Smartphones

Editor: Roy Want ■ Google ■ roywant@acm.org

Personal Touch-Identification Tokens

Tam Vu and Marco Gruteser

EDITOR'S INTRO

One of the goals of pervasive computing is the automatic personalization of computer infrastructure to create a customized user experience that's better suited to the task at hand. The ideas presented in this article are a significant step toward that goal. The authors provide a convenient and accurate mechanism to convey a unique identifier using no more than a signet ring pressed against the capacitive touchscreen of a computer. It brings to mind the "secret decoder ring" used by the fictional character, Dick Tracy—but this time it's for real. — *Roy Want*

In this emerging era of pervasive computing, we interact with and rapidly switch among a diverse set of digital devices. We tend to transition from a smartphone to a notebook when arriving in the office, and when we return home, we often switch to a tablet. In between, we might use large wall-mounted displays, car navigation systems, self-checkout kiosks at retail stores, and home security or smart home controls. Many of these devices—even if only used in the home—have multiple users. Over the next decade, this range of devices will likely increase, and our time with any single device will grow shorter.

We argue that it's time for touch-based personal tokens that let devices unobtrusively identify who is interacting with the device at any given time. This would let devices tailor services to users and apply authentication to control access to sensitive information and online services. Consider, for example, that a child reportedly spent more than $1,000 on in-app purchases while playing games on his mother's iPhone.[1] Devices that know their users could limit such spending from unauthorized users and serve age-appropriate content and media. Our approach is to use a wearable personal token to communicate an identification code through touch.

USER IDENTIFICATION CHALLENGES

Current user-identification techniques are often based on logins and PIN codes or passwords. The overhead of logging in was more acceptable when we used to sit down for long sessions of work in front of a PC, but it now has become a nuisance—to the point where many users forgo any authentication on their mobile devices. Biometric techniques, such as fingerprint readers or face recognition, are more resource intensive and face their own usability challenges. Fingerprint readers require space on a device, add cost, and can be difficult to use in low-humidity environments. Face recognition has higher processing requirements and can be difficult in low-lighting conditions.

Our goal has therefore been to find a less obtrusive way to identify users during their regular interactions with pervasive devices. Capacitive touch sensors have emerged as a dominant user interface technology for mobile and pervasive devices. Touch sensors reside in hundreds of millions of smartphones and tablets as well as in ATM machines, car dashboard displays, and even home appliances such as televisions, microwaves, and refrigerators.

Given that touch is the predominant way of navigating and interacting with today's computer-embedded devices, our approach seeks to identify who touches a device, which can be more accurate (or secure) than user-proximity sensing with short-range radios such as Bluetooth or NFC. When there are several potential users near a device, one of the most meaningful ways to identify who's really interacting with the device is to identify who is touching it.

A TOUCH-IDENTIFICATION TOKEN

We've explored how to design a wearable personal token that exploits the pervasive capacitive touchscreen and touchpad input devices as receivers for an identification code or password. Our current token is in the form of a ring, but other devices, such as wristbands or watches, are also possible. The token transmits electrical signals when brought in contact with a device's touchscreen or when the user's finger touches the screen. In this latter case, the signal is transmitted through the human skin. The transmitted electrical signals from the token essentially spoofs the

Published by the IEEE CS ■ 1536-1268/13/$31.00 © 2013 IEEE

device's touchscreen, mimicking the signals from up and down movements of a finger, while the finger (or token) itself remains stationary on the device.[2]

To understand this, consider how a capacitive touchscreen works.[3] It comprises an array of conducting electrodes behind a transparent insulating glass layer, which form a capacitor with the human finger when the user touches the screen. The finger is in turn capacitively coupled through the human body (and sometimes through the ground) to the device's case. A touch can therefore be detected by driving the electrodes with an AC signal to repeatedly charge and discharge the capacitor and measure the change in capacitance through the change in charge voltage. Where on the electrode array this change occurs reveals the location of the touch.

Owing to the small capacitance involved, a charge integration circuit might be necessary. When this change in voltage exceeds a certain threshold and meets several other filtering criteria, the screen's firmware reports the presence of a new touch event to the device's operating system, which then traverses up the software stack to the application level. As our wearable hardware token comes in contact with the screen, its own AC signal charges and discharges the capacitor formed with the screen's electrodes, leading to repetitive but irregular touch events captured by the touchscreen controller.

Our system then exploits this ability to generate touch events through electrical signals to modulate these events and communicate a short identification code between the token and device. As Figure 1 shows, the system can be viewed as a classical communication system with a

- transmitter (a hardware token);
- communication channel (the hardware and firmware of the device's touchscreen, the software stack, and its operating system); and
- receiver (the software component demodulating the sequence of touch events to receive the originally transmitted bit sequence).

