Continuous Smartphone Authentication using Wristbands

Shrirang Mare
University of Washington
shri@cs.washington.edu

Reza Rawassizadeh
University of Rochester
rrawasi@gmail.com

Ronald Peterson and David Kotz
Dartmouth College
{rapjr,kotz}@cs.dartmouth.edu

***Abstract*—Many users find current smartphone authentication methods (PINs, swipe patterns) to be burdensome, leading them to weaken or disable the authentication. Although some phones support methods to ease the burden (such as fingerprint readers), these methods require active participation by the user and do not verify the user's identity after the phone is unlocked. We propose CSAW, a *continuous smartphone authentication* method that leverages wristbands to verify that the phone is in the hands of its owner. In CSAW, users wear a wristband (a smartwatch or a fitness band) with built-in motion sensors, and by comparing the wristband's motion with the phone's motion, CSAW continuously produces a score indicating its confidence that the person holding (and using) the phone is the person wearing the wristband. This score provides the foundation for a wide range of authentication decisions (e.g., unlocking phone, deauthentication, or limiting phone access). Through two user studies ($N = 27, 11$) we evaluated CSAW's accuracy, usability, and security. Our experimental evaluation demonstrates that CSAW was able to conduct initial authentication with over 99% accuracy and continuous authentication with over 96.5% accuracy.**

I. Introduction

Mobile devices like smartphones and tablets provide access to a wide range of sensitive services and personal information (email, photos, social networks, bank transactions, health records, enterprise data, and more). An unlocked phone is vulnerable to snoopy family, friends, co-workers, and passers-by [1]. Smartphone authentication (i.e., unlocking) and deauthentication (i.e., locking) are, unfortunately, still manual processes that users have to repeat several times each day [2], [3]. As a result, many users trade-off security for usability by either choosing no phone locking mechanism [4] or choosing simple, easy-to-remember, easy-to-type, and easy-to-break passcodes or swipe patterns [5]. Furthermore, most current phone authentication methods provide one-time authentication and do not support continuous authentication; loaning one's phone to a colleague or family member (e.g., to make a call or play a game) typically gives them full access to all the content in the phone and services reachable through the phone.

In this paper, we propose CSAW (Continuous Smartphone Authentication using Wristbands), a system that allows a phone to passively and continuously verify that the phone is literally in the hands of its owner. CSAW (pronounced 'seesaw') is a fundamentally new service to applications and subsystems on the phone, which can serve as a foundation for *initial authentication* (e.g., the phone unlocks when picked up by the owner), *deauthentication* (e.g., the phone locks when accessed by someone other than the owner), *limitation* (e.g., the phone allows guest access to specific apps), and *delegation* (e.g., the owner can temporarily grant specific access to another person). Figure 1 illustrates how initial authentication works in CSAW. After initial authentication, CSAW continues to verify the user using data from the user's watch and the phone.

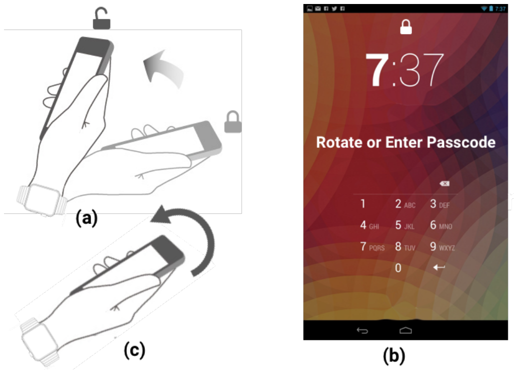

Fig. 1. CSAW phone unlock mechanism: (a) picking up phone with the watch hand unlocks the phone; (b) custom unlock screen, shown if pickup unlock fails, gives the user an option to unlock the phone by entering passcode or by performing a simple rotate gesture with the watch hand (as shown in c).

CSAW works by correlating the owner's wrist motion with phone motion and phone input, and continuously producing a score indicating its confidence that the person holding (and using) the phone is indeed the owner. The owner wears a smartwatch or a fitness band ('watch', for short) with built-in motion sensors (accelerometer and gyroscope) and a wireless radio (e.g., Bluetooth Low Energy) allowing it to share wrist-motion data with the phone. Like any smartwatch or fitness band, the watch is pre-paired with the phone and, in CSAW, serves as an *identity token* for authenticating to the phone. (We assume that the watch is indeed on the wrist of the phone owner; the owner can authenticate to the watch using a PIN [6] or using a biometric approach [7].)

Contribution. We present CSAW, a method for continuously and unobtrusively quantifying confidence that a phone's user is the phone's owner, leveraging the owner's watch as an identity token and correlating the owner's wrist motion with the phone's motion and touchscreen inputs. Our evaluation shows CSAW's accuracy was over 99% for initial authentication and over

Workshop on Usable Security (USEC) 2019
24 February 2019, San Diego, CA, USA
ISBN 1-891562-57-6
https://dx.doi.org/10.14722/usec.2019.23013
www.ndss-symposium.org

96.5% for continuous authentication, with high security (low false-positive rate) and high usability (low false-negative rate).

Although this paper focuses on smartphones, the techniques can also be applied to tablets or other mobile devices; we use the term *phone* to refer to smartphones unless otherwise noted. Similarly, we use the term *watch* to refer to smartwatches, although the CSAW capabilities could easily be met by typical fitness bands as well. Both types of wristband devices are becoming an everyday accessory, and CSAW adds another benefit to wearing one such device.

II. Related work

We situate our work in the large body of ongoing research on reducing the burden of smartphone authentication. Below we relate our work to efforts that leverage motion sensors in wearables for authentication to non-smartphone platforms, efforts on phone authentication without any external device, and efforts on phone authentication that leverage wearable devices.

Authentication using motion sensors in wrist wearables. Lester et al. were one of the first to propose using accelerometers to identify whether two devices were carried *together* by the same person [8]. Building on this idea, Mayrhofer et al. showed how to securely pair two unknown accelerometer devices by shaking them together [9]; Cornelius et al. showed how to verify whether two devices are on the same body but at different locations, by leveraging the common motion during walking [10]; and recently, Findling et al. showed how to transfer authentication state between two devices by shaking them together [11]. In these systems, a user has to provide some deliberate motion (e.g., walking or shaking devices together) for authentication, but in CSAW we explore how to leverage motion during natural phone use for authentication.

In our earlier work, we leveraged motion sensors in a wristband to authenticate PC users by correlating a user's wrist movement with the keyboard and mouse inputs [12], [13]. Acar et al. use motion sensors in a user's wristband to capture keystroke related information and do keystroke-based behavioral authentication for a PC [14]. Although CSAW uses a similar approach, CSAW focuses on smartphones, which presents new challenges (e.g., desktops are stationary and wrist motion is more predictable, whereas phones are mobile). These devices have different interaction modes, and the types of correlations that we perform in CSAW require different signal processing and methods.

Smartphone authentication. The common non-biometric phone authentication methods are 4-digit PIN and swipe pattern [3], even though they can be easily stolen and spoofed [15]–[19]. There have been several proposals for alternate authentication methods, some better suited for certain users than others; for example, instrumenting the back surface of a phone for easy authentication [20], providing a short-lived access to bypass authentication for short sessions [21], graphical passwords [22], or combining swipe and PIN [23]. In the same vein of reducing phone authentication burden, we propose an alternate authentication method that works best for users who wear (or could wear) a smartwatch.

Common biometric initial authentication methods are based on fingerprint and facial recognition [24]. Compared to these methods, CSAW offers different trade-offs: these biometrics do not require users to carry any additional device, but they do require special hardware on the phone, involve an inherent privacy trade-off and do not do continuous authentication, whereas although CSAW requires users to wear a wrist wearable, it does not need any special hardware on the phone, is privacy-preserving and supports continuous authentication.

Other efforts for initial authentication involve gesture-based approaches that leverage motion sensors in the smartphone to capture a secret gesture. For example, Das et al. proposed a group authentication method based a secret knock performed on the phone, which is recognized using an accelerometer and a microphone in the phone [25]; Yang et al. proposed free-form gesture-based authentication, where gestures are recognized using motion sensors in the phone [26]; and more recently, Lee et al. proposed a behavioral biometric based on users' unique pickup action [27]. Although gesture-based authentication methods do not require any special hardware on the phone, they are highly susceptible to mimic attacks [28].

