Battery-Free Identification Token for Touch Sensing Devices

Phuc Nguyen[†], Ufuk Muncuk[‡], Ashwin Ashok[*], Kaushik R Chowdhury[‡], Marco Gruteser[§], and Tam Vu[†]
[†]University of Colorado Denver, [‡]Northeastern University, [*]Carnegie Mellon University, [§]WINLAB, Rutgers University
[†]{phuc.v.nguyen, tam.vu}@ucdenver.edu, [‡]{umuncuk, krc}@ece.neu.edu, [*]ashwinashok@cmu.edu, [§]gruteser@winlab.rutgers.edu

ABSTRACT

This paper proposes the design and implementation of low–energy tokens for smart interaction with capacitive touch–enabled devices by associating the token's identity with its contact, or *touch*. The proposed token's design features two key novel technical components: (1) a through–touch–sensor low–energy communication method for token identification and (2) a touch–sensor energy harvesting technique. The communication mechanism involves the token transmitting its identity (ID) directly through the touch–sensor by artificially modifying the effective capacitance between the touch–sensor and token surfaces. This approach consumes significantly lower energy compared to traditional electrical signal modulation approaches. By enabling the token to harvest energy from touch–screen sensors or touch–surfaces the token is rendered *battery–free*. Through experimental evaluations using a prototype implementation, the proposed design is shown to achieve at least 95% identification accuracy. It is also shown to consume less energy than competitive techniques (NFC P2P and Bluetooth Low–Energy) for communicating a short ID sequence. The adoption of this technology among users is evaluated through a user study on 12 subjects.

CCS Concepts

•**Computer systems organization** → **Embedded and cyber-physical systems;** •**Hardware** → *Wireless devices; Power estimation and optimization;*

Keywords

Token Identification, Touch-screen Energy Harvesting, Touch Communication, Low Energy Token

SenSys '16, November 14–16, 2016, Stanford, CA, USA

DOI: http://dx.doi.org/10.1145/2994551.2994566

Figure 1: Examples of new applications that require the association of artifact's identity to its touch interactions.

1. INTRODUCTION

Human interaction with mobile devices through touch sensing has become the most common form of interaction ever since the inception of touch enabled mobile devices. Today, these interactions are being enriched with smart physical artifacts, or *tokens*. Yet, the possibility of advancing human–device interaction by associating identity of such tokens to the touch interactions has not been explored to the fullest. Associating identity to touch interactions enables new classes of applications on today's touch-enabled devices, as illustrated on Figure 1. Examples include wearable artifacts for user–authentication, smart tokens for multi-user gaming, or even for simultaneous multi-user collaboration for productivity. With technological advancements in 3D printing, designing and creating free–form tokens of various shapes and sizes to enable such applications have become easier than ever.

Association of identity to touch interactions requires reliable *identification* of the token on the touch sensing device. This primarily involves finding *where* exactly the token is placed on the touch–screen and *what* is the unique identity of that token. The touch–sensing firmware's API provided with today's touchscreen device operating systems (OS) al-

low for identifying the positions of any conducting material placed on the surface of the screen. However, this mechanism only allows to identify the positions where a token's conducting surface contacts the screen but not its identity.

One possible approach to token identification is to encode the identity as a unique physical pattern [8] which gets detected when the token makes contact with the touch surface. However, pattern detection will be highly error prone when the token is too small. Another approach [38] is to encode the identity (bits) as series of electrical pulses that trigger the capacitive touch sensing mechanism when the token contacts the surface. However, this mechanism requires significant amount of battery draw on the token, to generate electrical pulses with sufficient amplitude so as to be detected by the touch–sensor.

Motivated by the idea of *communicating* the token ID directly to the touch–screen device, we propose to address the token identification problem by leveraging the capacitive touch–sensing mechanism. We realize that this approach brings about two fundamental challenges: (a) designing a reliable communication link between the token and the capacitive touch sensor, and (b) minimizing energy consumption.

Low–Energy Token Identification. We address the design challenges for token identification through two key aspects of our proposed system design: (i) a novel mechanism for communicating through the touch–sensor through *capacitance variation*, and (ii) a mechanism to harvest energy from the touch surface. In the *capacitance variation* based communication mechanism, the token transmits bits by emulating a series of contact/no–contact made on the touch–sensing surface. This process results in varying the effective capacitance between the token contact area and touch–sensing surface. This variation is recognized as "Press" (contact made) and "Release" (contact released), emulating the events when a human touches the screen. To minimize errors in detecting these events, and to allow for using the same tokens across different touch–sensing devices, we introduce a self–calibrating mechanism on the token to adapt its communication parameters specifically to the device it is communicating with. The contact/no–contact process on the token is generated by turning an electrical switch on the token ON or OFF based on the encoded bits. Controlling a switch requires a very small amount of energy. We minimize it further through an module that harvests energy from the touch–sensing surface.

The energy harvesting module design is based on our novel insights that *the touch sensing surface of devices have an electric field created on the surface*. This electric field is a result of the scanning process of the touch–sensing module to detect human touch events by probing a monotone signal. Unlike other known harvesting techniques, RF [36], NFC [16]), or light [40], where the energy source availability on the touch–sensing device can be unpredictable, this electric field is always available on a device's touch–sensing surface when it is powered ON. To the best of our knowledge, this paper presents the first characterization of energy harvesting from a touch–sensor.

