Biometric Touch Sensing: Seamlessly Augmenting Each Touch with Continuous Authentication

Christian Holz and Marius Knaust
Yahoo Labs, Sunnyvale, CA
{christianh, mariusk} @ yahoo-inc.com

Figure 1: We enable commodity touchscreens to biometrically identify and authenticate users on *every* touch through the touchscreen itself. We seamlessly integrate continuous authentication into touch interaction, which fully replaces password dialogs. (a) Here, a tablet displays the home screen right away. (b) When touching the Mail icon, the tablet identifies the user and blocks unauthorized access. (c) When a registered user touches Mail, the device authenticates them and (d) opens *their* Mail. (e) Our watch prototype *Bioamp* senses biometric properties and modulates a high-frequency signal onto the user's skin, from which the touchscreen obtains the biometric features, identifies the user, and continuously authenticates them for each interaction.

ABSTRACT

Current touch devices separate user authentication from regular interaction, for example by displaying modal login screens *before* device usage or prompting for in-app passwords, which interrupts the interaction flow. We propose *biometric touch sensing*, a new approach to representing touch events that enables commodity devices to seamlessly integrate authentication into interaction: From each touch, the touchscreen senses the 2D input coordinates and at the same time obtains biometric features that identify the user. Our approach makes authentication during interaction transparent to the user, yet ensures secure interaction at all times. To implement this on today's devices, our watch prototype *Bioamp* senses the impedance profile of the user's wrist and modulates a signal onto the user's body through skin using a periodic electric signal. This signal affects the capacitive values touchscreens measure upon touch, allowing devices to identify users on each touch. We integrate our approach into Windows 8 and discuss and demonstrate it in the context of various use cases, including access permissions and protecting private screen contents on personal and shared devices.

ACM Classification Keywords:

H.5.2 [Information interfaces and presentation]: User Interfaces. Input devices & strategies.

UIST '15, November 08–11, 2015, Charlotte, NC, USA

ACM 978-1-4503-3779-3/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2807442.2807458

INTRODUCTION

User authentication on current touchscreen devices is typically performed separately from normal interaction with all apps. Password-protected devices display login screens *before* permitting regular interaction with the system. On shared devices, such as tablets, one of their purposes is logging in the correct user and loading their respective profile. The more general use is to *control access* to the device, however, including access to installed apps and stored data.

Logging into a system typically unlocks the user's entire account; access to personal data is wide open without further verification until locking the device or the current login session times out [25]. All applications run in the context of the logged-in user's identity, providing access to personal data, such as emails to everyone with physical access to the device.

More disconcertingly, mobile devices increasingly contain highly sensitive data. This includes payment information that many applications store and reuse without any user authentication (e.g., Amazon, Square Cash, and Uber on iOS). All of these readily charge stored credit cards, relying solely on the initial login screen that unlocks the device. However, many users have not enabled logins screens [19] out of convenience [4] or use only weak PIN codes [1]. Some applications thus prompt for passwords *during* usage to verify access to personal data (e.g., in-app purchase or banking apps), interrupting the flow of regular interaction with modal dialogs.

In this paper, we propose incorporating user authentication on commodity touch devices *into* all interaction. Our approach, *biometric touch sensing*, notifies running applications not just of 2D touch events, but also of the user's identity for each touch, such that apps can authenticate the user upon touch. Devices obtain the identity through the touchscreen—the *same* sensor that detects the touch event.

BIOMETRIC TOUCH SENSING: TOUCH := (X,Y,USER ID)

Figure 1 demonstrates biometric touch sensing during the use of a traditional tablet device. The device implements our sensing approach on the operating system level, identifying and authenticating users *each* time they touch the screen. (a) The tablet greets users with the start screen and shows available apps right away. (b) When this user attempts to launch Mail by touching the icon, the system detects the touch event, biometrically identifies the user from the touch, and forwards the event to the launcher. Since this user has no local account, the launcher rejects the input and displays a message. (c) When a registered user touches the Mail icon, the system repeats this procedure, verifies that the identified user has an account on this device, authenticates the user to access the data, and (d) launches Mail using the user's credentials, which loads their settings and retrieves their emails.

The key feature of biometric touch sensing is what is *absent* from all interaction: password dialogs that introduce an *explicit* step for authentication and interrupt direct interaction. Instead, systems that implement our approach continuously authenticate users on a per-touch basis during interaction.

Authentication is fully transparent to the user, because the operating system handles authentication as part of all interaction. In our case, the system verifies permissions and authenticates users *post-input* as shown in Figure 2b. Here, an app displays store contents right away (instead of prompting for credentials before use) and informs the user about the *outcome* of a touch event, here denying the purchase of a game.

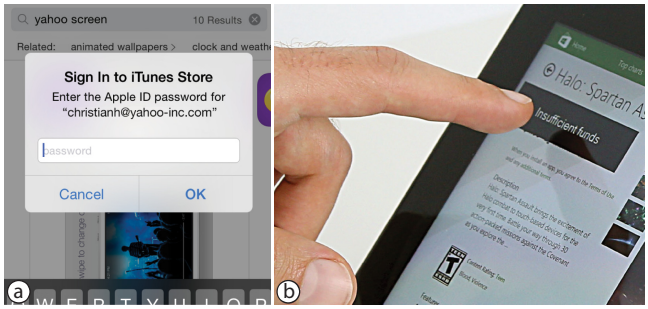

Figure 2: (a) Traditional login dialogs or in-app password prompts interrupt interaction for authentication. Using biometric touch sensing, (b) touchscreens authenticate users upon each touch and verify permissions before starting an operation, here denying an in-app purchase action for this user.

Since today's commodity touchscreens cannot yet sense a user's biometric features from a touch, we build on wearable devices with biometric sensors. Our prototype *Bioamp* senses biometric features and sends them through the user's body, such that the tablet can identify the user from each touch. Bioamp thereby modulates a code onto the body through a high-frequency alternating voltage. The touchscreen detects this as fluctuations in its signal, which we obtain through a debug interface. To compensate for the low update rates on commodity devices, Bioamp sends the biometric information using Bluetooth and modulates an identifier onto the skin. The tablet then looks up the respective biometrics and *directly* associates them with the touch to authenticate the user for the input event.

As biometric features, we use impedance measurements through the user's body [11], specifically the impedance profile of the user's wrist [8], which has been proven user-unique and robust over time [7]. Bioimpedance is adequate for the use case we envision: seamless user authentication on shared tablet devices, such as in a family household.