Much of prior work on continuous authentication for smartphones has focused on biometric approaches, exploring the use of touch gestures [29]–[32], in-air gestures [33], touch keystrokes [34], and app usage patterns [35]. Behavioral biometrics work well when users stick to their usual behavior, but are error-prone when there is variability in mobile usage or user's mobility; some variability and mobility can be accommodated, as Crawford and Ahmadzadeh [36] did with walking and stationary activity for keystroke biometric, but the approach becomes intractable given the wide range of activities and conditions. On the other hand, if users have a fix patterned behavior, it is easy to copy and mimic [37], [38]. Our method is user-agnostic – not a biometric – so it is less susceptible to variations in mobile usage and mobility; in fact, user mobility provides us with additional data to correlate.

Riva et al. proposed "progressive authentication" (PA), which combines multiple signals (biometric, continuity, and possession) to determine a confidence level for a user's authenticity [39]. This metric is combined with a user-configured protection level to determine when authentication should occur, thus reducing the need for authentication. CSAW and PA are actually complimentary – CSAW could be integrated in PA as one of the modules that determine user's authenticity.

Smartphone authentication with wearables. Much of the smartphone authentication effort has been focused on methods that do not involve an additional device, because carrying an additional device is a usability burden. However, as wearable devices are becoming common, they provide another modality to leverage for authentication.

Vu et al. [40] and Nguyen et al. [41] proposed a user identification method where users wear a special ring that transmits their identity to the phone's touchscreen, but in the clear; so this method is suitable for personalization, but not for secure authentication. Azimpourkivi et al. proposed using a camera-based method where for authentication users take a photo of one of their access tokens [42]. Frank proposed a similar method, but the camera is in the wearable access token [43]. These methods require the user to take a photo every time the user wants to authenticate, which is too burdensome for the frequent authentications that are common for smartphones.

The closest work to CSAW is the TwistIn method proposed by Leung et al. [44]. In this method, users wear a watch and authenticate to their phone by twisting the forearm holding the phone. The twist action is captured by the watch and the phone, and then compared by the phone, using the same-action principle used in the Shake-well-before-use system [9]. In CSAW, we use a similar gesture as an optional action to boost the user-verification confidence score, but in CSAW we first aim to perform seamless initial authentication using the natural phone pick-up action. Moreover, unlike TwistIn, CSAW can continuously authenticate the user by leveraging natural wrist movements during phone use.

III. Inspiration and Applications

CSAW is inspired by the fact that smartphones and tablets are *hand-held* mobile devices. We interact with our phones using our hands: we provide touchscreen input (taps and swipes), we pick them up, we carry them around. Thus, the phone's motion and touchscreen inputs should correlate with the owner's hand movements (measured by the motion sensors on the watch), and can be leveraged to verify that the owner is in fact the person picking up the phone and providing input to the phone. Indeed, CSAW can passively continue to monitor the correlation and deauthenticate the user (lock the phone) when the wrist motion no longer matches phone input or motion.

CSAW's underlying goal is to provide a continuous quantitative estimate of the confidence that a phone's current user is in fact the phone's owner, assuming that the owner is wearing the owner's watch. Before we dig deeper into the specific CSAW method in Section IV, let us look at how a mobile device could use the CSAW confidence metric to support initial authentication, deauthentication, limitation, and delegation.

Initial authentication (or unlocking). CSAW's interaction correlations can be used to implement fast initial authentication, leveraging natural interactions that happen before the user uses the phone; for instance, getting the phone out of her pocket or picking up the phone from a desk. Modern smartphones already leverage the natural pickup action to provide quick access to notifications by automatically turning on the phone display when the phone is picked up [45], [46]. Using the same pickup action CSAW can authenticate a user, so she need only press the home button (or swipe) to access the phone. CSAW could also be combined with a fingerprint or a face reader, or a mechanism such as Smart Lock from Android, to make a strong multi-factor authentication system.

Deauthentication via continuous authentication. Common authentication schemes (such as those based on PINs, swipe patterns, or fingerprints) do not perform automatic deauthentication, i.e., they authenticate the user once but then rely on the user to deauthenticate (lock) the device. The current common solution for deauthentication is *timeouts*, i.e., to lock the device after a period of inactivity. However, choosing a timeout duration that works across all context is difficult, since users' phone session times vary based on the time of the day and context [3]. A short timeout means a user may have to unlock their phone more often, whereas a long timeout may make the phone vulnerable to snoopy family members, friends, or co-workers [1], [4]. As a result, most smartphone users manually lock their phones before putting it away [21]. With automatic deauthentication based on CSAW there is no need for timeouts or to manually lock the phone, but if desired, individuals can comfortably set a long timeout, knowing that if anyone else attempts to use their phone before the timeout, CSAW will recognize the other user and then lock the phone.

Limitation and access management. Consider a smartphone owner who lends her phone to a child to play games, or who lends her phone to a friend to search the Internet for a recipe. She may worry that app notifications could display personal information or that the borrower may launch an app with sensitive information. CSAW could support a notification engine that presents app notifications only when the owner is using the phone, or support an OS home-screen app launcher to limit which applications can be used when a guest (non-owner) is using the phone.

Delegation to other CSAW users. Consider again a smartphone owner who wants to lend her phone to a spouse or trusted co-worker to allow them to access specific applications (e.g., view specific photo albums) without allowing them full access to the phone. A service based on CSAW could provide an interface to delegate access to a trusted user, essentially, introducing that user's watch (and that user's identity) to the phone, so that the delegated user may use the owner's phone in the future for certain approved purposes. Although this mechanism requires thoughtful attention to the user interface (and such a design is outside the scope of this paper) the CSAW system could be easily extended to support such a use case.

IV. CSAW method

CSAW monitors wrist motion, phone motion, and phone inputs so it can determine whether they are correlated and produce a summary metric we call the *confidence score*, a value between 0 and 1 that indicates CSAW's confidence that the user of the phone is indeed the watch-wearing owner. Specifically, it produces fresh scores every 1-2 seconds: an instantaneous estimate $C(t)$ based on the latest data at time t, and an exponentially-weighted moving average (EWMA) $\overline{C}(t)$ that smoothes recent scores:

$$\overline{C}(t) = (1-\alpha)C(t) + \alpha\overline{C}(t-1) \qquad (1)$$

where the factor α weights the contribution of the new confidence score. As described above, application and system policies can use these metrics to drive authentication-related decisions. In this section, we describe the system architecture of CSAW and the calculations that lead it to produce this score.

Figure 2 depicts CSAW's architecture and its modules. CSAW receives a steady stream of motion data from the phone and the watch, and touchscreen input data from the phone. These data flows are segmented into windows and examined by three modules; the grip detector determines how the user holds the phone, Motion-to-Motion correlator (M2MC) correlates phone and watch motion, and Motion-to-Input correlator (M2IC) correlates watch motion and phone input. The resulting correlation metrics are considered by the scoring engine that actually computes $C(t)$ (further discussed in Section IV-D).

Although CSAW outputs a confidence value frequently, not all of its modules need be active continuously. When the watch is not present, perhaps because the owner has stepped out of the phone's range, there is no watch data (W_m); the correlation

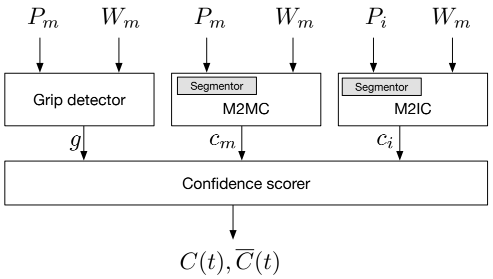

Fig. 2. CSAW architecture. CSAW uses the phone's motion data (P_m), the watch's motion data (W_m), and the phone's touchscreen input data (P_i) to verify a user. The grip detector module detects the phone grip (g) using P_m and W_m. The M2MC module computes a correlation score (c_m) based on P_m and W_m, and the M2IC computes a correlation score (c_i) based on P_i and W_m. Using the determined grip (g) and correlations scores c_m and c_i, the confidence scorer computes an instantaneous confidence score $C(t)$ based on the most recent data, and a moving average of the confidence score ($\overline{C}(t)$).

modules simply output 0 (not correlated) and that drives the confidence score $C(t)$ to 0 (lowest confidence). When the watch is present but the phone is experiencing no input or motion, perhaps sitting on a table or in a bag near the owner, all the modules can 'rest' (to save energy) and the confidence score is again $C(t) = 0$. Indeed, CSAW calculates the correlation between the watch motion and the phone motion/inputs *only* when there is a user-phone interaction, that is, any action by the user that provides input to the phone or changes the phone's position (which can be sensed by the accelerometer and gyroscope sensors). Examples of user-phone interaction include the user picking up her phone or sending a text message on her phone. As soon as phone input or motion is detected, the watch is instructed to start sending motion data to the phone, and the correlation modules become active.