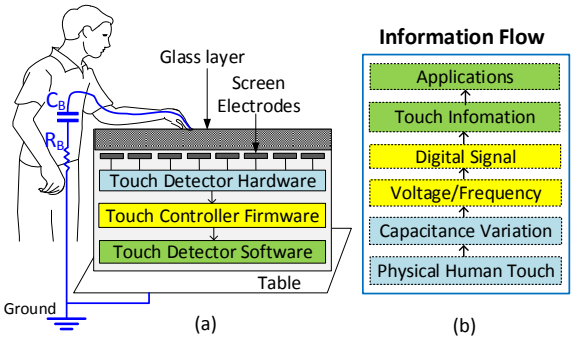


Figure 2: (a) Overview of touch–sensing in a touch–screen device, (b) The information flow of human touch event detection mechanism.

We make the following contributions in this paper:

(1) We design a system to communicate information from tokens to touch–screens by varying the capacitance of the touch–sensing surface. We introduce a self–calibrating mechanism on the tokens to minimize identification errors on the touch–sensing device and adapt usage across different types of touch–screen devices.

(2) We explore a novel way of harvesting energy from the touch–sensing surface. We characterize and design a touch–sensor harvesting component.

(3) We prototype a hardware token and implement the token identification software on touch–screen devices. We prototype multiple applications on these devices that can benefit from token identification. We design a prototype touch–sensor energy harvesting module that can be integrated with the token.

(4) We evaluate the energy consumption of the token and compare with that of NFC and Bluetooth LE hardware tokens. We also study the end-to-end application performance reliability using our system towards two use–case applications: *(i) gaming token identification* and *(ii) user authentication*. We conduct a user study to evaluate the acceptance of our design among human users.

2. TOUCH SENSING BACKGROUND

Before we delve into the details of our proposed touch based token identification system design, we first provide an overview of the technical principles of the workings of a capacitive touch sensor pervasively integrated with touch–screen devices.

As depicted in Figure 2 (a), the touch–sensing module in touch–screen devices is typically composed of three key components: (i) touch–sensing hardware, (ii) touch controller firmware, and (iii) touch–detection software.

A touch detector hardware, including the touch sensor, is arranged underneath a protective and insulating layer such as glass, polymer, or plastic [39]. It comprises of the supporting circuitry to sense when a conducting material makes contact with the screen, or generates a *touch*. Touch–sensing can be accomplished using various technologies; for example, ana-

log resistive [33], surface capacitive [12], surface acoustical wave [22], infrared optical technology [5]. Of this list, surface capacitive based touch sensing [10] has been the most prominent because of its low–energy consumption, high reliability, and low manufacturing cost[1]. We will thus focus on surface capacitive based touch–sensing devices in this paper.

In a capacitive touch–sensing device, the touch controller firmware detects a touch by measuring the capacitance variation caused on the touch–sensing surface when a touch generates. The controller basically acts as an analog to digital converter that converts the detected capacitive variations to equivalent digital information to be processed by the touch detection software in the device's OS kernel.

The capacitance variation is measured by the controller using an active probing mechanism that periodically initiates an alternating current (AC) signal at frequency (f_{prob}). This probe signal is transmitted through the touch–sensor's electrodes and the differential voltage and frequency/phase of the signal reflected off the touch contact (e.g human skin) is measured. If this differential is greater than a preset (calibrated) threshold, the controller records that a touch has been initiated by a conductive material *pressing* on the touch surface. When the difference is insignificant, the controller records that the touch has been *released*. In capacitive touch–sensing devices, a touch event is typically represented as a pair of "Press" and "Release" events.

The "Press" and "Release" events recorded by the controller are converted into equivalent digital information which is processed by the touch detection software in the device's operating system kernel and then forwarded to the application layer. This software module is essentially responsible for converting the digital signals into equivalent digital codes that are classified into different types of *touch events*; for example, the human finger *tapping* or *swiping* on the touch surface. The conversion from the human touch generation to the digital touch event registration is handled by an algorithm in this software module.

3. SYSTEM OVERVIEW

To allow the touch–enabled device to identify the artifact, we propose that it would be equipped with a necessary hardware token to encode an ID (bit stream) into equivalent digital signals, that in turns would be used as an external trigger to artificially vary the capacitance between the token surface and touch–sensing module, as illustrated in Fig. 3. The capacitance variations would be detected as equivalent "Press and "Release" events that are processed as touch events by the touch–sensing mechanism on the device. The token ID is decoded through the supporting software on the device by analyzing the generated artificial touch events.

The key challenges in our proposed approach are:

- The artificial touch event triggering mechanism must work within the limited energy budget of the token.

[1]As reported in [19], more than 92% touch–sensing devices shipped in 2014 is based on this technology and the number is predicted to rise to 98% by 2018.

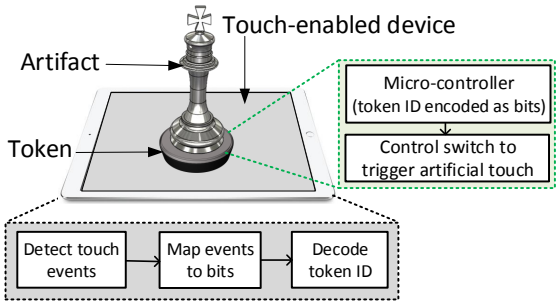


Figure 3: System Overview Diagram. An artifact is embedded with a token that communicates its ID to a touch–screen device by varying capacitance on the touch–sensing surface.

- The reliability of the touch event detection largely depends on knowledge of the probe frequency, without which the software will not be able to match the timing of the registered events with that of the touch events actually initiated.
- Since the probe frequency may vary across different touch–screen devices, adaptation of the system to different devices becomes an additional challenge.

We address the challenges in designing a capacitive touch based token identification system by devising a novel low–energy and high reliability mechanism for communicating the token information to the touch–screen device. To further minimize the system's energy consumption, we explore the possibility of rendering the token *battery–free* by designing a module to harvest energy from touch–screen when the token is in contact with the screen and aim to channelize this energy towards the token identification process. We will now provide an overview of the token identification and the energy harvesting aspects of our proposed system design.