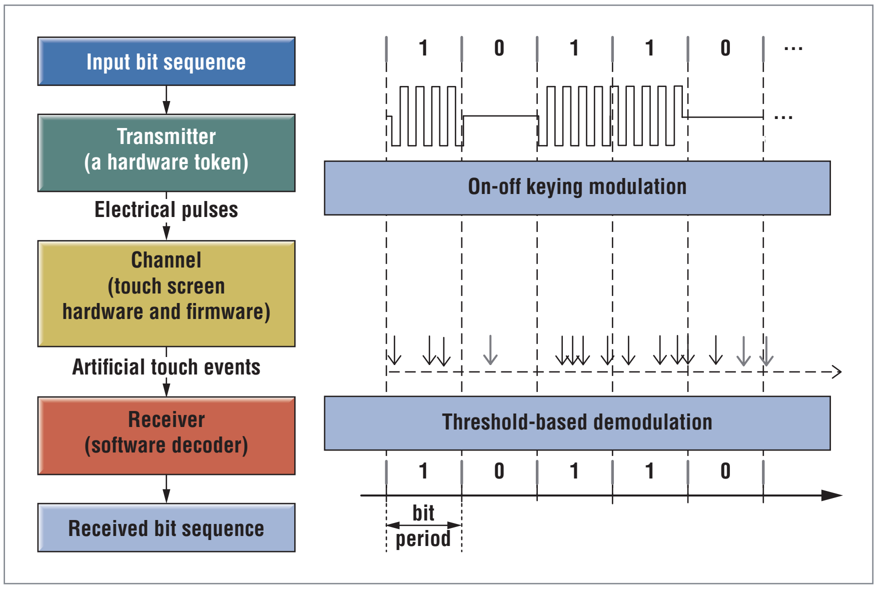


Figure 1. Modeling the system as a classical communication system that includes a transmitter, communication channel, and receiver.

The transmitter generates electrical pulses that are modulated to carry the input bit sequence using *on-off keying modulation*—that is, a *one* bit is represented by turning on the electrical signal, and a *zero* bit is represented by switching that signal off. As a result, the pattern of the sequence of touch events generated follows the original bit sequence. These events thus can be used to reconstruct the originally transmitted bit sequence on the receiver side—otherwise unknown to the screen. The software component, acting as the receiver, counts the number of touch events in each bit period to determine which bit was transmitted. It receives a one bit if the number of events appearing in that bit period is greater than a certain threshold; otherwise, it receives a zero bit.

Decoding Challenges

Because touchscreens weren't designed for communication purposes, and we only had operating-system level access to the touchscreen hardware, this approach posed several challenges. The touchscreen responded differently to the same input, generating different received bit patterns. For example, the number of touch events registered when a one bit is sent after a long sequence of zeros is lower than that of a one bit that comes after a long sequence of ones. The variable delay between the transmission of a bit and its reception at the receiver makes it even more challenging to demodulate it to retrieve the original bit sequence.

In addition, the channel adds an unknown delay between the receiver and the transmitter due to the unknown processing delay of the device, which depends on the device's workload. As a result, traditional demodulation techniques aren't applicable. Instead, we devised a two-step process to decode the originally transmitted data: first calibration, then correlation. During offline calibration, the software component selects a threshold, which is used for bit detection during the correlation phase.

The offline calibration step computes the expected number of touch events when a one bit or zero bit are transmitted. To determine this number, the step must be performed once per device, per data rate, during initialization. The hardware token repeatedly transmits a bit sequence that's known to the receiver.

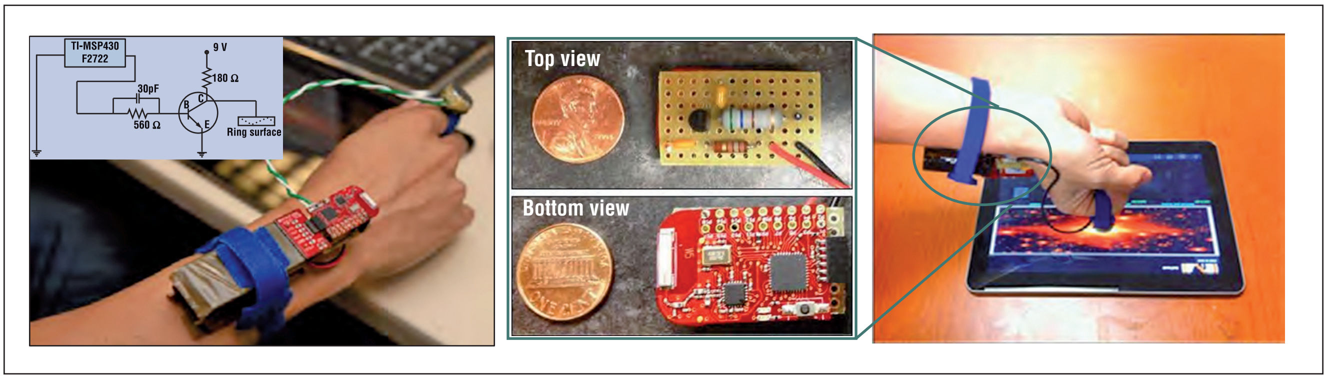


Figure 2. A wearable battery-powered hardware token that transmits a short message from the ring to a touchscreen-enabled device.

