user, or at same time by different users). For tapping (Tap-5x), SAW looks at individual keystrokes and matches them with wrist movement based on the keystroke timings; for mouse wiggle (Mouse-wiggle), SAW looks at the changes in mouse direction and compares them with the changes in wrist movements. This fine-grained correlation allows SAW to authenticate a user just after a few keyboard or mouse inputs, authenticating users quickly, within 2 s. SAW thereby addresses both the challenges that make ZEBRA unsuitable for initial authentication. Because both SAW and ZEBRA use a wristband, they are actually complementary to each other – SAW does initial authentication and ZEBRA does continuous authentication; a user can authenticate to a desktop using SAW, and once authenticated, ZEBRA could continuously verify the user's presence with its continuous authentication method.

ZEBRA is *Memorywise-Effortless*, *Easy-to-Learn*, *User-Agnostic*, and it offers *Quasi-Nothing-to-Carry* and *Quasi-Easy-Recovery-from-Loss*. Because there is no secret shared during authentication, ZEBRA is resilient to physical observations, leaks from other verifies, and phishing, but offers only *Quasi-Resilient-to-Theft* because a wristband can be stolen. Because using a desktop conveys user's intent, ZEBRA offers *Requiring-Explicit-Intent* and *Resilient-to-Relay-Attacks*, but it is susceptible to simple mimicking attacks [23]. Compared to SAW, ZEBRA is not *Physically-Effortless* and *Efficient-to-Use* because a user has to type and move a mouse for 10 s to 20 s to authenticate.

8.2 Password-based Methods

Passwords are the most commonly used authentication method for desktop computers. Passwords are not *Memorywise-Effortless*. They offer *Nothing-to-Carry*, but are not *Physically-Effortless* as passwords need to be typed. Passwords are *Easy-to-Learn* and *Efficient-to-Use*, and they offer *Easy-Recovery-from-Loss* as they can be easily reset. They are *User-Agnostic* because they do not depend on either physiological or behavioral characteristics of a user. Passwords are not *Resilient-to-Physical-Observation* as they can be easily captured by shoulder surfing [52] or by filming [7], and for the same reason (i.e., they can be easily stolen) they are also not *Resilient-to-Theft*. They are not *Resilient-to-Leaks-from-Other-Verifiers* if the user has same password for different computers. They are

also not *Resilient-to-Phishing* as the attacker can use the obtained password to impersonate the user. They offer *Requiring-Explicit-Intent* as users have to enter their passwords to authenticate, and are *Resilient-to-Relay-Attacks* assuming that the adversary cannot tamper with the target desktop to intercept and relay the password.

8.3 Proximity-based Methods

In proximity-based authentication methods, a user carries a wireless authentication token and whenever the token is within a certain distance of the target computer, the user is authenticated. Commonly used proximity tokens include NFC cards, smartwatches [4], wristbands [42], rings [41], or smartphones [32]. Different tokens provide different affordances. NFC cards are flexible to carry (in a pocket or on a lanyard) and do not need charging, but they are also easy to lose or forget; lost (or stolen) cards present security risk because they still allow access as the user (until they are reported as lost and deactivated). Smartphones are less likely to be left behind and the user need not carry a separate token, but they can be inconvenient to be used as tokens if the user has to take her phone out of pocket or purse and unlock the phone to authenticate. Wearable tokens (watch, wristband, ring) offer better security, because, once worn, they stay with the user and they can be designed to detect when they are taken off so that they can be deactivated. In terms of deployment, a NFC-based solution requires issuing a (low-cost) card per user, but it also requires issuing a NFC reader per desktop; whereas SAW does not require any additional hardware on the desktop, but it does require a wristband per user (or could be incorporated in wristbands or smartwatches that users would use anyways).