Although the figure shows two correlation modules, the CSAW architecture is extensible; other sensor or contextual information could be correlated or used as input to the scoring module. Such extensions are opportunities for future research.

A. Grip detector

Phones can be used with one hand or with both hands. We describe a grip using two characters XY, where X is the hand holding the phone and Y is the hand providing input. The owner can hold (or provide input) with the watch-hand (W), the non-watch-hand (N), or with both hands (B); thus, X and Y can each be W, N, or B. If the phone is not held in either hand, we denote it with U (unknown). Thus, there are seven possible grips: BB, NN, NW, WN, WW, UN, UW; Figure 3 shows three types of grips, BB, NW, and WN.

Knowing the grip during a user-phone interaction helps CSAW use the appropriate correlation method to improve the accuracy of the confidence score. For example, motion-to-motion correlation is stronger when the watch and the phone are tightly coupled (WW, WN, BB grips), but error-prone when the watch-hand is used to provide touch inputs (NW, UW). Using the orientation of the watch and the phone, this module produces an output indicating the grip only when the phone is in use. The output at time t is a two-character string $g(t)$ representing the grip. Among the seven grips, CSAW can detect and support authentication for five grips (all grips except NN and UN, where the watch is not in physical contact with the phone).

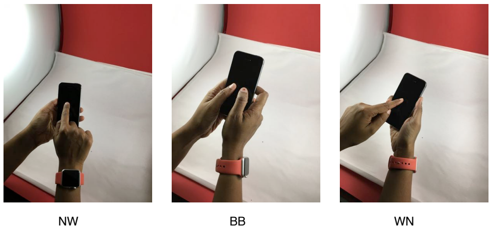

Fig. 3. Watch orientation in three grips (NW: non-watch-hand, watch-hand; BB:both hands; WN: watch-hand, non-watch-hand)

B. Motion-to-Motion Correlator (M2MC)

The motion-to-motion correlator module compares the watch motion with the phone motion to determine whether the phone is held by the user wearing the watch. When the phone is in motion or in use, this module receives two continuous streams of motion data, from the watch (W_m) and the phone (P_m), each a series of sensor data samples of the form $(t, ax, ay, az, gx, gy, gz)$, where t is the time when the sample was collected, and ax, ay, az and gx, gy, gz are the values from accelerometer and gyroscope sensors along their x, y, z axes respectively. The input data streams are segmented using sliding windows of size w_m with overlap fraction o_m; CSAW uses $o_m = 0.5$ and $w_m = 1$ s when the phone is locked and $w_m = 2$ s when it is unlocked. The module outputs a correlation score $c_m(t)$ indicating how well the two motions correlate ($0 \le c_m(t) \le 1$) at time t.

Features. For each segmented window, M2MC computes a correlation feature vector F_m with 256 features from time and frequency domains, as we explain next. We begin with *mean, standard deviation, mean value crossing rate, variation, interquartile range, median, mean absolute deviation, skew, kurtosis, power, energy*, and *peak-to-peak amplitude*, each a statistical representation of a signal. Then we compare the two signals by computing two numbers for each of those 12 statistics – absolute difference and relative difference (ratio) – resulting in 24 features. These features, however, compare aggregate statistics of the two signals without examining whether they vary the same way with time. To those 24 *non-temporal* features we add eight *temporal* features, as follows. Two (*cross-correlation* and *correlation-coefficient*) measure similarity between two signals and how they vary together in the time domain, and a third (*coherence*) measures similarity and variance in the frequency domain. Five more features select the two highest peaks in both signals and compare their corresponding peak timestamps, amplitudes, and inter-peak times. Thus, for any two signals we compute 32 features. Since there are four signals (x, y, z and magnitude) each from the accelerometer and gyroscope, the final feature vector F_m has $256 = 4 \times 2 \times 32$ features.

To determine the correlation score, M2MC uses a model that estimates the probability that a given feature vector F_m represents two motions that correlate. We use a random-forest binary classifier trained to classify a feature vector as 0 (not correlated) or 1 (correlated). The classifier is trained earlier (using data from a population of volunteer subjects) to generate a universal model. For a given feature vector, the classifier computes probability estimates for the two labels (0 and 1); M2MC outputs the classifier's probability estimate for label 1 as its correlation score $c_m(t)$.

C. Motion-to-Input Correlator (M2IC)

The motion-to-input correlator module compares watch motion with phone touch inputs to determine whether the touch inputs are given by the owner using her watch-hand. It outputs a correlation score $c_i(t)$ indicating how well the wrist motion correlates with the phone input at time t, where $0 \leq c_i(t) \leq 1$. If there are no touch inputs, this module outputs a zero.

This module receives a stream of wrist motion data (W_m, as above) and a stream of phone touch events (P_i) of the form $p = (t, id, x, y)$, where t is the time when the sample was collected, id is the unique ID assigned by the phone OS to identify a series of touch events performed by a single tracker/finger, and x and y are the coordinates where the user touched the screen. (We used a Nexus 6 Android phone in our experiments and these values are what the Android OS reports; another phone OS may report touch events slightly differently.) When the phone display is on, the Android OS is constantly sampling the touchscreen to capture touch events. A *tap* is a series of touch events with the same tracking ID (id) but different timestamp (t) and position (x, y). Similarly, a *swipe* is also a series of touch events with the same id and with different t, x, and y, but a swipe is a longer touch interaction than tap, so it contains more samples than a tap.

We segment the input data streams (W_m and P_i) based on touch interactions; that is, each window of data represents exactly one touch interaction (tap or swipe). For a touch interaction ending at time t we extract the corresponding motion data from the watch (using the start and end time of the touch interaction). We then compute a feature vector from the watch data, using the same 12 standard statistics (*mean, standard deviation, mean value crossing rate, variation, interquartile range, median, mean absolute deviation, skew, kurtosis, power, energy*, and *peak-to-peak amplitude*) of the accelerometer and gyroscope magnitudes. The result is a 24-feature vector F_i.

We then use a two-tier classification approach to correlate wrist motion with the touch input. In the first tier, we determine whether F_i is a touch interaction (tap or swipe) using a Naive Bayes binary classifier trained earlier using wrist-motion data from tap and swipe touch interactions as positive labels, and wrist-motion data from other activities (such as walking, wrist stationary, and typing on computer) as negative labels. In the second tier, we use an interaction-specific model to identify the likelihood that F_i (representing the watch motion) is indeed the touch interaction (phone input) performed by the user. (Note that these classifiers are not subject-specific, i.e., they are not trained for any particular user, which avoids the need for new users to train the system before use and makes the method resilient to changes in behavior over time or due to changes in context.) M2IC outputs the likelihood score from the second tier as its correlation score $c_i(t)$.

D. Confidence Scorer

This module periodically outputs the confidence score $C(t)$ at time t. To generate the confidence score, the module uses $c_m(t)$ from the M2MC module, $c_i(t)$ from the M2IC module, and $g(t)$ from the grip module. CSAW intentionally favors M2MC, because compared to M2IC, M2MC's output is more frequent and reliable as it computes correlation over more data and does not depend on the user's touch events. Indeed, CSAW uses M2IC only when output of M2MC is not reliable, i.e., when the watch and phone are not tightly coupled (the phone is not held with the watch-hand). In short,

$$C(t) = \begin{cases} c_i(t) & \text{if } g(t) \in \{\text{NW}, \text{UW}\} \\ c_m(t) & \text{otherwise} \end{cases}$$

The instantaneous confidence score can be too sensitive to act on for most applications as it could result in false-negatives – taking the watch hand away from the phone for a few seconds may cause $C(t)$ to drop. So this module also outputs a moving average $\overline{C}(t)$ as in Eq. 1, to smooth out the momentary fluctuations in the confidence score.