CONTRIBUTION: NO MORE PROMPTS OR TIMEOUTS

Our main contribution is a new representation of input events on commodity multitouch devices and the resulting interaction, which allows applications to continuously authenticate users upon each touch. While traditional devices sense 2D touch coordinates, biometric touch sensing obtains the user's identity from the *same* touch contact on the operating system level and provides touch events in the form of (*x*, *y*, *user ID*) to applications. This seamlessly integrates user authentication into all touch interaction—fully transparent to the user. The resulting interaction with touch devices thus requires no interruptions through password prompts for authentication purposes or session timeouts to prevent unauthorized access.

Biometric touch sensing renders the use of passwords obsolete, demoting it to a fallback mechanism. In case the user's identity or biometric data is unavailable, touch interaction continues to function, but will require traditional passwords again, which makes our approach backward compatible.

Until commodity touchscreens are capable of sensing biometric features directly from the touch contact, our standalone prototype Bioamp *converts* users' biometric features into an electric signal that the touchscreen observes upon touch. Notably, Bioamp merely relays these features, whereas the touch device actually *identifies* the user. We demonstrate how to process this signal on commodity touchscreens at the example of a Surface 2 Pro/Windows 8.1 tablet without any hardware changes and explain the integration of our approach on the operating system level.

While we chose bioimpedance as *one possible* type of biometric feature, biometric touch sensing generalizes to other types of biometrics, such as capturing users' fingerprints directly upon touch [14]. Similarly, to complement existing touch devices, Bioamp's approach also generalizes beyond sensing impedance profiles, anticipatory of potential future smartwatches or head-mounted devices that may sense unique features and transmit them through the body.

RELATED WORK

Wearable biometric sensors: Several wearable devices sense biometrics, such as health-tracking watches that record the user's pulse (e.g., Apple Watch). Nymi is a wristband that senses heart rhythms for identification using ECG [5]. While ECG patterns afford user identification [17,27], they vary with activity and require touching the device with both hands. Nymi thus identifies users only once a day and then acts as a token until taken off. It then authenticates users on *other* systems through proximity, which lacks the directness of touch and breaks with multiple simultaneous users.

Cornelius et al. introduced a wrist-worn device that identifies the wearer using bioimpedance of the wrist [8], thoroughly

evaluated the reliability of wrist impedance [7,8], and found a 98% balanced-accuracy for eight participants over a day's worth of samples. They also proved the suitability of wrist bioimpedance for long-term identification, showing that impedance values remained stable enough for verification 4.5 months later. This inspired us to reuse bioimpedance of the wrist as one possible feature that enables commodity touch devices to identify users biometrically on a per-touch basis.

Using the body as an electric conduit: The human body has been explored as a conductor between the same or more devices in the context of body-area networks. Zimmerman's work connects two devices through the human body to keep the connection personal, i.e., impossible to capture for external devices [28]. Baldus et al. explore the same effect using multiple wearable prototypes and quantify the throughput of through-body data transmission [2]. Matsushita et al.'s wearable key is a wrist worn device that modulates its ID onto the user's body, such that a custom area electrode and sensor can detect the user upon touch [21]. Our current implementation builds on this work using a similar form factor. More importantly, however, our implementation runs on commodity touch devices that use ITO-based drive and sense lines and built-in touch chips that perform signal filtering.

User identification on touch devices has been an active research area for tabletop systems. Some techniques extend to tablets that provide the capacitive touch image. For example, HandsDown identifies users by their hand contours [24]. Bodyprint appropriates the touchscreen as an image sensor to scan their ear or hand geometry for identification [16].

Beyond identification, some systems associate a touch with a particular user. DiamondTouch is a table that electrically connects chairs to the surface, closing a circuit when a sitting user touches the table [9]. While this does not identify users, it detects the chair they sit in. Bootstrapper identifies users based on their shoes and traces their arm back to a table edge to associate touches with users [22]. Harrison et al. showed identification of two users on capacitive touchscreens [13], building on Touché [23] for impedance measurements through the user's body. Also related to our prototype Bioamp is Vu et al.'s system [26], which uses rings as identification tokens. Each ring transmits an electric pattern that the touchscreen observes when users press the ring against it. All these systems do not sense unique biometric values, but instead distinguish or identify users through explicit mappings. Finally, in previous work, we used a fingerprint scanner to sense touch input with high accuracy by modeling it on a per-user level [15]. We also introduced Fiberio, a tabletop system that integrates fingerprint scanning and diffuse illumination into the same surface in a custom-built table prototype to scans users' fingerprints during interaction on a display [14].

Previous systems that associate touch events with users have mostly focused on the sensing component using custom-built prototypes. In this paper, we bring per-touch authentication to commodity touchscreen devices and operating systems, and demonstrate the resulting interaction for users.

A NEW REPRESENTATION OF TOUCH INPUT EVENTS

Devices that implement biometric touch sensing have a notion of *who* is invoking *which* input action on multitouch devices at all times, both in single user as well as in multi-user scenarios. Thus, user interfaces can seamlessly integrate user authentication into all touch interaction with the system.

Biometric touch sensing streamlines secure interaction with apps and the operating system and discards interruptions that result from traditional login screens or password prompts to authenticate individual operations. Instead, the system shows privacy-insensitive content *right away*. Only upon receiving input events does the system check who has touched an item on the screen, what access permissions are necessary to invoke it, and if the identified user is authorized to do so.

If a user is authorized to invoke the associated action, the operating system allows the touch event, customizes it with the user's profile, for example to launch apps with the correct credentials as shown in Figure 1, and executes the action. If the user is not authorized, however, the operating system blocks the event and displays a feedback message.

From a user's point of view, the operating system behaves as if there were only one user with exclusive access: (repeated) authentication becomes unnecessary, because there is seemingly only one context in which all applications run. Even so, the system repeatedly authenticates the user, just in a way that is transparent to the user. Biometric touch sensing thus combines the convenience of single sign-on with the security of authenticating every single action and interaction.

To implement biometric touch sensing on current touch devices, we prototyped a wearable device that complements the touchscreen to accomplish biometric sensing on each touch.

BIOAMP: COMPLEMENTING COMMODITY TOUCHSCREENS

Figure 3 shows our prototype Bioamp. It comprises three parts: communication and user interface, biometric sensing, and data transfer to the touch device through the user's body.