As discussed before, the commonly used protocols in proximity-based methods are susceptible to relay attacks (NFC, Bluetooth, and Wi-Fi). There are secure distance-bounding techniques that are difficult to fool [45], but those techniques require special hardware, which is not common in consumer desktops and laptops. Apple's auto unlock method uses both Bluetooth and Wi-Fi as a defense against relay attacks, but it requires a special Wi-Fi chipset (802.11v compatible) [5]. In SAW, there is no need for any special hardware on the desktop or in the wristband (assuming both already have either Bluetooth or a standard Wi-Fi radio).

Proximity-based methods are *Memorywise-Effortless*, *Physically-Effortless*, *Easy-to-Learn* and *Efficient-to-Use*. We rate them *Quasi-Nothing-to-Carry* because the proximity tokens can be integrated in smartwatches or smartphones, and *Quasi-Easy-Recovery-from-Loss* because tokens are not as easy to recover as passwords, but can be recovered by buying another token. These methods are *Resilient-to-Physical-Observation*, *Resilient-to-Leaks-from-Other-Verifiers*, and *Resilient-to-Phishing*, but they are not *Resilient-to-Relay-Attacks*. These methods are *Quasi-Resilient-to-Theft* because the tokens can be secured with a PIN, but they do not offer *Requiring-Explicit-Intent* as the user is authenticated whenever she is in the proximity of the device, without any consent.

8.4 Biometric-based Methods

Biometric authentication methods are *Memorywise-Effortless*, *Nothing-to-Carry*, and generally *Easy-to-Learn* and *Efficient-to-Use*, but they do present certain challenges: they are easy to steal (do not offer *Resilient-to-Theft*) and difficult to revoke (do not offer *Easy-Recovery-from-Loss*). They are also not *Resilient-to-Leaks-from-Other-Verifiers* and *Resilient-to-Phishing* as the leaked or stolen credentials can be used to authenticate as the user. Biometric methods, by design, are not *User-Agnostic*. Fingerprint-based methods are *Physically-Effortless* as with modern fingerprint sensors providing a fingerprint is as easy as pressing a key. They are *Resilient-to-Physical-Observation* because (unlike voice or face) fingerprints are hard to capture by observing or filming the user. They offer *Requiring-Explicit-Intent* as the act of providing a fingerprint conveys user consent.

Voice-based authentication schemes are *Quasi-Physically-Effortless* as speaking is not as easy as pressing a key, but it is natural to most users. We rate voice-based methods as not offering *Requiring-Explicit-Intent* and *Resilient-to-Relay-Attacks* as voice can be synthesized [39], replayed, or relayed. Unlike fingerprint or voice, face-based schemes are *Physically-Effortless* as the user simply has to be in the view of the camera, and because

this is a better indication of intent to use a device than simply being in proximity of the device, we rate them as *Quasi-Require-Intentionality*.

9 CONCLUSION

In this paper we present SAW, a new authentication method that allows users to authenticate by simply tapping a key multiple times or wiggling the mouse – a simple, natural interaction. Through in-lab user studies, we explored SAW's feasibility and users' perception about SAW's usability and security. From our user studies, SAW proved to be *quick*, authenticating participants within 2 s; *effortless*, and several participants liked the natural tap authentication interaction; *usable*, with a low false-negative rate of 2.5%; and *secure*, with a low false-positive rate of 1.8% even in the most advantageous conditions for an adversary. In comparison, passwords are burdensome and error-prone, with error rates (false-negative rate) as high as 7.9% for laptop and desktop authentications. We envision SAW – with its low false-negative rate and seamless authentication – to augment password-based methods, and reduce the number of times people have to enter passwords. Because SAW also supports seamless authentication for multiple users, it is particularly useful in multi-user shared environments where the authentication burden on users is high (e.g., hospitals, enterprise environments).