Upon receiving a sequence of touch events corresponding to that known bit sequence, the receiver software synchronizes with the transmitter, then counts the number of touch events in each one and zero bit to calculate the average number of expected events in each.

During the actual communication, assuming all possible messages are known, the software decoder then computes the correlation between the touch event sequence and all possible messages using the expected number of events. The message that yields the highest correlation value is selected as the originally transmitted bit sequence.

Evaluation

Our initial performance evaluation shows that the communication channel can deliver a short bit sequence at a low data rate. We connected a function generator's output—that is, modulated electrical signals with bit sequences of different lengths and data rates—to an Android tablet's touchscreen. The receiver correctly retrieved the originally transmitted 2-to-3-bit-long sequences more than 99 percent of the time, with a data rate of 4 bits per second. As the data rate or bit sequence's length increased, the detection rate gradually decreased.

In another set of experiments in which the output of the function generator was connected to a human finger swiping across touchscreen's surface, we examined whether the receiver could detect the presence of the electrical signals without the hardware token directly contacting the screen. Note that, in this case, contact times longer than a typical touch are required, but such longer contact times can be expected during swiping motions.

Our results showed that the detection rate increased with the duration of the swipe—at first sharply, from approximately 68 percent for 300 millisecond swipes to approximately 92 percent for 500 ms swipes, but then it gradually approached 100 percent as the swipe duration increases to 1,400 ms.

Encouraged by the function generator's results, we prototyped a battery-powered wearable token. This current token achieved detection rates of 82 to 90 percent when transmitting 2-to-5-bit long sequences at the data rate of 4–5 bits/s. As Figure 2 shows, the token, in a form of a wearable ring, consists of a custom-built amplifier and a programmable microcontroller with a flash memory for holding the bit sequence. The overall detection rate could probably be improved through better circuit design, ring surface engineering, and retransmission of the transmitted bit sequence.

APPLICATIONS

Our work thus far demonstrates the feasibility of communicating short codes from personal tokens to touch receivers. This technique could be directly applied to parental-control applications, multiuser games, and weak authentication for mobile devices. Further improvement in the transmission rate, reliability, and security would result in immense opportunities to create a unified user identification and authentication scheme around personal tokens rather than passwords (see Figure 3).

Akin to SIM cards today, which are identification tokens for devices in a cellular network, wearable tokens could provide identification for people.[4] Conceptually, SIM cards were an adequate solution for people accessing a network through a single device. However, with access to diverse devices—such as smartphones, laptops, tablets, and cars, which might be shared among multiple users—it's becoming more important to understand which user is interacting with the device at any given time.

In addition, with future shared data plans (shared across devices), data usage from any device could be charged to a user-specific (rather than device-specific) account. This type of billing model could be realized with our proposed techniques, using the signet ring as a separate identification token—a portable SIM-ring—worn by users.

Beyond networking, the ring could also allow payment functions and replace credit cards (becoming a *credit ring*). It could authenticate monetary transactions on mobile phones and ATM machines and could even be used to access a smart home, unlocking the door for authorized access and loading user-specific preferences on certain devices, such as the entertainment system or home appliances.

Fully realizing these applications will likely require further refinement of the touch communication techniques. Although communication is feasible, even with an off-the-shelf touchscreen system (albeit at very low bit rates), there appears to be ample opportunity to increase data rates by gaining access to lower-level measurements via the touchscreen microcontroller firmware. If transmission through the skin isn't desired, other alternatives exist that have simpler design options and require less energy.

One approach would be to vary the effective capacitance between the ring and screen by inserting another capacitor at this point, whose area or thickness could be modulated. Using a larger form factor ring with a surface that creates multiple contact points with the screen (exploiting multitouch capabilities) could further improve the data rate.

Integrating identification and authentication functions into a wearable item, such as a ring or wristband, should reduce the probability of losing the device (compared to smartphones) and thus reduce the security risks related to unauthorized use. In applications with higher security requirements (where theft and loss still present a serious concern), these tokens could be one part in a multifactor authentication system. Or perhaps the ring could integrate biometric signature techniques[5] with the token, activating its transmission capability only when the token recognizes the owner's signature.