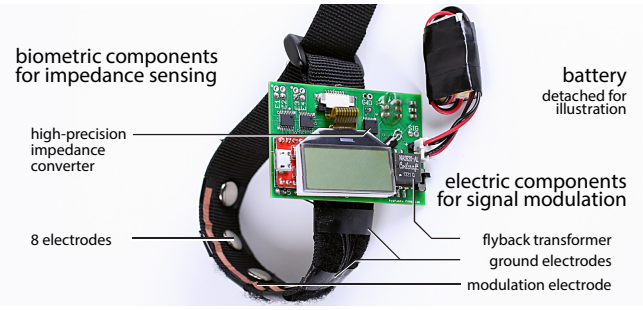

Figure 3: Our watch-like prototype Bioamp comprises biometric sensors to capture user-unique features and electric parts to transmit signals to the touchscreen through the body.

Processor, user interface and communication

A Blend Micro drives Bioamp, featuring an Atmel ATmega32U4 microprocessor and a Nordic nRF8001 Bluetooth Low Energy chip for communication with peripheral devices, which we use for fast data transmissions to the touch device to compensate for low touchscreen sampling rates. Two 110 mAh LiPo batteries power our prototype.

Sensing biometric features: bioimpedance of the wrist

As shown in Figure 3, eight electrodes at 20 mm steps are attached to the inside of Bioamp's band and capture impedance measurements. While the band is tight similar to health-sensing wearables, the snap-button electrodes cause no strain. Two ADG1608 analog 8-channel multiplexers connect the electrodes to an AD5933 high-precision impedance converter, which measures impedances between electrode pairs by sweeping through a range of twelve frequencies.

Bioamp measures impedances from twelve combinations of electrode pairs: four opposite-electrodes pairs and eight pairs with two electrodes between the pair (e.g., Electrode 1 and 4, 2 and 5, etc.). In essence, this collects impedances for each electrode and its three respective opposite electrodes, which we have found to work more reliably in tests than pairs of electrodes that are one electrode apart (e.g., [7,8]). Bioamp extracts the impedance phase and magnitude from each series of impedance measurements for a given electrode pair.

During each measurement, Bioamp collects 288 values (12 pairs × 12 frequencies × {magnitude, phase}) and derives 96 statistical features. For each electrode pair, we calculate a linear regression of the log transformed magnitude and phase responses, respectively, and extract their slopes and intercepts (2×24 values). We then compute the *differences* of impedances that result from neighboring pair combinations to integrate local impedance changes around the wrist and extract the same two feature types (another 2×24 values).

Sensing intervals and data transfer to the touchscreen

When a user puts on the band, Bioamp performs a number of impedance measurements to collect biometric features. Since Bioamp senses contact with skin, it is sufficient to collect biometric values initially and then ensure that the *same* user is wearing the device during further use. When Bioamp detects that the user has taken off the band, it stops transmitting signals, waits for the band to be put on again, and repeats the biometric measurements for subsequent modulation.

The display on Bioamp serves two purposes: While Bioamp initially displayed the time to act as a watch, the display more importantly serves as an *anchor* for users. Strong rotation of the wristband changes the captured biometric values [7], but our display causes users to put on the band such that electrodes sit roughly at the same location around the wrist.

Importantly, Bioamp does *not* perform any biometric user identification. It merely captures biometric features and forwards them to the touchscreen. If another user puts on the *same* device, Bioamp resets, captures fresh biometric values, and forwards the second user's biometric measurements.

Data transfer to the touch device

Bioamp modulates an electric signal onto the user's body through their skin, which the touchscreen detects upon touch. We thereby build on the properties of the human body to conduct oscillating voltages [28]. The touchscreen thus senses not just changes in capacitance between the surface and the finger, but also the fluctuations due to the modulated signal.

The challenge to let two separate and *unconnected* devices exchange electric signals through the user's body is illustrated in Figure 4: The two devices do not share a common ground, which complicates exchanging signals, because both devices have individual ground references [12].

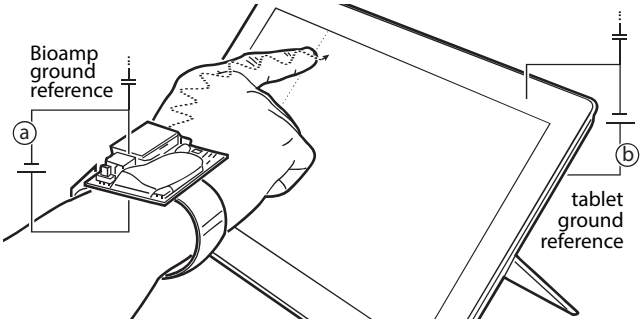

Figure 4: Data transfer. (a) Bioamp has a power source and ground reference and so does the touch device (b). Both connect through the body, but do not share a common ground.

When both devices share a common ground, a *Vpp* = 5V signal that is modulated onto the body produces a visibly strong effect on the touchscreen. However, the signal disappears with individual ground references, since both devices then only capacitively couple to the environment through air.

Bioamp thus creates a high-frequency signal and modulates this onto the user's body. As shown in Figure 3, we use a Coilcraft NA5920 flyback transformer to create a *Vpp* = 50V 150 kHz signal, which is in the range of the touch chip's frequencies. The high voltage causes a current draw through the body that is strong enough to affect the sensing on the touchscreen and persevere the filters on touch processors in commodity devices. At the same time, the resulting current is low and causes no shock to occur when the user touches ground. We maximize Bioamp's ground coupling to air with a strip of tape-covered copper around the outside of the band.

Since we have full control over Bioamp, yet limited control over the commodity touchscreen device and its sensing mechanism, the unmodified touch device determines the type of data encoding we can use. We therefore describe data encoding and decoding after the sensing part.

SENSING PLATFORM: AN UNMODIFIED TOUCH DEVICE

We sense the modulated signal on a commodity touchscreen device with unmodified hardware, which implements traditional 2D touch event detection. The majority of such touchscreens implement projected capacitive sensing [3], using a matrix of drive and sense lines to detect touch locations. In our case, we use a Microsoft Surface 2 Pro running Windows 8.1 to detect touch events and the modulated signal.

The difficulty of sensing a modulated signal upon touch is that the touch chip inside current touch devices is designed to *filter* the capacitive readings to reliably detect and track touch events. External factors, such as power lines may produce interfering signals that manifest themselves as noise in the sensed values. This is particularly true when the touch device is not plugged into a power outlet.

To detect Bioamp's signal from a touch contact, we essentially need to jam the sensing mechanism, such that (a part of) our signal comes through the filtering on the touch chip. The signal will then appear in the touch data that we obtain in software on the touchscreen device. We thus require the capacitive measurements across the entire screen.