ACKNOWLEDGMENTS

We thank Andrés Molina-Markham for participating in the early phases of this research. We also thank Aditya Vashistha for providing feedback on the paper and Alex Takakuwa for his inputs on revising the paper. This research results from a research program at the Institute for Security, Technology, and Society at Dartmouth College, supported by the National Science Foundation under award number CNS-1329686. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

REFERENCES

[1] Anne Adams and Martina A. Sasse. 1999. Users Are Not the Enemy. *Commun. ACM* 42, 12 (Dec. 1999), 40–46. https://doi.org/10.1145/322796.322806

[2] Petar S. Aleksic and Aggelos K. Katsaggelos. 2006. Audio-Visual Biometrics. *Proc. IEEE* 94, 11 (Nov. 2006), 2025–2044. https://doi.org/10.1109/JPROC.2006.886017

[3] android 2018. Face Unlock On Android 4.0. http://www.huffingtonpost.com/2011/10/19/face-unlock-ice-cream-sandwich_n_1020207.html

[4] Apple 2018. Apple Watch. Retrieved May 11, 2018 from https://www.apple.com/watch/

[5] Apple Unlock 2018. How to Unlock Your Mac with Your Apple Watch - Apple Support. Retrieved May 11, 2018 from https://support.apple.com/en-us/HT206995

[6] Atama 2018. Atama Sesame 2 Wireless Proximity Lock. Retrieved May 11, 2018 from https://atama.io/sesame2

[7] Davide Balzarotti, Marco Cova, and Giovanni Vigna. 2008. ClearShot: Eavesdropping on Keyboard Input from Video. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 170–183. https://doi.org/10.1109/SP.2008.28

[8] Kathy Baxter, Catherine Courage, and Kelly Caine. 2015. *Understanding Your Users: A Practical Guide to User Research Methods* (second ed.). Morgan Kaufmann.

[9] Nick Bilton. 2015. Keeping Your Car Safe from Thieves. New York Times. Retrieved May 11, 2018 from https://www.nytimes.com/2015/04/16/style/keeping-your-car-safe-from-electronic-thieves.html

[10] bluetooth [n. d.]. Bluetooth technology adding mesh networking to spur new wave of innovation. https://www.bluetooth.com/news/pressreleases/2015/02/24/bluetoothtechnology-adding-mesh-networking-to-spur-new-wave-of-innovation Last accessed May 2018.

[11] J. Bonneau, C. Herley, Paul C. van Oorschot, and F. Stajano. 2012. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 553–567. https://doi.org/10.1109/SP.2012.44

[12] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (Oct. 2001), 5–32. https://doi.org/10.1023/a:1010933404324

[13] John Brooke and Others. 1996. SUS - A Quick and Dirty Usability Scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[14] Cory Cornelius, Ronald Peterson, Joe Skinner, Ryan J. Halter, and David Kotz. 2014. A Wearable System that Knows who Wears it. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. https://doi.org/10.1145/2594368.2594369

[15] Mark D. Corner and Brian Noble. 2002. Zero-Interaction Authentication. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*. 1–11. https://doi.org/10.1145/570645.570647

[16] D. Dolev and A. Yao. 2006. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory* 29, 2 (Sept. 2006), 198–208. https://doi.org/10.1109/TIT.1983.1056650

[17] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2010. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*. https://www.semanticscholar.org/paper/26e6b1675e081a514f4fdc0352d6cb211ba6d9c8

[18] Lishoy Francis, Gerhard P. Hancke, Keith Mayes, and Konstantinos Markantonakis. 2010. Practical NFC Peer-to-Peer Relay Attack Using Mobile Phones. In *Proceedings of the International Workshop on Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*. https://doi.org/10.1007/978-3-642-16822-2_4

[19] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. 2008. Improving Wireless Privacy with an Identifier-Free Link Layer Protocol. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 40–53. https://doi.org/10.1145/1378600.1378607

[20] Health 2018. HealthCast Inc. Retrieved May 11, 2018 from http://www.gohealthcast.com/index.html

[21] Rosa R. Heckle. 2011. Security Dilemma: Healthcare Clinicians at Work. *IEEE Security & Privacy* (2011). https://doi.org/10.1109/MSP.2011.74

[22] Cormac Herley. 2009. So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users. In *Proceedings of the Workshop on New Security Paradigms Workshop (NSPW)*. ACM, 133–144. https://doi.org/10.1145/1719030.1719050