E. Confidence booster

CSAW can help with another important category: second-chance actions. In situations when the confidence value derived from natural user-phone interactions is low, the system could ask the owner to perform some explicit actions to improve confidence. For example, in a system using CSAW for initial authentication, the phone might fail to unlock due to low confidence in the M2MC correlation during a pickup maneuver. The phone could display a notice asking the user to place the phone in their watch hand and then quickly rotate their hand (and phone) as shown in Figure 1; CSAW can easily correlate this motion (as shown in the next section).

V. Study 1: Feasibility and Security Evaluation

The objective of this study was to capture natural user-phone interactions such as picking up phone, checking emails, browsing the Internet, reading news, and typing, to test CSAW's feasibility and security. In particular, we wanted to know: *(i)* whether CSAW's approach for initial and continuous authentication is feasible, i.e., what is CSAW's accuracy in correctly identifying a user; *(ii)* what is CSAW's accuracy in detecting an adversary and how long does CSAW take to detect an adversary; and *(iii)* whether our choice of correlation features improve CSAW's accuracy.

A. Participants

Through flyers posted across our campus, we recruited 27 participants (21 male and 6 female, 13 undergraduate and 14 graduate students) for this study. The study was approved our by organization's Institutional Review Board (IRB); participants received USD 10 to participate in the study.

B. Procedure

In this study we performed two experiments: one to collect data to evaluate CSAW's feasibility in verifying a user, and a second experiment to evaluate CSAW's security against a mimicking attack.

Experiment 1. In our experience, asking participants to simply 'use their phone for 20 minutes' yielded limited user-phone interaction, as they were likely to use a limited number of apps to pass the time. Instead, we gave participants well-defined

TABLE I. FPR AND BAC ACROSS ALL SUBJECTS FOR PICKUP ACTION AND ROTATE ACTION; MEAN (STANDARD DEVIATION).

Activity	Pickup action		Rotate action	
	FPR (%)	BAC (%)	FPR (%)	BAC (%)
PC use	0.0 (0.0)	99.9 (0.3)	0.0 (0.0)	99.6 (0.5)
Phone use	0.0 (0.0)	99.9 (0.3)	0.0 (0.0)	99.6 (0.5)
Same (Pickup/Rotate)	1.7 (1.2)	99.0 (0.7)	0.9 (0.3)	99.1 (0.5)
Walking	0.0 (0.0)	99.9 (0.3)	0.0 (0.0)	99.6 (0.5)
Stationary	0.0 (0.0)	99.9 (0.3)	0.2 (0.3)	99.5 (0.5)

tasks such as ‘read and take action (reply/delete) on 10 emails’, ‘read news for 3 minutes’, ‘Google answers for 5 questions’ or ‘type the given sentences’. We asked participants to wear a Shimmer [47] (a wrist-wearable research device) on their dominant hand, and we provided them with an Android phone for the the experiment. To facilitate tasks for participants on the provided phone, we created dummy user accounts for apps that participants would use (Gmail, Twitter, and Flipboard), and initialized those apps with data. We initialized Twitter and Flipboard accounts by subscribing to some news sources; we initialized Gmail by sending emails (sampled from the Enron email dataset [48]) to the dummy account.

Participants took about 30-40 minutes to finish the user study, performing the following specific tasks: (1) Pick up the phone from the desk with grip WW, unlock it, and put it back on the desk; (2) Do five search queries with grip BB; (3) Read and act (reply/delete) on emails with grip WW; (4) Skim articles on Flipboard for 3 minutes with grip NW; (5) Type given sentences on the phone with grips WW, NW, and BB; and (6) Pick up the phone and do a ‘rotate’ action (rotate it clockwise, and then counter-clockwise to the starting position) with grip WW.

Each participant did about 15 minute of continuous phone use (search, email, news, typing), and provided 5 pickup actions and 5 rotate actions. During the experiment, we captured motion-sensor data (accelerometer and gyroscope) at 200 Hz from the Shimmer wristband and the phone, and also the phone touch events (using the Android `adb` utility).

Experiment 2. We wanted to evaluate how well CSAW performs against a mimicking attack and how quickly CSAW detects a different user. Participants acted as adversaries mimicking the victim (researcher). They were asked to mimic the ‘pickup’ action (to test initial authentication) and use the phone for 30 seconds (to test continuous authentication). For the pickup task, the victim (researcher) picked up a test phone and the adversary (participant) picked up the target phone while mimicking and synchronizing his pickup action with the victim’s pickup action. For the phone-use task, the user (researcher) handed over the phone to the guest user (participant), and the guest continued using the phone while the user typed on a nearby computer. In order for CSAW to support limitation or delegation, it is important to identify change in user quickly, and this second task was designed to assess just that. We repeated both the tasks five times with each participant. During this experiment, we measured the researcher’s wrist movement (using Shimmer) and target phone movement.

C. Results

We evaluate CSAW’s accuracy (Section V-C1), its security with respect to its goals and threats (Section V-C2), and its feasibility in an out-of-lab setting (Section VI).

1) Accuracy: Since all of our aforementioned scenarios use the confidence score as input into an authentication policy decision, one way to evaluate the quality of the confidence score is to evaluate the quality of some authentication decisions. In such decisions, a ‘positive’ decision means that the policy decided that the user of the phone is the expected user and access should be permitted; a ‘negative’ decision means the opposite. A ‘false positive’ is a security failure, because it means that an imposter is incorrectly judged to be the owner and access is allowed. A ‘false negative’ is a usability failure, because it means that the owner is incorrectly judged to be an imposter and access is denied. We thus focus on the FPR, which is the fraction of decisions that were false positives, and the FNR, which is the fraction of decisions that were false negatives, as our primary metrics; we want these close to zero.

From the study data, we generated positive samples (with the watch data and the phone data from the same subject) and negative samples (with the watch data and the phone data from different subjects). Generating samples this way allows us to evaluate FNR as well as FPR, but it results in an imbalanced test dataset with a larger number of negative samples than positive samples. So we report the accuracy using the balanced accuracy (BAC) metric, which takes into account imbalance in test samples [49]; we want this number to be close to one.

Initial authentication is an important use case so we begin there. For our evaluation, we consider the simplest possible authentication policy, which makes a new decision each time CSAW provides a new confidence score C. This policy is a simple threshold comparison and classifies using the immediate confidence score: positive if $C > 0.5$ and negative if $C \leq 0.5$.

Initial authentication. With CSAW one should be able to pick up a phone and achieve initial authentication, so we used the data from the ‘pickup’ task in the user study. In this scenario, the phone is locked and sitting on a table; when a person picks up the phone, the phone should unlock for the owner (no false negatives) and not the imposter (no false positives). When the phone is locked there is no input, so CSAW ignores the grip and M2IC, and produces a confidence score from M2MC alone. Our experiments show that balanced accuracy for M2MC was more than 99%. The average FNR was 0.8% (± 1.0), indicating high usability. The FPR was at or near zero, indicating high security, but let’s look closer. Consider situations where an imposter picks up the phone while the user is working on a PC, using another phone, picking up another object, walking, or stationary; Table I shows the average FPR and BAC for each of these cases, presenting the mean and standard deviation over a 10-fold cross validation on our dataset. In nearly every case, there were virtually no false positives and near-perfect accuracy. The hardest case occurred when the imposter and the user both performed the same type of action (pickup); nonetheless, CSAW performed well with a low FPR of 1.7% (± 1.2). For applications where even this FPR is too high, the policy could increase its confidence threshold to reduce FPR (with a possible increase in FNR), or consider a series of multiple confidence values, but a full exploration is left for future work.