We obtain the "raw" capacitive sensor data on the touch device using the debugging interface for testing touchscreen functionality that is implemented in the chip [16]. Specifically, we collaborated with Atmel to get capacitive readings using their 'Object Server' and 'Hawkeye' tool, which is a software solution to get a signed 8-bit capacitive image. Importantly, we obtain debug values only *after* the touch chip has filtered the raw sensor values. Unfortunately, Atmel did not provide us with an option to disable or reduce the amount of filtering, which limits our flexibility in using encoding schemes. Atmel's debug image updates at irregular intervals around 29 Hz. Figure 5a shows the capacitive image we get from a single touch without and (b) with a modulated signal.

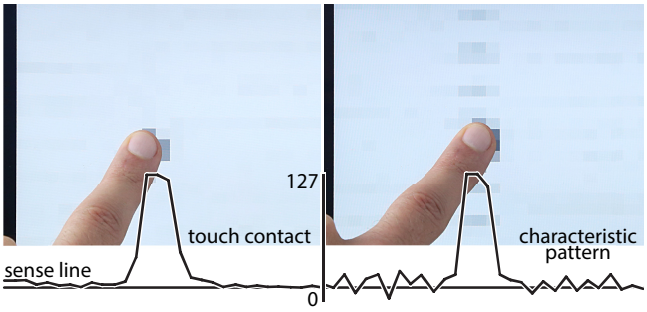

Figure 5: The debug interface of Atmel's touch chip provides a signed 8-bit capacitive image, (a) here on a Surface 2 Pro with a single touch. (b) When Bioamp modulates a signal onto the body, a characteristic pattern appears along the sense line. (Note how some values drop below zero during modulation.)

Figure 5b shows the image when Bioamp modulates a signal onto the user's body. In addition to the obvious spike for the touch itself, a series of dots along the same sense line appears, which result from Bioamp's modulation. The plot of values along the sense line reveals a zigzag pattern whose values partially drop below zero. Our software on the touch device detects this pattern by comparing the variance of values along a sense line where a touch is present with those of neighboring lines with no present touch contact. We determine a modulated signal to be present if the variance is larger than the mean variance of sense lines without a touch plus two standard deviations of the values. Our approach supports multitouch interaction for two or more users as long as two different users touch separate sense lines; otherwise, signals may not be decoded accurately. In practice, however, the pitch of 4 mm between sense lines is small enough, such that users' fingers are on separate lines. Our implementation supports full multitouch interaction for a single user in all cases.

SIGNAL ENCODING AND DECODING THROUGH SKIN

Bioamp encodes values using on-off keying to send a message to the touchscreen through the user's body. The limitation to this encoding type is a direct result of our limited access to raw sensor values on the unmodified commodity touch device. We did not observe systematic effects of more favorable encoding schemes on the debug image in our tests.

The combination of on-off keying and the debug interface's update rate of ~29 Hz throttles the throughput to a reliable transmission rate of 12 bps. Higher rates are error prone due to the varying update rate of the debug interface. On top of this, both devices communicate on a simplex channel, which is why we need to use a low-enough frame rate to guarantee reliable transmission. We test more rates in our evaluation.

Our limited transmission rate through the body means that we cannot send biometric values in reasonable time. As a workaround, Bioamp transfers biometric features to the touch device over Bluetooth. The touch device assigns each Bioamp an ID in response, which Bioamp then repeatedly modulates onto the body. Upon decoding modulated codes, the touch device looks up the corresponding biometric features. This workaround thus supports simultaneous users.

Since Bioamp and touchscreen communicate over a simplex channel, we need to send codes repeatedly and require a separator between codes for edge alignment. To support touches with reasonably short durations, we chose to encode up to 7 codes as 1, 11, 101, 111, 1011, 1101, 1111, separated by a double zero bit. This variable code length affects transmission speed: '1' transmits in 250 ms (i.e., a 1 and a 0 before and after as separator), '11' takes 333 ms to transmit (0110).

Note that this workaround maintains the directness of our approach, associating touch contacts directly with the user's biometric features. The low through-body transmission rate is a result of our decision to implement biometric touch sensing on today's commodity touchscreens with unmodified hardware. Full control over the touch chip would allow us to detect Bioamp's modulations at much higher rates before any filtering occurs, similar to Atmel's active stylus, which communicates with the chip through sense lines, transmitting 64 bit at 100 Hz. Ultimately, we envision Bioamp to modulate biometric values directly onto the user's body.

BIOMETRIC IDENTIFICATION ON THE TOUCH DEVICE

Each touch device maintains a database of registered users with an associated biometric feature vector. Initially, a touch device obtains a user's features from Bioamp's recordings to associate them with the user accounts on the device.

To identify a user, the touch device runs a support vector machine that classifies feature vectors (SMO with logistic models, polynomial kernel, *complexity* = 2.0, *exponent* = 1.0, implemented by Weka) and fits logistic regression models to the output (1-vs-1 with pairwise coupling). The use of these models provides a confidence value for each classification. We use this to strictly reject classifications with less than 95% confidence to avoid false positive matches.

INTEGRATING PER-TOUCH AUTHENTICATION INTO THE OS

As shown in Figure 6, we introduce a software layer on the touch device that intercepts all touch events, identifies the

user, authenticates the input event, and reinjects the touch event upon success. If the user is not permitted to invoke the respective operation, the system either shows an error message or switches the context of the app to the right user, such as by closing and reopening the Mail application.

The native component of our system uses Windows 8's Accessibility API to intercept and reinject touch events. The managed part inspects the user interface through the Automation API and configures or restarts processes according to the authenticated user. Our software layer is transparent and requires no change of apps running on the system.

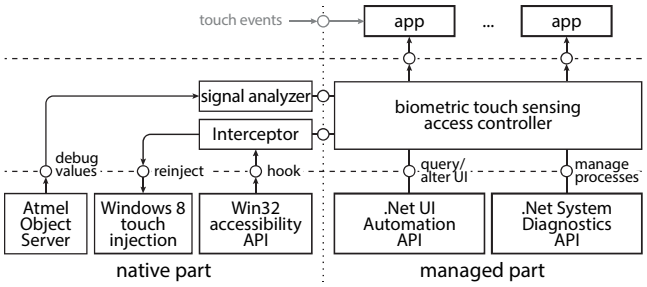

Figure 6: Our software layer on the touch device intercepts all touch events, authenticates the user for the event, and reinjects the touch upon success or displays an error otherwise.

The OS layer currently handles all authentication, such that existing single-user apps need no modification; the OS handles user switching as shown below. Since some multi-user apps may benefit from managing user access themselves, future implementations will handle access control on the OS level by default, but allow apps to request manual control.