[23] Otto Huhta, Swapnil Udar, Mika Juuti, Prakash Shrestha, Nitesh Saxena, and N. Asokan. 2016. Pitfalls in Designing Zero-Effort Deauthentication: Opportunistic Human Observation Attacks. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2016.23199

[24] Ziv Kfir and Avishai Wool. 2005. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard. In *Proceedings of the International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*. https://doi.org/10.1109/SECURECOMM.2005.32

[25] Ross Koppel, Sean Smith, Jim Blythe, and Vijay Kothari. 2015. Workarounds to Computer Access in Healthcare Organizations: You Want My Password or a Dead Patient? *Driving Quality in Informatics: Fulfilling the Promise – Studies In Health Technology and Informatics* 208 (Feb. 2015), 215–235. http://www.cs.dartmouth.edu/~sws/pubs/ksbk15-draft.pdf

[26] Robert J. Kosinski. 2013. A literature review on reaction time. http://www.cognaction.org/cogs105/readings/clemson.rt.pdf

[27] Narayanan C. Krishnan and Diane J. Cook. 2014. Activity Recognition on Streaming Sensor Data. *Pervasive and Mobile Computing* 10, B (Feb. 2014), 138–154. https://doi.org/10.1016/j.pmcj.2012.07.003

[28] Arun Kumar, Nitesh Saxena, Gene Tsudik, and Ersin Uzun. 2009. A Comparative Study of Secure Device Pairing Methods. *Pervasive and Mobile Computing* 5, 6 (2009), 734–749. https://doi.org/10.1016/j.pmcj.2009.07.008

[29] Rajesh Kumar, Vir V. Phoha, and Anshumali Jain. 2015. Treadmill attack on gait-based authentication systems. In *Proceedings of the International Conference on Biometrics Theory, Applications and Systems (BTAS)*. https://doi.org/10.1109/BTAS.2015.7358801

[30] Albert Levi, Erhan Çetintas, Murat Aydos, Çetin Kaya Koç, and M. Ufuk Çaglayan. 2004. Relay Attacks on Bluetooth Authentication and Solutions. In *Proceedings of the International Symposium on Computer and Information Sciences (ISCIS)*. https://doi.org/10.1007/978-3-540-30182-0_29

[31] Xiaohui Liang, Tianlong Yun, Ronald Peterson, and David Kotz. 2017. LightTouch: Securely Connecting Wearables to Ambient Displays with User Intent. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. 1–9. https://doi.org/10.1109/INFOCOM.2017.8057210

[32] MacID 2018. MacID - Unlock Your Mac with Just Your Fingerprint. https://www.macid.co

[33] Shrirang Mare. 2016. *Seamless Authentication for Ubiquitous Devices*. Ph.D. Dissertation. Dartmouth College Computer Science. http://www.cs.dartmouth.edu/reports/TR2016-793.pdf Available as Dartmouth Computer Science Technical Report TR2016-793.

[34] Shrirang Mare, Mary Baker, and Jeremy Gummeson. 2016. A Study of Authentication in Daily Life. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*. https://www.usenix.org/conference/soups2016/technical-sessions/presentation/mare

[35] Shrirang Mare, Andrés Molina-Markham, Cory Cornelius, Ronald Peterson, and David Kotz. 2014. ZEBRA: Zero-Effort Bilateral Recurring Authentication. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 705–720. https://doi.org/10.1109/SP.2014.51

[36] Suhas Mathur, Rob Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. 2011. ProxiMate: Proximity-based Secure Pairing using Ambient Wireless Signals. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 211–224. https://doi.org/10.1145/1999995.2000016

[37] Rene Mayrhofer and Hans Gellersen. 2009. Shake Well before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing* 8, 6 (June 2009), 792–806. https://doi.org/10.1109/TMC.2009.51