3.1 Token Identification through Capacitive Variations

We integrate a transmitter module on the token that translates information (e.g. ID) bits into a series of ON–OFF pulses using a microcontroller. The pulses control the ON–OFF states of a switch on the token which in turn controls the mechanical contact of the token's conductive surface on the touch–screen surface; ON implies the token is in contact and OFF implies it is not. This switching mechanism triggers capacitance variations between the conductive surface of a token and the touch–screen contact point creating artificial "Press" (ON state) and "Release" events (OFF state), which get registered as touch events by the touch–sensing mechanism on the device. This method of communicating bits by *emulating* the process of touch–event generation on a touch–screen device expends minimal energy on the token as the electric switch can be controlled with very low current draw from a battery.

The rate of generation of touch events depends on the probe frequency of the screen, f_{prob}. The reliability of the touch event detection largely depends on knowledge of this

probe frequency. The rate of sampling the touch events (by the touch controller) on the device must have a deterministic relation with this probe frequency. If not, the software on the device will not be able to match the timing of the registered events with that of the touch events actually initiated, resulting in erroneous touch events. To address this issue, we incorporate a self–calibrating mechanism into the token that allows for automatically detecting this probe frequency when contacting the touch sensing surface of a touch–screen device. With the knowledge of this probe frequency, the token will be able to adapt the rate of generation of the "Press" and "Release" events such that the threshold for filtering out erroneously initiated touch events can be predicted on the touch–sensing device and thus minimize the errors in detecting the artificial touch events.

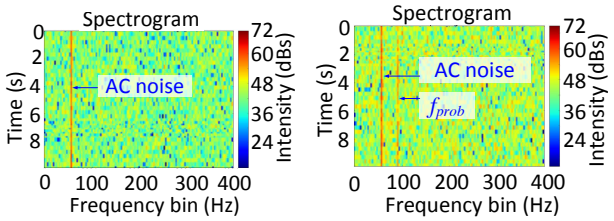
The token identification process involves two phases:
(i) *Self–Calibration* phase, where the touch–screen is profiled for measuring probe frequency, f_{prob}, and the touch events generated by the capacitance variations are characterized as a function of this probe frequency. Therefore, the effective sampling duration and detection thresholds for reliably detecting artificial touch events is estimated.
(ii) *Sensing* phase, where information encoded as bits is translated into equivalent capacitance variations to generate artificial touch events by the token, that are sensed by the touch–screen sensing mechanism on the device. The sensed touch events are decoded into equivalent information bits through software in the device.

3.2 Energy Harvesting from Touch–Screen

Based on our understanding of the touch–sensing mechanism, we realize that the electrodes residing below the touch–screen surface periodically undergo a charging and discharging phenomenon to assist the scanning process to sense a touch. Through a simple experiment we observed that the charge and discharging results in a small voltage leakage which resides on the screen's surface. Based on this observation we design a circuitry to harvest this *leakage voltage* from the touch–screen to charge a storage capacitor. The energy stored in the capacitor will be used to power up the token when it makes contact with the screen. The key idea of our energy harvesting component design is to scan the touch–screen surface over a frequency range and filter the touch scanning frequency. Once the frequency is isolated, the current flow due to the leakage voltage is directed towards a capacitor using a rectifier. We will discuss the energy harvesting component design in more detail in section 6.

4. SELF-CALIBRATION VIA TOUCH-SENSOR PROFILING

To reliably and effectively generate touch events, it is important for the token to operate with a proper configuration that fits with the touch sensor it communicating with. Since different touch sensor on the market has drastically different internal operating parameters (e.g. sampling rate, probing signal frequency, etc.), the token needs to be able to learn these key parameters, from which it will derive the proper configuration for event generation and communication. In practice, however, directly measuring these parameters from the surface of the touch device is challenging since it is not possible to get physical access to the touch sensors unless the device is cracked opened. In addition, this information is often not available from the device's datasheets. Even when it is available, the actual operating probe frequency of the device is often different from what specified by the manufacturer. We introduced a novel profiling method to overcome this challenge.



(a) Screen is turned off (b) Screen is turned on

Figure 4: Frequency distribution of the electrical signal captured from touch–screen surface of Samsung Galaxy. We placed an electrode on the surface of the touch screen and captured the electrical signal as a digital quantity on a microcontroller.

Our method relies on the following intuition: since the capacitance variation is measured by the probe signal that creates an electric field on the touch surface, it might be possible to estimate the internal sensing parameters indirectly if we can capture the electric field generated by the probe signal. In addition, the probe frequency should be one of the frequency components of the electric field generated on the touch surface when it is switched ON. We confirm this intuition through a feasibility experiment (Figure 4) where we placed an electrode on the surface of a tablet's touchscreen and analyzed the frequency distribution of the electrical signal on the surface. Shown in the figure, probe frequency can be clearly identified when the screen is ON proving that (1) the electric field can be captured with a single electrode and (2) the captured electric field signal contains internal sensing parameters of our interest. We use this insight to develop a methodology to measure the probe frequency directly by token when it makes contact with the touch sensor's surface.

4.1 Timing Characterization of Touch Sensing

As mentioned earlier, we propose to use "Press" and "Release" events to represent and transmit data sequences. We first characterize these events by analyzing the charging and discharging behavior of the capacitance on the surface as illustrated in Figure 5.