USE CASES FOR REGULAR TOUCH INTERACTION

Touchscreens that implement biometric touch sensing integrate continuous authentication seamlessly into all interaction. Below, we demonstrate how this changes interaction in everyday scenarios on a shared touch device at the example of Windows 8. All scenarios are implemented as shown using the integration into Windows, the biometric identification, and the signal detection described above. (Please refer to the video figure for a demonstration, for which we recorded the impedance profile of both wrists for both actors.)

Access permissions for applications and personal data

User authentication is necessary when apps try to access personal data that has not been publicly shared. We distinguish three categories of access permissions: Apps that require no access to personal data and thus no authentication, apps that work without, but improve with access to personal data, and apps that do not function without such access.

1) Apps without personal data access

Apps in this category maintain no local state and do not permanently store or access any local data. This includes apps, such as a calculator or any kind of browsing activity that purges cookies and other local data at the end of the session. Using our approach, any user can open such apps, as content is stored only temporarily and the app does not require access to data on the device, such as typical web browsing: reading news, searching, looking up the weather, or watching video.

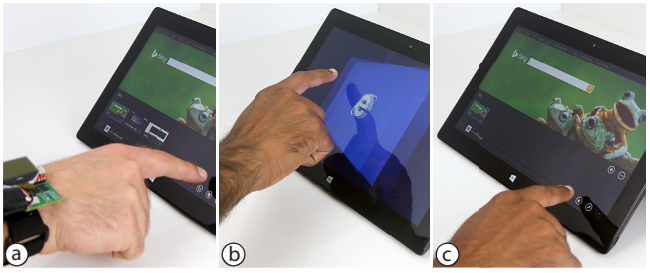

Figure 7: (a) When browsing apps maintain a user's session, access needs authentication, so other users resume their sessions and (b) unauthenticated users obtain (c) a fresh session.

2) Apps that may access personal data during runtime

Many of the apps in the first category build up a session for the user to maintain a state, which includes personalization and customizations, such as history and bookmarks in the case of a web browser, or previously visited restaurants and ratings in the case of Yelp. Therefore, when a user launches an app, the operating system starts or resumes it in the user's context as shown in Figure 7a, thereby loading tabs and cookies, for example to continue shopping in a web portal. When an unregistered or unauthenticated user starts the app, the browser needs to start a fresh session (Figure 7b&c).

Other apps allow users to browse content, but require a user's *authority* to execute specific operations. For example, everyone may browse a store, but buying apps requires payments and thus authentication as shown in Figure 8. Here, the OS starts the store with Paul's profile and shows recommendations based on his previous purchases. (a) Paul now buys a game simply by tapping the 'buy' button, which installs the game. While this interaction appears seamless, the system authenticates Paul when tapping 'buy', ensures that his account balance is sufficient, (b) charges his account and installs the game or (c) shows an error message and aborts.

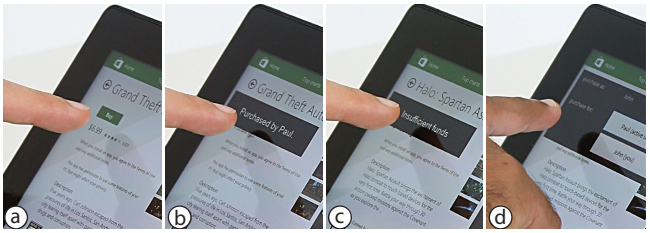

Figure 8: (a) While browsing a shop is public, a user needs to be authenticated for purchases. (b&c) Here, the tablet authenticates users for each payment, such that when (d) a different user touches 'buy', the amount is charged to their account.

The store example also presents the opportunity for a *different* user to pay for an item. When Paul has insufficient funds and John taps 'buy', the store charges John and asks him for whom the game should be installed (Figure 8d). Biometric touch sensing enables this type of transaction, because we model user authorization on a per-input event basis.

3) Apps that function solely based on personal data

Some apps primarily access personal data, such as finance, messaging and email. Users either read past items or create new items, all of which requires authentication. As shown in

Figure 1, the device will not open Mail for an unknown user, because Mail offers no functionality for unauthenticated use.

Other apps create content, but do not immediately access previously created content, such as the camera app. While any user may take a photo, biometric touch sensing ensures that the picture is stored in the photo gallery belonging to the user who pressed the trigger button as shown in our video figure.

Access control for displayed content

Though biometric touch sensing authenticates access upon input, we have less control over the displayed output. We describe how we may still protect output, including notifications from incoming events or output in foreground apps.

Non user-initiated output, such as incoming notifications

Incoming events may carry sensitive information, such as notifications and thus require access permission for reading. Unless the respective app is already in the foreground, touch devices typically pop up notifications temporarily. Some notifications are insensitive, such as the currently playing song.

To protect sensitive notifications, we indicate the *presence* of a notification and, on a shared device, show the recipient's name on the screen as shown in Figure 9. The recipient needs to touch and hold the notification to reveal the content, which vanishes when releasing the touch (similar to Snapchat).

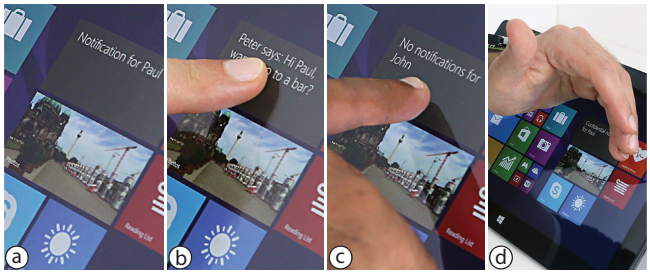

Figure 9: (a) The tablet indicates the presence of a notification for a particular user, but does not reveal the contents. To reveal the message, (b) the user needs to touch and hold the item, (c) after which the message disappears. (d) To protect against shoulder surfers, we detect shield poses to cover messages.

To protect sensitive content from potential shoulder surfers, we support the use of a shield pose (Figure 9d). Only when the user covers the notification area will the tablet display the content. The detection of this hand pose is straightforward by analyzing the shape of touches in the capacitive image (similar to ShieldPIN on a tabletop system [18]).

Protecting output during regular interaction with apps

In some cases, the screen content of an entire app needs to be protected, for example a banking application that should not remain in the foreground when a user leaves. In this case, we require the user to hold a constant touch on the screen, for example by grabbing the bezel of the device and holding down a thumb. When no touch is in contact, the system will close the app, either right away or after an adjustable timeout.