[38] Tey C. Meng, Payas Gupta, and Debin Gao. 2013. I can be you: Questioning the use of Keystroke Dynamics as Biometrics. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*. http://flyer.sis.smu.edu.sg/ndss13-tey.pdf

[39] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. 2015. All Your Voices are Belong to Us: Stealing Voices to Fool Humans and Machines. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*. https://doi.org/10.1007/978-3-319-24177-7_30

[40] Saul B. Needleman and Christian D. Wunsch. 1970. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology* 48, 3 (July 1970), 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

[41] NFC 2018. NFC Ring - Safe, Simple, Secure. http://nfcring.com

[42] Nymi 2018. Nymi. Retrieved May 11, 2018 from https://nymi.com

[43] Sean Peisert, Ed Talbot, and Tom Kroeger. 2013. Principles of Authentication. In *Proceedings of the Workshop on New Security Paradigms Workshop (NSPW)*. ACM, 47–56. https://doi.org/10.1145/2535813.2535819

[44] Aanjhan Ranganathan and Srdjan Capkun. 2017. Are We Really Close? Verifying Proximity in Wireless Systems. *IEEE Security & Privacy* (2017). https://doi.org/10.1109/MSP.2017.56

[45] Kasper B. Rasmussen and Srdjan Čapkun. 2010. Realization of RF Distance Bounding. In *Proceedings of the USENIX Security Symposium (USENIX Security)*. 25. http://www.usenix.org/events/sec10/tech/full_papers/Rasmussen.pdf

[46] R. Rawassizadeh, T. J. Pierson, R. Peterson, and D. Kotz. 2018. NoCloud: Exploring Network Disconnection through On-Device Data Analysis. *IEEE Pervasive Computing* 17, 1 (Jan 2018), 64–74. https://doi.org/10.1109/MPRV.2018.011591063

[47] reaction 2018. Human Reaction Time Test. Retrieved May 11, 2018 from http://www.humanbenchmark.com/tests/reactiontime/

[48] Arun Ross, Jidnya Shah, and Anil K. Jain. 2007. From Template to Image: Reconstructing Fingerprints from Minutiae Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 4 (April 2007), 544–560. https://doi.org/10.1109/TPAMI.2007.1018

[49] Shimmer 2018. Shimmer Research. Retrieved May 11, 2018 from http://www.shimmersensing.com

[50] Sara Sinclair. 2013. *Access Control in and for the Real World*. Ph.D. Dissertation. Dartmouth College Computer Science. https://www.cs.dartmouth.edu/~trdata/reports/abstracts/TR2013-745/

[51] Ole Tange. 2011. GNU Parallel: The Command-Line Power Tool. *;login: The USENIX Magazine* 36, 1 (Feb 2011), 42–47. https://doi.org/10.5281/zenodo.16303

[52] Furkan Tari, Ant A. Ozok, and Stephen H. Holden. 2006. A Comparison of Perceived and Real Shoulder-surfing Risks Between Alphanumeric and Graphical Passwords. In *Proceedings of the Symposium On Usable Privacy and Security (SOUPS)*. ACM, 56–66. https://doi.org/10.1145/1143120.1143128

[53] Shari Trewin, Cal Swart, Larry Koved, Jacquelyn Martino, Kapil Singh, and Shay Ben-David. 2012. Biometric Authentication on a Mobile Device: A Study of User Effort, Error and Task Disruption. In *Proceedings of the Annual Computer Security Applications Conference*. ACM, 159–168. https://doi.org/10.1145/2420950.2420976

[54] He Wang, Ted T. Lai, and Romit R. Choudhury. 2015. MoLe: Motion Leaks through Smartwatch Sensors. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*. http://web.engr.illinois.edu/~hewang5/papers/mole-final.pdf

[55] A. T. Welford. 1988. Reaction Time, Speed of Performance, and Age. *Central Determinants of Age-Related Declines in Motor Function* 515, 1 (Jan. 1988), 1–17. https://doi.org/10.1111/j.1749-6632.1988.tb32958.x

Received November 2017; revised May 2018; accepted September 2018