Confidence booster. Although the false negative rate for initial authentication was below 1%, every such failure is an annoyance to the owner. As discussed in Section IV-E, CSAW can improve usability in such cases by asking the user to perform a simple explicit action to boost its confidence in the owner's presence as the holder of the phone. We thus evaluated CSAW's performance for a rotate action, in which the user holds the phone in her watch-hand and rotates her forearm so that the watch and the phone both experience the same rotate motion (Figure 1). We used the data from the 'rotate' action in our user study, and formed the test dataset for different cases as discussed above. Because the phone is expected to be in the watch-hand (grip WN), this confidence booster uses only M2MC. The average FNR from a 10-fold cross-validation was 0.2% (± 0.6) and Table I shows the average FPR and BAC for different cases. The average FPR for the hard case ('Same' action, i.e., the user and the imposter both performed the rotate action) was low: 0.9% (± 0.3). Note that this rotate action is only required *if* the seamless pickup authentication (which has 1% failure rate) fails. So the error rate when both pickup and rotate actions fail to verify a user is negligible (less than 0.01%). Performing a rotate action takes less than 1 second, so this can be a quicker way for the phone owner to verify herself than entering a passcode or giving a fingerprint; if the user is already holding the phone with her watch-hand, she does not even have to change the phone grip.

Continuous authentication. Finally, we evaluated CSAW's performance for continuous authentication; in this case, the full CSAW infrastructure is relevant (M2MC, M2IC, and the grip detector), because the user may hold the phone in either (or both) hands, provide input with either (or both) hands, and switch grips during use. We drew samples from the 'phone use' tasks that subjects performed in our user study. In this case, the imposter has received (or taken) the unlocked phone from the phone owner, and attempts to continue using the phone while the owner performs some other activity nearby. (The 'imposter' may be a friend or family member, and not necessarily a malicious stranger; recall that continuous authentication is the foundation for limitation and delegation as well as deauthentication.) Table II lists these other activities: the owner was stationary, walking, using a PC, using another phone, or doing the same task as the imposter, while being in the radio proximity of the phone. The results show a relatively high 2-7% FPR. Keep in mind, though, that these decisions are made once every two seconds, so the chances are slim that an imposter can maintain a consistent series of positive outcomes for more than a few seconds. The average false negative rate for continuous authentication was 2.4% (± 0.7), which appears large enough to be unusable, but this is also produced every two seconds. Furthermore, this result applies to the simple-minded policy defined earlier in this section; we anticipate that a practical policy would use the EWMA $\overline{C}(t)$ or other smoothed version of the confidence score as the basis for its decisions. For instance, Figure 4 shows the instantaneous confidence score $C(t)$ for subject S1 and EWMA $\overline{C}(t)$ with $\alpha = 0.06$.

Grip detection. Knowing the grip allows CSAW to choose the appropriate correlator module for correlation and give a reliable confidence score; if the phone is not in the watch-hand, then the correlation score from M2MC is meaningless. We evaluated detection accuracies for the three grips shown in Figure 3: BB, WN, and NW. The orientation of the phone's touchscreen and the watch's face can be determined by the z-axis component of their acceleration. To measure the watch's relative face orientation with respect to the phone's touchscreen, we compute the difference of their z-axis acceleration, and use the difference to distinguish the grips. For CSAW it is critical to distinguish between grips NW and WN, as it uses these grips to choose the correlation module. Using a threshold of –5, we can distinguish between NW and WN grips in our dataset with 99.12% accuracy. Although this threshold proved accurate to distinguish the grips used by our participants while performing different tasks and using different apps, a larger user study will verify how well this threshold serves as a generic threshold. (Another approach is to build a classifier to distinguish cases.)

TABLE II. AVERAGE (STANDARD DEVIATION) ACROSS ALL SUBJECTS FOR CONTINUOUS AUTHENTICATION; $w = 4\,\mathrm{s}, o_m = 0.5$.

Activity	FPR (%)	BAC (%)
Stationary	1.6 (0.1)	98.0 (0.4)
Walking	1.7 (0.1)	97.9 (0.4)
PC use	1.7 (0.1)	98.0 (0.4)
Phone use	4.9 (0.3)	96.4 (0.4)
Same	7.2 (0.3)	95.2 (0.5)
All cases	1.9 (0.1)	97.8 (0.4)

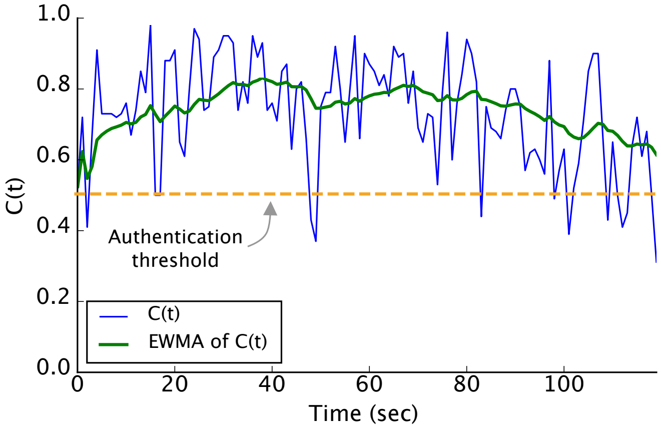

Fig. 4. Instantaneous confidence score for subject S1.

Sensors and Features. Recall from Section IV that M2MC uses 256 features for motion-to-motion correlation, including 40 temporal features. We hypothesized that temporal features are important in motion correlation to achieve low FNR and FPR required for secure applications. Consider the 'phone' use case in Table II with 200 Hz sampling rate, window size of 4 seconds, and overlap of 0.5 (which means a decision is made every 2 seconds). Table III shows that the temporal features clearly had a substantial impact on accuracy: the last two rows show CSAW accuracy without and with those temporal features. Table III also shows the accuracy when using only magnitudes (first row), only accelerometer (second row), and only gyroscope (third row). The fourth and bottom rows show accuracy when using both sensors but without and with temporal features, respectively. Temporal features substantially improved the accuracy.

2) Security: In this section, we evaluate the quality of CSAW to serve as a secure foundation for smartphone authentication. First, though, we must consider our adversary. Our adversary is a malicious individual with physical access to the smartphone, a curious family member or friend, or a curious colleague. The adversary may even be another authorized user

TABLE III. AVERAGE FPR, FNR, AND BAC DURING CONTINUOUS PHONE USE FOR DIFFERENT SUBSETS OF FEATURES AND SENSORS.

Features	FPR (%)	FNR (%)	BAC (%)
Only mag.	6.2 (0.9)	4.4 (0.6)	94.7 (0.5)
Only acc.	8.1 (1.0)	2.3 (0.6)	94.8 (0.3)
Only gyr.	9.1 (1.5)	5.5 (0.9)	92.7 (0.7)
All (No temporal)	7.4 (1.1)	4.0 (0.8)	94.3 (0.6)
All (with temporal)	4.8 (0.7)	2.1 (0.7)	96.5 (0.5)

(e.g., if the phone is a shared device and each authorized user is allowed access to only certain apps or information).

Our adversary seeks to achieve one of three related goals: (1) *Opportunistic snooping:* the adversary takes the opportunity to snoop the contents of the owner's phone when the owner is not near her phone; (2) *Stealing credentials:* the adversary steals the owner's credentials so he can use them to access the owner's device in her absence or access her accounts on other devices or websites; and (3) *Shadowing:* the adversary shadows the user so that he can gain access to her smartphone.

We assume the adversary **can** (i) observe (and film) the owner when she authenticates to her phone or when she is using her phone, (ii) collect the owner's touch inputs to her phone, and (iii) physically access the owner's phone, when she steps away leaving her phone behind.

We assume the adversary **cannot** (i) break the secure communications between watch and phone, (ii) obtain the encryption keys exchanged by the watch and the phone during their pairing, (iii) wear and use the owner's watch (recall Section I), and (iv) compromise the components that CSAW relies on to authenticate users – software and hardware on the watch, sensors and CSAW software on the phone.

In the face of the above threats, we can now describe how CSAW meets the following important security-related goals.