If no timeout is set, biometric touch sensing enables the system to hide contents for other users. As shown in Figure 10a, Paul is interacting with Mail. (b) John comes in, touches the screen, and the system authenticates John and (c&d) starts his email instead. If the user is unknown, the device will instead return to the home screen as shown in the video figure.

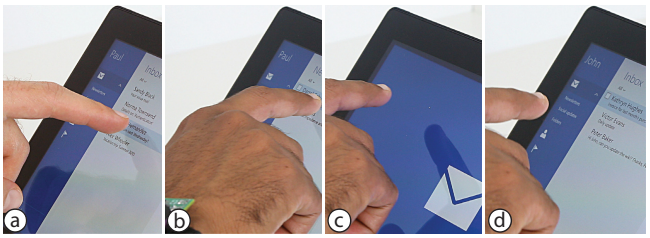

Figure 10: Biometric touch sensing enables devices to hide contents quickly. When Paul is logged in and John starts interacting, the device logs Paul out and starts John's Mail app.

Note that unlike browsing the Windows store, interaction with the Mail app constantly accesses personal information and thus requires authentication. This is why the system "logs out" the active user to protect personal data, which is unnecessary in the browsing context of the store example.

Sharing access permissions temporarily

Finally, biometric touch sensing enables sharing permissions temporarily. When multiple users interact with the tablet, the behavior described above is undesirable. Our system thus observes touches over time: If one user interacts with sensitive content and another user touches this content either simultaneously or shortly thereafter, the second user may access this item temporarily. Figure 11 shows this interaction in the context of private photos. The temporary sharing is inspired by real life situations, where one person hands over an item to another. We have found that a 5 s timeout is sufficient, which also suits situations where one user asks another to look up information on their device, for example when driving.

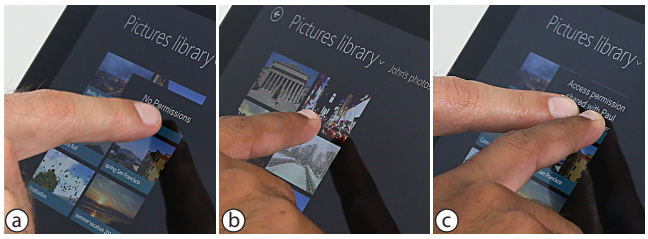

Figure 11: Temporarily sharing access permissions. (a) Paul has no access to (b) John's album, (c) but John temporarily shares access permissions by simultaneously touching the item.

Temporary access sharing works on an object-hierarchy basis. If a user shares access to an item in a directory, the second user may navigate this item and all its subdirectories, but not parent directories. Once the second user closes the item, that user loses the temporarily granted access permissions.

We also support temporary access sharing for content that is even more sensitive. Here, we require the owner and another user to initially touch the item simultaneously. The owner may then let go and the item remains visible as long as the second user maintains contact (similar to our notifications).

Explicit logins for customized sessions and registration

While biometric touch sensing allows devices to present user interfaces upfront, such as the home screen, it limits their

personalization. For example, all users share a background image and live tiles cannot show friends' pictures or data.

We account for this by supporting explicit desktops as shown in Figure 12a. The device initially displays the start screen with apps that are available to all users and a login button. Upon touch, the system identifies the user and (b) logs them into their desktop, showing their pinned apps and images.

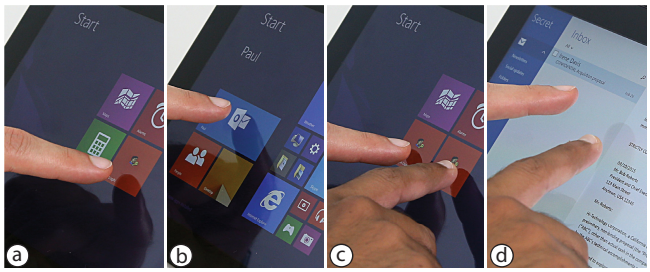

Figure 12: (a) Tapping the login button takes the user to their (b) personalized desktop. (c) For confidential data that requires the presence of two users, both users need to touch the login button simultaneously to (d) open a confidential inbox.

Multi-user logins for additional security (four-eyes principle)
Biometric touch sensing also supports logins that require more than one person to be present at a time. As shown in Figure 12c, two users need to touch a single login tile simultaneously to unlock and open sensitive emails (d).

TECHNICAL EVALUATION

We evaluated two aspects of our system. First, we verified that Bioamp reliably identifies users for our envisioned scenario. Second, we evaluated through-body transfer rates with a commodity touchscreen and direct measurements.

1) Evaluation of user identification accuracy with Bioamp

We recruited 10 participants (ages 22–45, 2 female) for a controlled lab evaluation of our Bioamp prototype as shown in Figure 3. Bioamp extracted all features and we processed all features on the Surface 2 Pro device as described above.

Procedure: The experimenter told participants to put on Bioamp like a watch and tighten it around their wrist. Bioamp then started capturing biometric values. After a trial, participants fully released the prototype and placed it on a table to account for moving electrodes during use. Altogether, participants performed five repetitions during the evaluation.

Data processing: We performed a 5-fold cross validation to evaluate identification precision and rejection of unknown users. During each fold, we trained the classifier with four trials of eight participants' feature vectors and then classified users using the remaining 18 trials. Leaving out two feature vectors during each fold simulated unregistered users.

Results: The classifier produced 38 true positives (>.95 confidence), one false negative (.57) and one true negative (.31). The latter two were correctly rejected because of the low confidence score. For simulating unknown users, the classifier produced 50 true negatives (mean confidence = .27, largest = .41). Overall, the classifier produced no false positive matches. The speed of classification was near instant thanks to the previously trained support vector machine classifier.

2) Evaluation of transmission rates through the body

This evaluation tested the success of signal transmission through the experimenter's body as a simplex channel with varying transmission rates between Bioamp and the tablet.

Apparatus: Bioamp repeatedly modulated a 20-bit signal onto the body, consisting of ten 1 bits followed by ten 0 bits. Our software running on the tablet decoded the signal using the touch chip's debug interface (ATMEL interface). The pattern was static for edge alignment to calculate bit error rates. Since the low update rates of the debug interface limit transmission rates a priori, we added the SIMULATED TOUCH CHIP interface: an ECG electrode attached to the touchscreen, using gel between the electrode and the surface to approximate a direct connection to the sense line as shown in Figure 13a. We connected the electrode to the tablet's audio port for sampling at 4 kHz, which simulated the use of a fast touch sensor (e.g., [20]). The tablet decoded modulated signals as described above and used the variance of audio measurements when no touch was present as a baseline for comparison and normalization of recorded values during transmission.