Let ΔC be the capacitance variation that the touch sensor observes. This variation is essentially the difference between the capacitance value between two temporal checkpoints (sensing duration), preset by the internal sensing algorithm. Upon a touch event initiation, the capacitance value

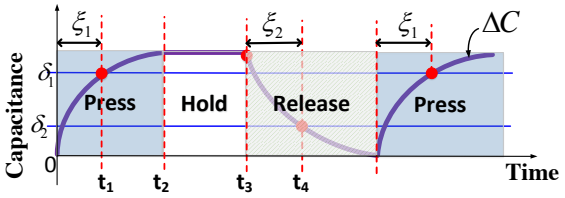


Figure 5: Illustration of capacitance variations as a function of time.

increases as the charge accumulates, and the sensor detects a "Press" event when the measured ΔC is greater than threshold value δ_1 (at Time $= t_1$). The capacitance increases until it reaches a saturation point (at Time $= t_2$) and stays in a "Hold" state until the touch is released. When the touch is being released (at Time $= t_3$), the capacitance value gradually decays (discharging) and the sensor detects a "Release" event when $\Delta C < \delta_2$ (at Time $= t_4$). The capacitance continues to decay until it reaches the reference level (0) and stays in that state until the next touch event is triggered.

Therefore, the sensing duration of a touch event can be characterized by the timing duration of "Press" and "Release" events as,

$$\begin{aligned} T_{press} &= \xi_1 + \tau_{SS} \\ T_{release} &= \xi_2 + \tau_{SS}. \end{aligned} \tag{1}$$

where, τ_{SS} is the propagation delay in conveying the sensed information from the sensor to the application layer through the touch device's software stack.

Here, δ_1 and δ_2 are the thresholds for detecting "Press" and "Release" events, respectively, and are preset by the device manufacturer. This implies that the value of the sensing durations ξ_1 and ξ_2 is not easily available and vary among devices depending on the touch sensor used, and thus have to be measured. Hence, in our design we propose to measure these sensing durations for each touch device through a one time self-calibration phase.

This timing characterization helps in designing the equivalent trigger pulse durations to generate the artificial "Press" and "Release" using the token. However, due to the unpredictable delay factor τ_{SS}, some of the touch events may be missed (not detected) by the sensing mechanism due to the timing mismatch of the token transmission rate and the touch sensor's sampling duration. If the sampling duration (or rate) of the touch sensor is known it will be possible to calibrate the token to the sensor's "Press" and "Release" sensing durations precisely. Knowledge of the sampling duration requires the measurement of the screen's probe frequency, f_{prob}.

4.2 Probe Frequency Estimation

Based on our preliminary feasibility experiment results in Figure 4 we realized that it is possible to measure the probe frequency, f_{prob} directly from the touch sensor's surface by analyzing the frequency spectrum of the electric field signal captured on the touch surface. However, performing frequency analysis on the token would consume a lot of energy and require a relatively powerful microprocessor.

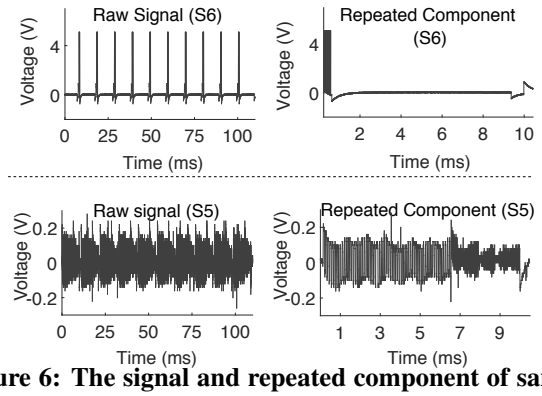


Figure 6: The signal and repeated component of sample devices.

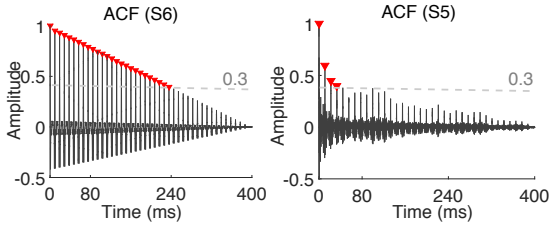


Figure 7: The result of auto-correlation function from the signal of Samsung Galaxy S6 and S5.

To mitigate this problem and minimize energy requirement, we develop an alternative time–based technique to extract the probe frequency by leveraging auto-correlation concept. We analyzed the auto–correlation function (ACF) of the signal captured from the touch sensor surface to identify the most time–repetitive signal component. This most repetitive component is the probe signal. Its length in time domain is equal to the distance from the first peak to the second peak of the ACF (period p); the frequency f_{prob} is computed as $f_{prob} = \frac{1}{p}$.

Figure 6 shows the time series of electric field signal captured on the touch surface of Samsung Galaxy S6 and S5, in which a repetitive pattern of the probing signal can be clearly identified. Their corresponding ACFs are shown in Figure. 7; here a threshold value of 0.3 was used to terminate the ACF computation. The selection of the threshold only impacts the running time of this one–time self–calibration process but not the accuracy of f_{prob} estimation. To empirically validate our algorithm, we performed the self–calibration on 12 other devices, with results reported in Figure 8. In the course of this profiling experiment we also measured the execution time of this one–time calibration process to be about 4 seconds, which is the total time taken for the token to determine f_{prob} from the time it makes contact with the screen.