Verify the owner. CSAW can verify the individual who is actually using the phone. CSAW continuously provides a quantitative score regarding its confidence that the owner is the person actually using the device, based on 1) the fact that the owner's watch is physically proximate (in radio range and able to transmit wrist-movement data to the phone) and 2) the correlation between the owner's wrist movements and the phone's observed motion and inputs. Unlike approaches based purely on physical proximity, it is not sufficient for the owner to be near the device. In Section V-C1, we show that CSAW was highly accurate in determining whether the user was indeed the owner, that is, with fewer than 2% false-positive results.

Continuous. CSAW is specifically designed to continuously provide a quantitative score regarding its confidence that the owner is the person actually using the device, enabling the phone's operating system to deauthenticate the user whenever the confidence dips below a certain threshold. Thus, CSAW can continuously authenticate the owner when she is interacting with her phone, and deauthenticate (lock) when anyone else tries to use her phone.

Resilient to physical observation. In physical observation attacks, the attacker impersonates the user after observing her authenticate one or more times. Attacks include shoulder surfing, filming, or thermal imaging the touchscreen [50], [51]. Since CSAW does not require the user to enter a secret, none of these attack methods are available to the adversary.

Resilient to mimic attacks. In mimic attacks, the adversary gains access to the user's smartphone when the user leaves the smartphone unattended, and attempts to fool CSAW by providing inputs to the phone while observing and mimicking the user's wrist movements. Because CSAW authenticates a user based on phone-watch motion correlation, there is a possibility that an adversary who can successfully mimic the user's watch hand movements (in real time) while picking up or providing input to the phone can be mistakenly recognized as the legitimate user by CSAW. In our user study we evaluate this scenario by asking a participant (adversary) to mimic the researcher's (owner) wrist movements while performing two tasks: i) picking up a locked phone (to test initial authentication) and ii) using an unlocked phone (to test continuous authentication). Among the 135 pickup mimic attempts by 27 study participants, CSAW correctly identified 131 mimic attempts as true negatives (not from the user); thus, CSAW's FPR was 2.9% for initial authentication. In the phone-use test, CSAW correctly identified that all participants using the phone were not the owner, but the time varied: on average CSAW took 2.31 ($\pm$0.68) seconds to identify that the participant (adversary) was not the owner. Overall, CSAW is strongly resilient against mimic attacks. Since CSAW uses PIN as the fallback mechanism, an adversary might capture the user's PIN to bypass CSAW's initial authentication, but the adversary would be caught by CSAW's continuous authentication and locked out after 2-3 seconds.

VI. STUDY 2: OUT-OF-LAB EVALUATION

In our first study, CSAW proved quick, accurate, and usable for initial and continuous authentication with an accuracy of 99% and 96.5%, respectively, in a laboratory setting. In our second study, we sought to evaluate the feasibility of CSAW's initial authentication in an out-of-lab setting. The goal of this study was to answer the questions: (a) how often do participants use their phone in a way that CSAW might actually be useful or how often can CSAW save the need to enter the PIN, and (b) whether participants find CSAW useful as a phone unlock mechanism.

A. Participants

We used internal mailing lists and snowball sampling to recruit Android users. Eleven participants (8 male and 3 female; 6 undergraduate and 5 graduate students) enrolled in the study. Participants were required to have an Android phone and to use an Android watch during the study. Two of our participants had an Android watch that they regularly used; for others we loaned Android smartwatches for the duration of the study. The study was approved by our organization's IRB; participants received USD 15 to participate in the study.

B. Procedure

To enroll in the study, participants visited our lab. After signing the consent form, we installed the CSAW app on their Android phone. To set up CSAW's custom unlock, participants had to disable their existing unlock method. We walked them through CSAW's settings, where they first chose a 4-digit PIN as their fallback unlock method (in case CSAW's unlock failed) and then enabled CSAW's unlock mechanism. Participants were told that they had to wear their Android watch during the study, but they could wear the watch on either hand. We asked participants to test CSAW's unlock method during their lab visit until they were comfortable with its use. We asked participants to use CSAW as their unlock method during the study. We warned participants that CSAW is a test prototype and as such may be less secure than their existing unlock method under certain conditions (e.g., if someone manages to kill the CSAW app from a locked phone, they can bypass CSAW's security). Participants could withdraw from the study if they had any concerns (no one did).

During the study, CSAW performed initial authentication for users. (In this study, CSAW did not do continuous authentication, because the study was specifically designed to test the feasibility of initial authentication in an out-of-lab setting.) When participants picked up their phone with their watch hand, CSAW would automatically attempt to authenticate and unlock the phone. When CSAW failed to unlock after pickup (either because the phone was picked up with the non-watch hand or a failure in CSAW), CSAW's custom unlock screen allowed participants to unlock the phone with a rotate gesture (with the phone in the watch hand) or by entering their chosen PIN; thus, if participants wanted they could ignore CSAW and simply use PIN to unlock their phone. Every time CSAW was successful, it saved participants from having to enter a PIN, but when it failed, it did not add any additional burden because participants could use a PIN as they normally would.

The study lasted for five days for each participant, and throughout the study, the CSAW app logged all phone and watch motion during phone pickup and phone use, and all the events regarding authentication (e.g., whether CSAW succeeded in an authentication attempt, if the user chose the rotate action or entered their chosen PIN). At the end of the study, we conducted a semi-structured interview with the participants where we asked them to share their experiences and fill out a brief survey about their phone use and CSAW's usability.

C. Result

To answer our study objective, we first analyze participants' phone unlock data to determine how often CSAW might be useful for initial authentication, and then present participants' subjective feedback on the usability of CSAW.

How often CSAW might be useful. CSAW's utility depends on how a participant picks up her phone (e.g., with watch hand, in which case CSAW can authenticate, vs. non-watch hand, for which CSAW cannot authenticate). However, to determine how participants pick up their phone, we would have to either ask the participants to manually self-report, after every phone pickup, how they picked up their phone or ask the participants to carry an additional device that may automatically capture this information. Both options significantly increase participant burden in the study, so we decided to use the actual phone unlock data to estimate the instances when CSAW might be useful.

Collectively, all participants performed 2,707 phone unlocks during the study, with 54.14 unlocks per day on average ($std =$ 18.9). Some of the prior studies observed that people unlock their phones on average about 40 times per day, but interact with their phone (e.g., turn on display to see notifications without unlocking their phone) on average about 70 per day [52]. In our implementation of CSAW, participants had to unlock their phone even to see notifications, which may explain the higher number of unlocks per day. Overall, 70% of phone unlocks were performed with CSAW (about 45% were from the pickup gesture and 25% were from the rotate gesture); the other 30% of unlocks were with PIN. The PIN unlock could be because CSAW failed and the participant chose to use PIN, or because the participant picked up the phone with their non-watch hand and could not use CSAW to unlock the phone and used PIN instead. As mentioned earlier, we do not have ground truth in our data to distinguish these two cases, and moreover, the point of this study was not to measure CSAW's accuracy, but to gauge how often CSAW might be useful as an initial authentication method. And from these preliminary results, it appears CSAW was useful for 70% of authentication attempts, i.e., saving participants from having to enter their PIN 70% of the time.

The fraction of unlocks per day that were performed with CSAW reduced as the study progressed (Figure 5). We believe this trend might be due to the novelty effect or the researcher bias, which may have caused participants to change their behavior to make an effort to use CSAW (e.g., remembering to use watch-hand to pick up the phone), but a few days into the study participants reverted to their usual habit of picking up the phone. Thus, the number of CSAW unlocks towards the end of the study are perhaps a more correct reflection of how useful participants found CSAW. It is encouraging that the majority of phone unlocks per day, even on the last day of the study, were with CSAW.

When we asked participants whether they noticed any change in their behavior during the study, five participants said they did: three participants reported that they switched their watch to the hand they normally use to pick up their phone; two participants said they did notice small changes in their phone pickup behavior, and for one of them the change in behavior was worth the usability benefit.

> *I did notice I was using my watch-hand more to pick up the phone. I thought this is what the study is about and this is what I am supposed to do. But I actually found it convenient to unlock it that way [with CSAW], and then I did not mind the change.*
> — 27yr, Male, Student

User feedback on CSAW. In the interviews after the study, we asked participants questions about their phone use and their experience using CSAW. Figure 6 shows the answers to the Likert-scale questions in our survey. Shoulder surfing during authentication is a concern many participants shared, and one of the advantages of CSAW is that it is resilient to shoulder surfing because there is no secret that is visibly entered (like PIN or swipe) during authentication. Participants did report feeling less concerned about shoulder surfing with CSAW.