Procedure: To evaluate the impact of the environment, we tested the four types of surfaces for the tablet: a metal, wood, and plastic table, and the user's lap when sitting on a couch as shown in Figure 13. For the table scenarios, the experimenter rested their arms on the table in one condition, while lifting them above the surface in another, resulting in 3 tables × 2 arm poses + lap = 7 conditions. During each trial, the experimenter touched the screen for ~5 s, lifted the finger for ~10 s, and repeated the trial 5 times overall. During each block, Bioamp was configured to transmit the value pattern at one of the following rates: 5, 10, 12, 14, 15, 30, 100, 1k, 2k Hz. We measured and detected the modulated signal on both interfaces and computed the bit error rate for each transmission rate by averaging the rates of all trials.

Figure 13: Conditions during the through-body evaluation. An ECG electrode simulated a direct connection to the sense line.

Results: As shown in Figure 14, ATMEL's interface provides no bit errors up until a transmission rate of 12 Hz for all tested conditions. This indicates that our heuristic for computing the presence of a modulated signal is stable for these frame rates, but fails to detect modulations accurately at faster transmission rates. Since Bioamp communicates with the touchscreen via a simplex channel without any feedback, we require a bit error rate of zero for proper use.

The SIMULATED TOUCH CHIP showed no bit errors up to 30 Hz in all conditions. Only when the experimenter rested their

arms on the metal table did the bit error rate increase to 0.01 at 100 Hz. Figure 14 also shows that bit error rates for the SIMULATED TOUCH CHIP increase to ~0.05 above 1 kHz. Upon closer inspection, we found that the spread of variances of signals is not completely disjoint at these speeds anymore, causing our detection to produce less than perfect results.

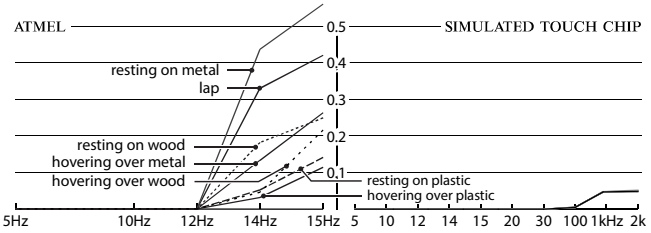

Figure 14: Bit error rates. (left) ATMEL's debug interface affords transmission rates of 12 Hz with no error. (right) The SIMULATED TOUCH CHIP return larger bit error rates at 1 kHz.

Discussion

The results of our evaluation show that the design and algorithm of Bioamp achieves comparable results to previous work on wrist-based bioimpedance (e.g., [8]). Our sample size is a good approximation of a large household, which validates our scenario. Importantly, no trial was misrecognized (no false positives), even though participants took off the band between trials. Our simulation of unknown users shows that the extracted features afford reliable rejection of unauthorized access. Of course, assessing the true security of wrist bioimpedance requires a much larger sample size.

More importantly, our evaluation confirms the viability of our proof-of-concept implementation using a transmission channel of the touchscreen that was *never designed* for this purpose. While the tablet's debug interface reliably supports transmission rates up to 12 bps, we showed that full control over the touch chip affords much higher transmission speeds. While we found that our heuristic for detecting a signal occasionally detects bits erroneously, we assume that a stronger output voltage *Vpp* might stabilize the measured signal variances and set them apart more clearly again.

In our evaluation, Bioamp transmitted 2 KB/s to the touchscreen at a sampling rate of 4 kHz using on-off keying. To test the potential of our implementation at the upper bound of our platform, we implemented 4-level amplitude-shift keying at the maximum sampling rate of 48 kHz and obtained 96 KB/s throughput with a bit error rate of 0.18.

Our results show that placing the touch device on a conductive object, such as the metal table, has a limited impact. The bit error rates of the ATMEL interface may have resulted from the irregular update rates of the debug interface or from Atmel's compensation for interfering frequencies, causing the chip switches its drive frequency. The metal table affected the bit error rates most in the resting configuration, likely because the experimenter's arms (which carried the signal) then coupled to the ground of the touch device, thus lowering the measured differential between the screen and its own ground.

Our evaluation demonstrates the potential of through-body communication with touchscreens when detecting signals on the chip *before* performing any signal filtering, similar to Atmel's maXStylus. We also hope to benefit from future host-side processing on touch devices to process measurements at full resolution and speed on the CPU before any filtering. Alternatively, a touchscreen with faster sampling rates (e.g., 4 kHz [20]) would improve the practicality of our implementation and support decoding signals on fast taps. Future work also needs to determine if a biometric vector with fewer than the 96 values we used suffices for reliable user identification, in which case transmissions would be shorter and faster. We also tested signal modulation using a Nexus 5, which showed the same characteristic pattern on the sense line, but further evaluation needs to test if this extends to all manufacturers.

LIMITATIONS AND FUTURE WORK

The perhaps largest limitation of our current implementation is the requirement to wear a biometric sensor. While we built Bioamp in the spirit of emerging biometric wearables, no commercial product currently modulates signals onto the user's body for *direct* communication between two devices.

The design of biometric touch sensing we chose is akin to fingerprint scanning, i.e., sensing biometric signals upon touch, along with security concerns, such as replay attacks. Unlike fingerprints, however, Bioamp is an active component and can support certificates and signatures to prevent replay attacks. We currently examine the use of secure sketches and fuzzy extractors in Bioamp, which are designed to prevent revealing biometrics to prevent attackers from reconstructing and replaying them during attacks [10].

Finally, touch devices need to prepare for a malfunctioning biometric converter. In this case, they need to resort to regular password entry to authenticate access. Future work therefore includes assessing the general usability of our approach.

CONCLUSIONS

We presented biometric touch sensing, an approach to sensing input on touch devices that associates the user's identity with *every single* touch event. We enable commodity touch devices to identify users biometrically from the touch contact users make with the screen during interaction. We outlined the implications of biometric touch sensing for touch interaction with current operating systems in the context of access permissions, protecting displayed content, and temporary permission sharing, and demonstrated our implementation on Windows 8.1 on an off-the-shelf Microsoft Surface 2 Pro tablet. The key to implementing biometric touch sensing on today's touch devices is Bioamp, a watch prototype that complements touch devices, senses biometric features, and conceptually converts them into an electric signal that today's touch devices can detect upon touch. Bioamp is *not* an identification token (although it could act as one)—it only senses biometric features and forwards them to the touch device.