5. SENSING TOKEN'S ID THROUGH ITS CONTACT

In this section, we first describe our algorithm designs that allow a token to represent its ID through a time series of capacitance variations. We then show how the touch–enabled device decodes the token's ID from the series of touch events generated by such capacitance variation sequence. We dis-

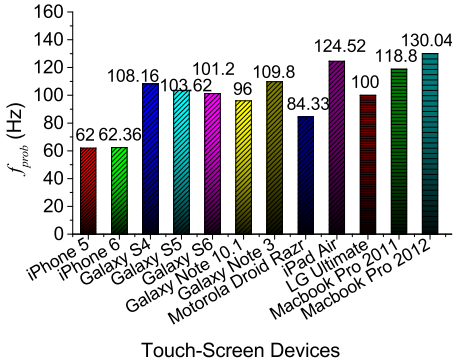


Figure 8: Measured probe frequency of different touch–screen devices.

cuss the mechanisms using a working example of Samsung Galaxy S5 device.

5.1 Representing Data through Capacitance Variations

As explained earlier, our token can create artificial “Press" and “Release" events on the touch device by varying its capacitance when they are in physical contact. We study the arrival time information of these events (on the device) to help design a data structure for information transmission. In the following discussion, we show how the token represents bit ones and bit zeros by controlling the timing information of the “Press" and “Release" events.

Pulse width modulation (PWM) is used to represent the data sequence. Specifically, a bit one is represented by a “Press" event followed by a “Release" event that are T_{one} milliseconds apart. Likewise, a bit zero is represented by a “Press" event followed by a “Release" event that are T_{zero} milliseconds apart (T_{one} must be different from T_{zero}). This means that the token needs to close the switch to vary its capacitance and hold the switch at the close position for T_{one} milliseconds in order to indicate to the receiver that it wants to transmit bit one. The holding time will be T_{zero} milliseconds if bit zero needs to be transmitted. The challenge here is determination of these two time constants.

From a data rate perspective, it is intuitive that smaller time constants are desirable as it will yield a higher data rate. However, if these two time constants are too small, the touch sensors cannot respond fast enough to register correspondingly generated events. Specifically, if the two time constants are smaller than the probe period ($\frac{1}{f_{prob}}$), the event will be missed by the touch controller as it would not have been sampled. In addition, we note that there is a variable delay from the moment that the token toggles its switch (i.e open or close) until a corresponding event is registered and delivered to the application layer of the touch–enabled device. This variable delay is captured in τ_{SS}: consolidation of the queuing and propagation delays in conveying the sensed information from the sensor to the application layer through the touch device’s software stack. Therefore, the token needs to select T_{one} and T_{zero} in such a way that such variation does not confuse the corresponding pulse width demodulation deployed in the software receiver on the touch–enabled device. Lastly, if the two time constants are too high, the system can operate only at very low data rate. We propose to determine T_{one} and T_{zero} as follows:

$$T_{one} = 2 \text{ x } (\frac{1}{f_{prob}}) + \text{Max}(\tau_{SS});\ T_{zero} = T_{one} + (\frac{1}{f_{prob}}) \quad (2)$$

in which $\frac{1}{f_{prob}}$ is measured from the self-calibration step and Max(τ_{SS}) is conservatively assigned to be 2 ms; note that OS–based propagation delay are typically smaller than 1 ms in almost all modern OS’s.

We pack a fixed–length payload into a data frame that has $[prefix||data||suffix]$ format. The $prefix$ is used as pilot symbols while the $suffix$ contains the parity check together with the frame ending indicator. A silence period of 3 x $\frac{1}{f_{prob}}$ is used for frame ending indication.

Algorithm 1: Finding threshold (γ) in time arrival for differentiating touches representing bits $1s$ and $0s$

input : $E_{discrete}$ - Event sequence in time domain
$TrBitSeq$ - Transmitted bit sequence 1^l0^l
output: γ

Extract the type of events: E_p is the set of press events and E_r is the set of release events.
/* Find the press delay of all emulated touches (Γ)*/
for $j = 1 \rightarrow \frac{max(E_{discrete})}{2}$ **do**
  $\Gamma[j] \leftarrow E_r(j) - E_p(j)$
/* Check the number of received events */
if $|E_p| \neq |E_r|$ *OR* $|E_p| \neq |TrBitSeq|$ *OR* $|E_r| \neq |TrBitSeq|$ **then** $\gamma \leftarrow NULL$
else
  for $k = 1 \rightarrow \max\Gamma$ **do**
    if $TrBitSeq(k) == 1$ **then** $\Gamma_1 = \Gamma_1 \bigcup \Gamma(k)$
    else $\Gamma_0 = \Gamma_0 \bigcup \Gamma(k)$
/* Find the threshold */
if $\exists$*temp*, $\max\Gamma_1 < temp < \min\Gamma_0$ **then** $\gamma \leftarrow temp$
else $\gamma \leftarrow NULL$
return γ;

5.2 Decoding Transmitted Data

The decoding process relies on the duration between a pair of “Press" and “Release" events to retrieve each communicated bit and then reconstructs the originally transmitted payload data frame. However, the key challenge here is the fact that the receiver software is not aware of what values of T_{one} and T_{zero} are being used by the transmitting token. In this regard, we incorporate a self–calibration method to determine a threshold γ to help the decoding mechanism identify whether a received duration represents a 1 or 0. To identify γ, the calibration process works as follow. The token sends a 100 bit sequence of alternating 1s and 0s. Based on the received series of events, the decoder finds a threshold γ which

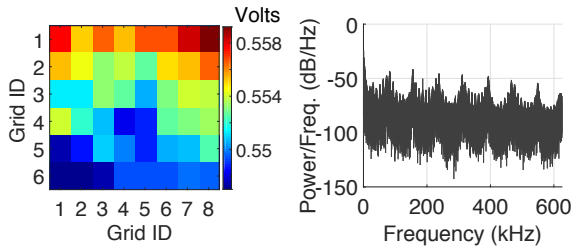


Figure 9: Distribution of measured voltage of the electric field on touch–screen surface (left). The touch surface of a Samsung Galaxy S6 was virtually partitioned into grids and one measurement was taken for each grid, Power spectral density of the measured electric field across the touch surface (right).