Five participants (45%) felt CSAW was easier to use than PIN, 3 participants (27%) felt it was equally easy, whereas three participants said PIN was actually easier for them than CSAW. When we probed these three participants, they said they found it easier to type PIN than perform the rotate gesture

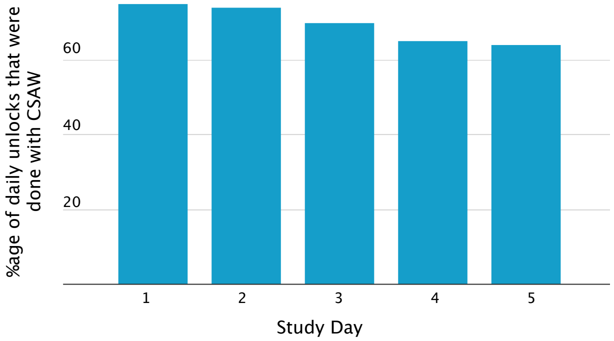

Fig. 5. Average percentage of unlocks per day per participant that were performed with CSAW during the five days of the study.

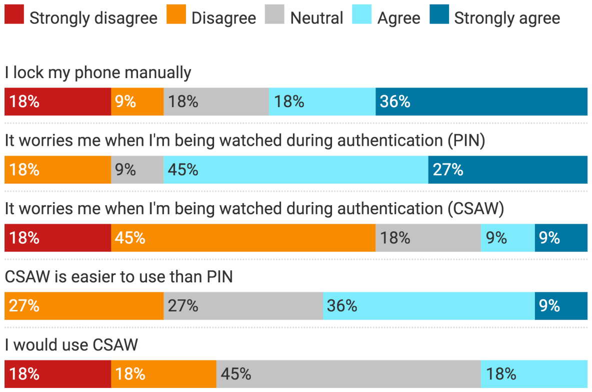

Fig. 6. Participant ratings on their phone use and aspects of CSAW.

and they did not like wearing a watch, which influenced their usability assessment of CSAW. The burden to carry a watch was also shared by other participants who do not wear a watch. When asked whether they would continue to use CSAW, nine out of 11 participants were either neutral or said no; the two participants who said they would use CSAW were participants who already wore (before the study) Android Watch on a daily basis. Thus, CSAW provides most usability benefit to smartphone users who already wear a smartwatch, by adding one more utility – authentication – to the watch. Overall, this study provides promising preliminary evidence that CSAW will be successful in real-world settings (at least for users who wear a watch), but a longer study (with more participants) is required to draw generalizable conclusions about CSAW's usability.

VII. Discussion

CSAW shows encouraging results, but there are some limitations and opportunities for future work.

Single-handed use. In Section V-C1, we show that CSAW performs well when the watch is worn on the hand that holds the phone or on the hand that touches the phone screen. There are moments, however, when the owner's watch hand may be uninvolved – neither holding the phone nor touching the phone; for example, the user may wear the watch on her right hand but use only her left hand to hold the phone and touch the screen. In such cases, CSAW is unable to verify the owner as the user, and would have to rely on alternative means for authentication (e.g., require a PIN code), or ask the user to change the phone grip. For some users (like one of our participants) the gains in usability may outweigh the inconvenience of minor deviations from natural interactions caused by this issue.

Securing the smartwatch. CSAW secures the phone, but not the watch. However, fitness bands and smartwatches are considered personal wearable devices and are not likely to be shared with others. Furthermore, CSAW could securely link the watch to its owner (to prevent any unintended sharing) by detecting when the watch is removed and requiring its owner to authenticate to the watch (e.g., by using a PIN on the watch [6] or using a biometric approach [7]). Alternatively, users could log into a phone using the owner's PIN (or other authentication method) while wearing the watch, and thereafter CSAW can assume that the watch wearer is the owner of the phone.

Bluetooth pairing. Some use cases (particularly access limitation and delegation) require the wireless technology to allow pairing multiple watches to multiple phones, and allow simultaneous communications between a watch and multiple phones. BLE 4.1 supports scatternet operation so this capability is now becoming available. Such cases are more likely for tablets than smartphones; we envision a user working with multiple phones and tablets, or a school/workplace in which tablets are shared among several users.

VIII. Summary and Conclusions

In this paper we introduce CSAW, a novel approach to unobtrusively and continuously authenticate a smartwatch user to her smartphone. CSAW verifies the user by correlating her wrist motion with the phone's motion when she is holding the phone in her watch-hand, and with the phone's inputs if she is using her watch-hand to provide input. Based on the correlation, CSAW continuously provides a confidence score that can act as a foundation for other subsystems and applications to verify the user and take appropriate policy decisions. Our evaluation from an in-lab user study shows that CSAW can authenticate users with 99% accuracy using the simple natural phone pick-up action. During continuous phone use, CSAW could verify the user with 96.5% accuracy every 2 seconds. In our preliminary out-of-lab study, participants performed about 70% phone unlocks with CSAW, suggesting that CSAW could be useful to reduce authentication burden. CSAW is also power efficient: it draws less than 2% power on the phone and the watch by optimizing the use of sensors. CSAW's confidence score can feed a variety of authentication-decision algorithms, allowing each to be tuned to achieve high security (very low FPR) or high usability (low FNR). Moreover, CSAW uses hardware commonly found in most fitness bands and smartwatches, so CSAW could be integrated into such devices without adding hardware. Finally, CSAW could extend to tablets and may be applicable to other kinds of handheld devices like health devices or mobile payment handsets.

Acknowledgments

We thank our anonymous reviewers and our shepherd, Yasemin Acar, for their valuable feedback. 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 CNS-1329686.

References

[1] D. Marques, I. Muslukhov, T. J. V. Guerreiro, L. Carriço, and K. Beznosov, “Snooping on mobile phones: Prevalence and trends,” in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, 2016. Available online: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/marques

[2] C. Herley, “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, 2009. DOI 10.1145/1719030.1719050

[3] D. Hintze, P. Hintze, R. D. Findling, and R. Mayrhofer, “A large-scale, long-term analysis of mobile device usage characteristics,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 2, Jun. 2017. DOI 10.1145/3090078

[4] (2013) Survey reveals consumers exhibit risky behaviors despite valuing their privacy on mobile devices. Accessed December 2018. Available online: https://www.lookout.com/news-mobile-security/sprint-lookout-mobile-privacy-survey

[5] A. Adams and M. A. Sasse, “Users are not the enemy,” *Communications of the ACM*, vol. 42, no. 12, Dec. 1999. DOI 10.1145/322796.322806

[6] Apple Watch. Accessed December 2018. Available online: http://www.apple.com/watch/

[7] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz, “A wearable system that knows who wears it,” in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, Jun. 2014. DOI 10.1145/2594368.2594369

[8] J. Lester, B. Hannaford, and G. Borriello, “"Are you with me?" - Using accelerometers to determine if two devices are carried by the same person,” in *Proceedings of the International Conference on Pervasive Computing (Pervasive)*, 2004. DOI 10.1007/978-3-540-24646-6_3

[9] R. Mayrhofer and H. Gellersen, “Shake well before use: Intuitive and secure pairing of mobile devices,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, Jun. 2009. DOI 10.1109/TMC.2009.51

[10] C. Cornelius and D. Kotz, “Recognizing whether sensors are on the same body,” *IEEE Pervasive Computing*, vol. 6696, June 2011. DOI 10.1007/978-3-642-21726-5_21

[11] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, “ShakeUnlock: Securely transfer authentication states between mobile devices,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, April 2017. DOI 10.1109/TMC.2016.2582489

[12] S. Mare, R. Rawassizadeh, R. Peterson, and D. Kotz, “SAW: Wristband-based authentication for desktop computers,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 2, no. 3, Sep. 2018. DOI 10.1145/3264935