Furthermore, the current design could be enhanced with a feedback channel using a photodetector.[6] The ring could receive information from the mobile device through this visual channel, where the device encodes the information in the pixel intensities of the screen location where the token is touched. This would enable a challenge-response protocol, which could greatly enhance the security of an authentication system.

Such improvements should enable these tokens to become a pervasive user identification technology that replaces cumbersome logins with a simple touch.

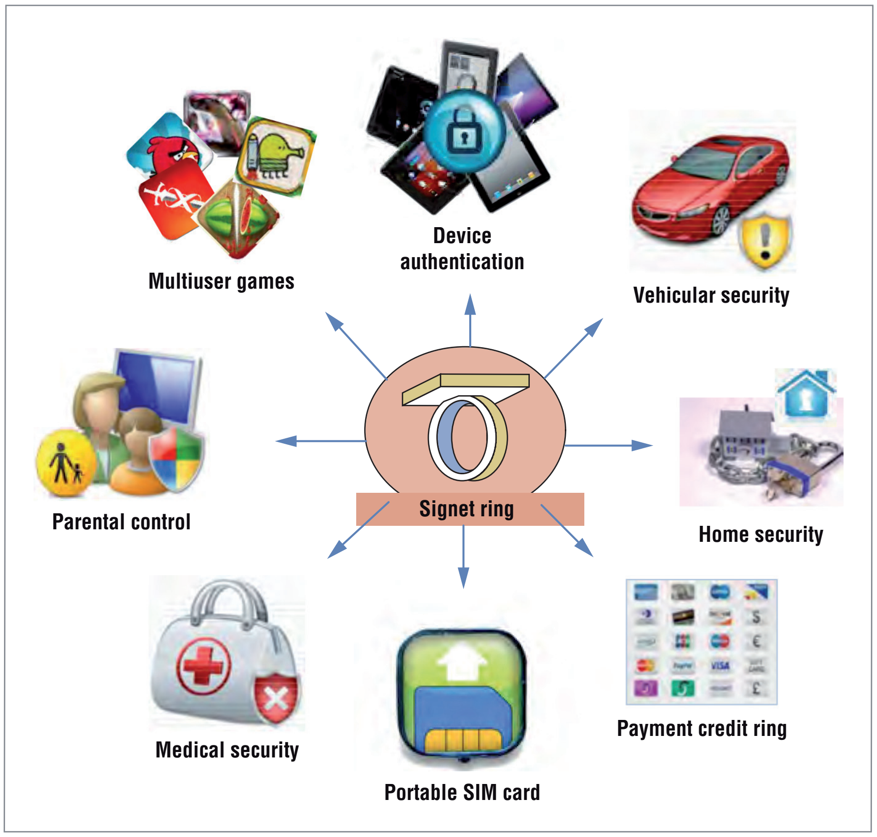


Figure 3. Potential capacitive-touch-communication applications.

ACKNOWLEDGMENT

We thank Akash Baid, Simon Gao, Richard Howard, Janne Lindqvist, Predrag Spasojevic, Jeffrey Walling, and Ivan Seskar for their contributions toward developing personal touch tokens.

REFERENCES

1. C. Foresman, "Apple Facing Class-Action Lawsuit Over Kids' In-App Purchases," *Ars Technica*, 15 Apr. 2011; http://arstechnica.com/apple/2011/04/apple-facing-class-action-lawsuit-over-kids-in-app-purchases.

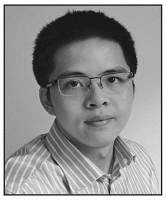
2. T. Vu et al., "Distinguishing Users with Capacitive Touch Communication," *Proc. 18th Ann. Int'l Conf. Mobile Computing and Networking* (MobiCom 12), ACM, 2012, pp. 197–208.

3. W. Westerman and J.G. Elias, *Capacitive Sensing Arrangement*, US patent 2006/0232,567,2006.

4. C. Newton, "Google's Password Proposal: One Ring to Rule Them All," *Cnet News*, 18 Jan. 2013; http://news.cnet.com/8301-1023_3-57564788-93/googles-password-proposal-one-ring-to-rule-them-all.

5. C. Cornelius et al., "Who Wears Me? Bioimpedance as a Passive Biometric," *Proc. 3rd USENIX Workshop on Health Security and Privacy*, 2012.

6. H. Melchior, M. Fisher, and F. Arams, "Photodetectors for Optical Communication Systems," *Proc. IEEE*, vol. 58, no. 10, 1970, pp. 1466–1486.



Tam Vu is a PhD candidate at Rutgers University. Contact him at tamvu@cs.rutgers.edu.

Marco Gruteser is an associate professor in the Wireless Information Network Laboratory (Winlab) at Rutgers University. Contact him at gruteser@winlab.rutgers.edu.