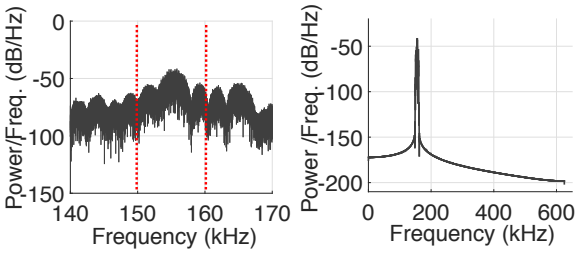


Figure 10: Power spectrum containing the peak power at 155KHz (left), and corresponding power spectrum of the band–pass filtered signal (measurements for Samsung Galaxy S6) (right).

can be used to reconstruct the bit sequence. Once the threshold is calculated, the demodulation is straightforward. One possible realization of the threshold selection is described in Algorithm 1.

6. TOUCH SCREEN ENERGY HARVESTING

The scanning mechanism of the touch–sensing module to detect touch events creates an electric field on the touch–screen surface. The availability of this electric field in contact range of the token opens up the possibility of harvesting this indirect *energy source* by using it as a voltage source to drive the token, thus making it a *battery–free token*.

6.1 Touch–Screen as a Voltage Source

We conducted an experiment where we attached a conductive material on the contact surface of the token, and profiled the electric field on the touch–screen surface for its voltage (will refer to as *leakage voltage*) and frequency spectrum. We can infer from Figure 9 (left) that the leakage voltage is almost uniform across the surface. However, we observe from Figure 9 (right) that the frequency distribution is spread across a band. Thus, isolating the frequency corresponding to the AC leakage voltage source is necessary.

As the touch–screen energy source is at physical contact distance, the path–loss is almost zero. Hence, we realize that the peak in the power spectrum will be dominated by the leakage voltage signal. Figure. 9 (left) and (right) show frequency band where the peak lies and the bandpass filtered

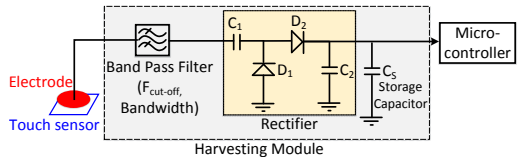


Figure 11: Schematic of the touch energy harvester.

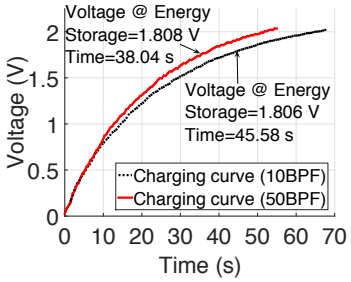


Figure 12: The charging curve of harvested energy. 10BPF = 10th order bandpass filter, 50BPF = 50th order bandpass filter.

spectrum, respectively. In this way, we characterize a touch–screen as an AC voltage source with certain peak–voltage and source frequency.

6.2 Energy Harvesting Component Design

Treating the touch–screen as a voltage source, we design a hardware module for harvesting energy from the touch–screen surface. This module can be integrated with the token by wiring it in series the conductive surface of the token. The key components of this module include a bandpass filter, a rectifier, and a capacitor. A schematic of the module is shown in Figure. 11. The band pass filter isolates the components of voltage emission signal which have high power density at peak signal frequency. The rectifier functions as a half−wave voltage rectifier. It uses Schottky diode as non-linear component which operates much faster than traditional diode due to its metal-semiconductor junction which gives a forward voltage drop of as low as 0.15V[30]. The capacitor stores the rectified energy by a rectifier and provides energy to micro-controller.

6.3 Use–Case Analysis

Considering the Samsung Galaxy S6 use–case, we design a harvesting module that uses a 10^{th} order Butterworth Band–Pass Filter (BPF) with center frequency of 155KHz and the lower and upper cut–off frequencies of 150KHz and 160KHz, respectively. The cutoff range can be determined during the one –time calibration phase The harvesting module uses 4-stage Dickson diode-based voltage rectifier. We employed an Avago HSMS−285C [4] Schottky diode which has a turn-on voltage of 150mV measured at forward current of 100μA as non-linear component of rectifier. We used a 470μF 5V capacitor as the energy sink, whose value is calculated for matching the peak signal frequency. Powering the token mainly requires powering up the micro-controller and the relay switch (to trigger pulses that initiate capacitance vari-

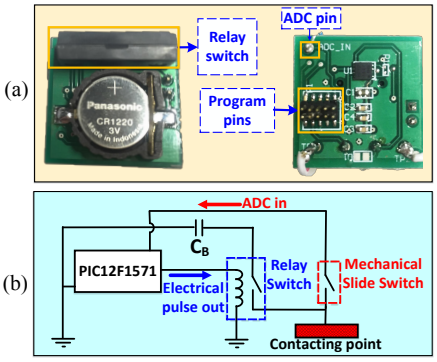


Figure 13: (a) PCB design and (b) Token's schematic.

ations), which requires at least a supply voltage of 1.8V. At this voltage, the generated current by energy harvesting module is 100mA. We measured that the current required to generate 1 bit on the token is about 10mA, thus the harvested module can generate about 10 bits once powered up from a cold start.