[13] S. Mare, A. Molina-Markham, C. Cornelius, R. Peterson, and D. Kotz, “ZEBRA: Zero-effort bilateral recurring authentication,” in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, May 2014. DOI 10.1109/SP.2014.51

[14] A. Acar, H. Aksu, S. Uluagac, and K. Akkaya, “WACA: Wearable-assisted continuous authentication,” in *IEEE Symposium on Security & Privacy workshop on Bio-inspired Security, Trust, Assurance and Resilience (BioStar)*, 2018. Available online: https://arxiv.org/abs/1802.10417

[15] Q. Yue, Z. Ling, X. Fu, B. Liu, W. Yu, and W. Zhao, “My Google Glass sees your passwords!” BlackHat, 2014, accessed December 2018. Available online: http://www.cs.uml.edu/~xinwenfu/paper/VisionBasedAttack_Fu_2014.pdf

[16] F. Maggi, S. Gasparini, and G. Boracchi, “A fast eavesdropping attack against touchscreens,” in *Proceedings of the International Conference on Information Assurance and Security (IAS)*. IEEE, Jan. 2011. DOI 10.1109/ISIAS.2011.6122840

[17] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, “Smudge attacks on smartphone touch screens,” in *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*. USENIX Association, 2010. Available online: https://www.usenix.org/legacy/events/woot10/tech/full_papers/Aviv.pdf

[18] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu, “Fingerprint attack against touch-enabled devices,” in *Proceedings of the Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*. ACM, Oct. 2012. DOI 10.1145/2381934.2381947

[19] A. Roy, N. Memon, and A. Ross, “MasterPrint: Exploring the vulnerability of partial fingerprint-based authentication systems,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, Sep. 2017. DOI 10.1109/TIFS.2017.2691658

[20] A. D. Luca, E. von Zezschwitz, N. D. H. Nguyen, M.-E. Maurer, E. Rubegni, M. P. Scipioni, and M. Langheinrich, “Back-of-device authentication on smartphones,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2013. DOI 10.1145/2470654.2481330

[21] D. Buschek, F. Hartmann, E. von Zezschwitz, A. D. Luca, and F. Alt, “SnapApp: Reducing authentication overhead with a time-constrained fast unlock option,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2016. DOI 10.1145/2858036.2858164

[22] F. Schaub, M. Walch, B. Könings, and M. Weber, “Exploring the design space of graphical passwords on smartphones,” in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*. ACM, 2013. DOI 10.1145/2501604.2501615

[23] E. von Zezschwitz, A. De Luca, B. Brunkow, and H. Hussmann, “SwiPIN: Fast and secure PIN-entry on smartphones,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2015. DOI 10.1145/2702123.2702212

[24] “2018 User risk report,” Proofpoint, a division of, Wombat Security, accessed December 2018. Available online: https://www.wombatsecurity.com/user-risk-report

[25] S. Das, G. Laput, C. Harrison, and J. I. Hong, “Thumprint: Socially-inclusive local group authentication through shared secret knocks,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2017. DOI 10.1145/3025453.3025991

[26] Y. Yang, G. Clark, J. Lindqvist, and A. Oulasvirta, “Free-form gesture authentication in the wild,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2016. DOI 10.1145/2858036.2858270

[27] W.-H. Lee, X. Liu, Y. Shen, H. Jin, and R. B. Lee, “Secure pick up: Implicit authentication when you start using the smartphone,” in *Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2017. DOI 10.1145/3078861.3078870

[28] H. Khan, U. Hengartner, and D. Vogel, “Augmented reality-based mimicry attacks on behaviour-based smartphone authentication,” in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 2018. DOI 10.1145/3210240.3210317

[29] V. Sharma and R. J. Enbody, “User authentication and identification from user interface interactions on touch-enabled devices,” in *Proceedings of the ACM Conference on Wireless Network Security (WiSec)*, 2017. DOI 10.1145/3098243.3098262

[30] X. Wang, T. Yu, O. J. Mengshoel, and P. Tague, “Towards continuous and passive authentication across mobile devices: An empirical study,” in *Proceedings of the ACM Conference on Wireless Network Security (WiSec)*, 2017. DOI 10.1145/3098243.3098244

[31] L. Li, X. Zhao, and G. Xue, “Unobservable re-authentication for smartphones,” in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, 2012. Available online: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/02_1_0.pdf

[32] H. Xu, Y. Zhou, and M. R. Lyu, “Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones,” in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, 2014. Available online: https://www.usenix.org/conference/soups2014/proceedings/presentation/xu

[33] S. G. Kratz and M. T. I. Aumi, “AirAuth: A biometric authentication system using in-air hand gestures,” in *Proceedings of the Conference on Human Factors in Computing Systems: Extended Abstracts (CHI EA)*, 2014. DOI 10.1145/2559206.2574797

[34] T. Feng, X. Zhao, B. Carbunar, and W. Shi, “Continuous mobile authentication using virtual key typing biometrics,” in *Proceedings of the International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013. DOI 10.1109/TrustCom.2013.272

[35] X. Wang, T. Yu, M. Zeng, and P. Tague, “XRec: Behavior-based user recognition across mobile devices,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 3, Sep. 2017. DOI 10.1145/3130975

[36] H. Crawford and E. Ahmadzadeh, “Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics,” in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, 2017. Available online: https://www.usenix.org/conference/soups2017/technical-sessions/presentation/crawford

[37] T. C. Meng, P. Gupta, and D. Gao, “I can be you: Questioning the use of keystroke dynamics as biometrics,” in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, 2013. Available online: http://flyer.sis.smu.edu.sg/ndss13-tey.pdf

[38] A. Serwadda and V. V. Phoha, “When kids’ toys breach mobile phone security,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2013. DOI 10.1145/2508859.2516659

[39] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, “Progressive authentication: Deciding when to authenticate on mobile phones,” in *Proceedings of the USENIX Security Symposium (USENIX Security)*, 2012. Available online: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/riva

[40] T. Vu and M. Gruteser, “Personal touch-identification tokens,” *IEEE Pervasive Computing*, vol. 12, no. 2, April 2013. DOI 10.1109/MPRV.2013.33

[41] P. Nguyen, U. Muncuk, A. Ashok, K. R. Chowdhury, M. Gruteser, and T. Vu, “Battery-free identification token for touch sensing devices,” in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, 2016. DOI 10.1145/2994551.2994566

[42] M. Azimpourkivi, U. Topkara, and B. Carbunar, “Camera based two factor authentication through mobile and wearable devices,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 3, Sep. 2017. DOI 10.1145/3131904

[43] F. Stajano, “Pico: No more passwords!” in *Security Protocols XIX. Security Protocols 2011*, B. Christianson, B. Crispo, J. Malcolm, and F. Stajano, Eds. Springer-Verlag Berlin, Mar. 2011, vol. 7114. DOI 10.1007/978-3-642-25867-1_6

[44] H.-M. C. Leung, C.-W. Fu, and P.-A. Heng, “TwistIn: Tangible authentication of smart devices via motion co-analysis with a smartwatch,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 2, no. 2, Jul. 2018. DOI 10.1145/3214275

[45] (2017, Dec.) Use gestures on your device - Pixel phone help. Available online: https://support.google.com/pixelphone/answer/7443425?hl=en

[46] (2017, Dec.) Use raise to wake on your iPhone - Apple support. Available online: https://support.apple.com/en-us/HT208081

[47] (2016, Feb.) Shimmer Research. Available online: http://www.shimmersensing.com

[48] (2016, Mar.) Enron email dataset. Available online: http://www.cs.cmu.edu/~enron/

[49] D. R. Velez, B. C. White, A. A. Motsinger, W. S. Bush, M. D. Ritchie, S. M. Williams, and J. H. Moore, “A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction,” *Genetic Epidemiology*, vol. 31, no. 4, 2007. DOI 10.1002/gepi.20211

[50] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, “Beware, your hands reveal your secrets!” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. ACM, 2014. DOI 10.1145/2660267.2660360

[51] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, “Blind recognition of touched keys on mobile devices,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2014. DOI 10.1145/2660267.2660288

[52] M. Harbach, A. D. Luca, and S. Egelman, “The anatomy of smartphone unlocking: A field study of Android lock screens,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2016. DOI 10.1145/2858036.2858267