We believe that the concept of biometric touch sensing generalizes beyond our specific implementation. While we chose bioimpedance as *one* possible biometric feature and a wrist-worn prototype to explore our approach on today's

touch devices, we expect future touchscreens to biometrically identify users from each touch (e.g., fingerprints on in-cell screens [6]) and implement biometric touch sensing.

In addition, the use of a separate device for biometric sensing that complements interaction generalizes beyond Bioamp. As shown in Figure 16, such a device may (a) record features of the user's iris through a head-mounted camera, detect biometric features, such as palm prints and fingerprints on (b) a keyboard, a mouse or (c) a pen. In all these scenarios, our concept of continuous authentication on a per-interaction level applies, either by using the body as a conduit to transmit the features through an intermediary device that makes direct contact with the touchscreen in the case of the pen, or a separate input controller, such as a mouse or keyboard. Biometric pens and separate input controllers are particularly interesting in the context of interaction with large displays.

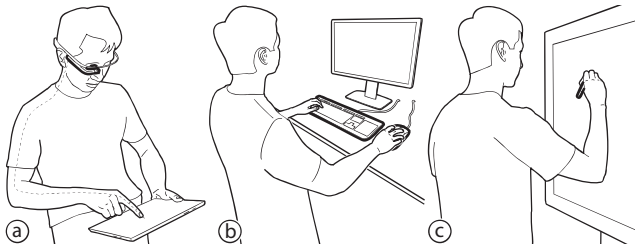

Figure 16: Bioamp's concept generalizes to any device that scans biometric features and forwards them to a device, such as (a) a head-mounted device that scans the iris and modulates the features onto the body, (b) a keyboard or a mouse that scans fingerprints and forwards them to the computer, or (c) a pen that scans fingerprints an sends them to the touchscreen. All these devices support per-interaction authentication.

ACKNOWLEDGEMENTS

We thank Flavio Ribeiro and Chad Solomon for establishing our collaboration with Atmel. We thank Edward Wang for repeated discussions and Tobias Grosse-Puppendahl for feedback, as well as Cheng Xu for early input and feedback.

REFERENCES

1. Amitay, D. Most Common iPhone Passcodes. http://danielamitay.com/blog/
2. Baldus, H., Corroy, S., Fazzi, A., Klabunde, K., Schenk, T. Human-centric connectivity enabled by body-coupled communications. *IEEE Communications Magazine* (47) 6.
3. Barrett, G. and Omote, R. Projected-Capacitive Touch Technology. *Information Display*. (26) 3, 2010. 16–21.
4. Ben-Asher, N., Kirschnick, N., Sieger, H., Meyer, J., Ben-Oved, A., Möller, S. On the need for different security methods on mobile phones. *Proc. MobileHCI '11*.
5. Bionym Nymi wristband for ECG-based authentication. http://www.getnymi.com/
6. Brown, C.J., Kato, H., Maeda, K. and Hadwen, B. A Continuous-Grain Silicon-System LCD With Optical Input Function. *Solid-State Circuits*, 42(12), 2007.
7. Cornelius, C. Usable security for wireless body-area networks. *PhD Dissertation*, Dartmouth College, 2013.
8. Cornelius, C., Peterson, R., Skinner, J., Halter, R., Kotz, D. A Wearable System That Knows Who Wears It. *MobiSys '14*.
9. Dietz, P. and Leigh, D. DiamondTouch: a Multi-User Touch Technology. *Proc. UIST '01*, 219–226.
10. Dodis, Y, Reyzin, L., Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Proc. Eurocrypt '04*, 523–540.
11. Grimnes, S., Martinsen, Ø.G. Bioimpedance and Bioelectricity Basics, Academic Press, London, 2008.
12. Grosse-Puppendahl, T., Herber, S., Wimmer, R., Englert, F., Beck, S., Wilmsdorff, J., Wichert, R., Kuijper, A. Capacitive Near-Field Communication for Ubiquitous Interaction and Perception. *Proc. Ubicomp '14*.
13. Harrison, C., Sato, M., Poupyrev, I. Capacitive Fingerprinting: Exploring User Differentiation by Sensing Electrical Properties of the Human Body. *Proc. UIST '12*.
14. Holz, C. and Baudisch, P. Fiberio: A Touchscreen that Senses Fingerprints. *Proc. UIST '13*, 41–50.
15. Holz, C. and Baudisch, P. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. *Proc. CHI '10*, 581–590.
16. Holz, C., Buthpitiya, S., Knaust, M. Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts. *Proc. CHI '15*.
17. Israel, S. A., Irvine, J. M., Cheng, A., Wiederhold, M. D., Wiederhold, B. K. ECG to identify individuals. *Journal of Pattern Recognition*, 2005.
18. Kim, D., Dunphy, P., Briggs, P., Hook, J., Nicholson, J., Nicholson, J. and Olivier, P. Multi-Touch Authentication on Tabletops. *Proc. CHI '10*, 1093–1102.
19. Kurkovsky, S., Syta, E. Digital natives and mobile phones: A survey of practices and attitudes about privacy and security. *Proc. ISTAS '10,* 441–449.
20. Leigh, D., Forlines, C., Jota, R., Sanders, S., & Wigdor, D. High rate, low-latency multi-touch sensing with simultaneous orthogonal multiplexing. *Proc. UIST '14*, 355–364.
21. Matsushita, N., Tajima, S., Ayatsuka, Y., Rekimoto, J. Wearable key: device for personalizing nearby environment. *Proc. ISWC '00*, 119–126.
22. Richter, S., Holz, C. and Baudisch, P. Bootstrapper: Recognizing Tabletop Users by their Shoes. *Proc. CHI '12*.
23. Sato, M., Poupyrev, I., Harrison, C. Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects. *Proc. CHI '12*, 483–492.
24. Schmidt, D., Chong, M., and Gellersen, H. HandsDown: Hand-contour-based User Identification for Interactive Surfaces. *Proc. NordiCHI '10*, 432–441.
25. Suoranta, S., Tontti, A., Ruuskanen, J., Aura, T. Logout in single sign-on systems. *Policies and Research in Identity Management*, 147–160. Springer, 2014.
26. Vu, T., Baid, A., Gao, S., Gruteser, M., Howard, R., Lindqvist, J., Spasojevic, P., Walling, J. Distinguishing users with capacitive touch communication. *Proc. Mobicom '12*.
27. Wang, Y., Agrafioti, F., Hatzinakos, D., Plataniotis, K.N. Analysis of human electrocardiogram for biometric recognition. *EURASIP Journal,* 2008.
28. Zimmerman, T. Personal area networks: near-field intrabody communication. *IBM Journal 35*(3.4), 1996, 609–617.