Based on the charging graph of the storage capacitor during harvesting, as shown in Figure 12, it takes about 45s to charge up to 1.8V from cold start; this can be reduced to 38s using a higher order bandpass filter. Note that this duration is the time the system requires to power up the token circuitry from zero supply voltage and zero residual charge. This measurement is done offline, we reserve the circuit integration for future work. The current size of the circuit is 2.5cm x 2cm. The key challenge of integrating energy harvesting part is how to reduce circuit's size making it wearable. In addition, a throughout evaluation is needed to understand the charging behaviors of all touch-devices and to optimize the energy harvesting performance across those devices. During operation we propose to keep the token at this minimum supply voltage even during idle modes. Since the harvesting and the token identification process can happen in parallel, effectively, the token operation can be rendered *battery–free*.

7. PROTOTYPE IMPLEMENTATION

We prototyped a hardware token and implemented the software for token identification using smartphone and tablet touch–screen devices as a running examples.

Hardware Prototype for Identification Token. The schematic and printed circuit board of our prototype is shown in Figure 13. The token consists of a microcontroller PIC12F1571 [26] with a flash memory unit. The microcontroller is programmed to generate ON/OFF electrical pulses corresponding to the 1s/0s of the token ID bit stream. These pulses open and close a Reed relay switch [25]. The switching process varies the capacitance of the token's contact point with the touch surface by connecting and disconnecting the contact point through a 100μF capacitor.

We provide a mechanical slide switch on the token to allow toggling between two operating modes: calibration or communication. In the calibration mode, the token conducts the one–time probe frequency profiling procedure if registering with the touch–sensing device for the first time. During subsequent operations this mode involves the token self–calibrating its transmission rate based on the probe frequency and sampling rate of the touch–sensing device. The token software (electrical pulse generation, CRC computation, pilot and header generation, and parameter extraction) has been developed on MPLAB X IDE development platform.

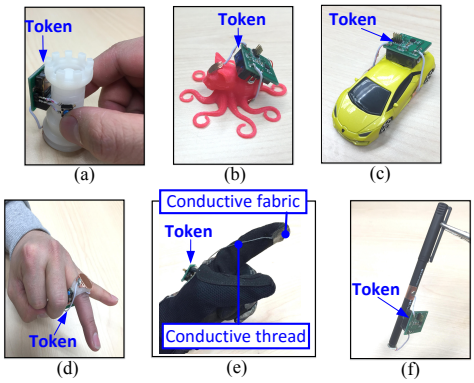


Figure 14: Example set of our developed prototypes: (a) a chess piece, (b) a 3D printed object, (c) a kid's toy, (d) a smart ring, and (e) a smart glove, and (f) a smart pen.

We use a coin cell (3V) battery to power the token when the harvesting module is detached. This also serves as a backup power source during the calibration phase. The size of our current token prototype is 4cm^2 (negligible thickness). We aim to reduce the form factor using surface mounted components in future designs.

Software on Touch Devices. We implemented the software modules for token identification as individual apps on Android OS enabled touch–screen devices. The apps are set to detect "Press" and "Release" events using the MotionEvent class [13] from the touch–sensing API provided in Android. The class helps to extract the event $time$ and $touchtype$ which are the key parameters used to map the detected touch events into bits.

Augmenting Physical Artifacts. We integrated the token with real world artifacts as shown in Figure 14.

Smart 3D printed artifacts. We integrated our token with 3D printed artifacts that includes a gaming artifacts (Figure. 14 (a–c)) and a wearable ring (Figure. 14 (d)). We attached the token to these artifacts with the token's contact surface facing out. We use the chess piece and the ring tokens towards evaluating a prototype *object identification* application.

Smart glove. We created a smart glove contraption that can be identified by a touch–screen device, by augmenting a commodity fabric glove (Mechanix [24]) with our token. The finger tip on the glove was covered with a conductive material which was wired to the contact point of our token using a low impedance conductive thread (annotated in Figure. 14 (e)). We use this prototype to evaluate the reliability of a prototype *two–factor authentication* application for touch–screen devices.

Smart stylus (pen). We created a smart stylus contraption

by connecting the tip of the stylus to the output of our token. This augmentation enables the supporting application on the touch–screen device to associate every touch of the stylus on the screen surface with its associated ID. This feature can help provide multi–user support for collaborative working applications as well as multi–user gaming.

8. EVALUATION

We conduct experiments to evaluate the energy consumption and identification reliability of our token identification system. In particular we evaluate the following:
(i) Token energy consumption per identification attempt, and compared with NFC and Bluetooth tokens.
(ii) Bit error rate of proposed communication mechanism.
(iii) End–End application performance for prototype apps: (a) object identification through touch, and (b) two–factor authentication in a single step.
(iv) User study evaluation.[2].

8.1 Energy Consumed Per Identification

The energy consumption of the token identification system includes that of the token and the touch–sensing device. We evaluate the energy consumption on the token and compare it with competitive token identification technologies.

8.1.1 Energy consumption of our Token

The energy consumption of the token includes that of the *micro controller* and *relay* switch. The energy consumption can be expressed analytically as,

$$E = U \times (I_{relay} + I_{mc}) \times \frac{L}{f}, \tag{3}$$

where I_{relay} is the forward current to drive the relay switch, U is the supply voltage, I_{mc} is the current draw by the micro controller, L is the token ID data size, and f is the data rate.

The micro-controller from Microchip can operated in an extreme low power mode at 0.03mA/MHz with supply of 1.8V [26]. The OMRON relay(G3VM-_AYX/@DYX) [28], draws 10mA forward current at 1.63V forward voltage. Based on the calibration, the sampling duration on a Samsung Galaxy S5 must be minimum at $\Delta t = \Delta s = 12ms$; implies the capacitance variation technique requires $\frac{1000(ms)\times 2}{T_{bit1}+T_{bit0}}$) = 30 ms for 16 bits data size.Therefore an effective data rate of 33.3bits/s can be achieved. The transmission duration for 16 bits is measured to be 0.48 seconds. Therefore, the energy consumption is $E = 1.8 \times (10 + 0.03mA) \times 0.48 = 8.6mJ$.

We measured the average power and current draw (Figure. 15) from the coin battery for each component (profiling, relay and circuit) of the token when transmitting a series of 1s and 0s for 0.2s. The average power consumption of transmission (profiling is done apriori) is 12.99 mW for a duration of 0.2s at an average current draw of 5mA.

8.1.2 Comparative Evaluation

We prototyped a Bluetooth BLE and NFC P2P token implementations as shown in Figure. 17. The BLE token uses a low energy HM–10 module [7], driven by an Arduino Pro Mini [2] to transmit an ID decoded by the BLE module of an Android device. The NFC token uses a Sparkfun RFID module [34] for communication controlled by an Arduino. We setup a host–based card emulation on Android to receive the ID transmitted from the NFC module. We measure the energy consumption for each identification attempt (transmit and decode by an Android device) using a Monsoon power monitor [20].

We report the token's energy consumption per identification for different token ID sizes in Figure. 16. We can observe that the token energy consumption is linearly proportional to the ID size and also it monotonically increases at a significantly faster rate than BLE and NFC. However, we observe that our token consumes less energy than NFC and BLE at small data payload sizes; crossover occurs at 304 and 416 bits, respectively.

Both, BLE and NFC have high initialization overhead, compared to our approach, due to pairing and waking up from idle mode. However, BLE and NFC have much higher data transmission rates (2.1Mbit/s and 424kbits/s, respectively) compared to our approach (40bps), and that the overhead only incurs only one time per identification, the benefit amortizes as the ID length increases. Therefore, NFC and BLE outperform our approach when the ID length precisely exceeds 304, 416 respectively. We note that a large number of identification applications [3] typically consider 128 bit IDs, in which case our system can outperform BLE and NFC. We also observed through our measurements that the idle mode (token is ON but no transmissions) energy consumption of our token (3.35mW) is atleast 10x energy efficient than BLE (44.49mW) and NFC (60.54mW) tokens.

Meanwhile, state-of-the-art technique for generating the touch [38] to the screen uses 9V voltage and inject to the screen continuously at the data rate of 4 bits/s. Given the requirement of transmitting 16 bits of data, the Capacitive Touch Communication (CTC [38]) technique, which operates at 4bits/s, consumes 1800mJ of energy. The energy consumption of CTC is about two orders of magnitude larger than our approach as the former requires a 9V signal injection into the screen, which drains a lot of battery power merely to generate this signal.

8.2 Benefits of Self-calibration

As discussed earlier the self–calibration through touch–screen profiling stage is the key factor in minimizing the energy consumption of artificial touch generation. This process also helps in achieving high communication reliability. We evaluate this reliability Bit Error Rate (BER) metric.

Recall that the self–calibration is done through a profiling step in which the token extracts key parameters that characterize the touch–sensing mechanism; its detection frequency, charging and discharging times, and touch event propagation time. Without this step, the token must make a heuristic approximation about what communication parameters are best suited for interacting with the particular touch–sensing device.

[2]the human user study was approved by our institution's IRB.

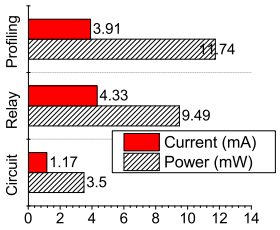


Figure 15: Power & current draw of token's components.

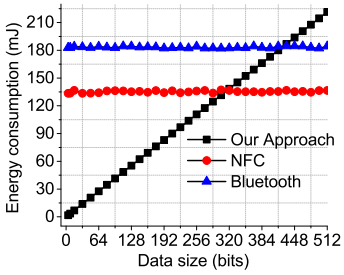


Figure 16: Comparison of energy consumption vs. token ID data size.

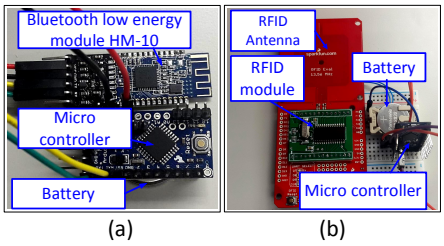


Figure 17: Prototype setup of Bluetooth Low Energy and NFC P2P.

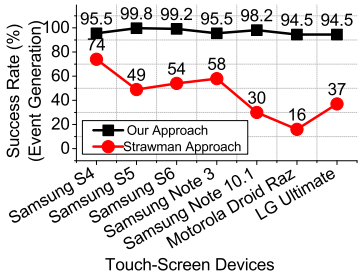


Figure 18: Touch event generation success rate on different devices.

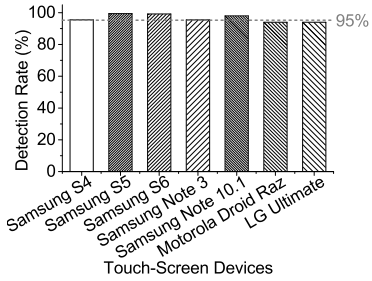


Figure 19: Object detection rate of our approach.

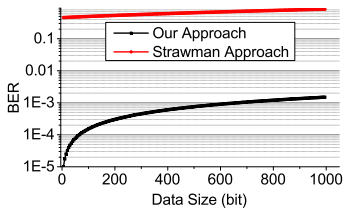


Figure 20: Communication BER versus data size. Strawman uses a heuristic calibration mechanism with touch-based communication.

Strawman Approach. Let us consider a strawman example for a heuristic that determines the communication parameters. Let us consider that the sensor's detection frequency, f_r is the same as the screen's refreshing rate. For example, Samsung S6 devices feature the CapSense touch controller which operates at 120Hz refreshing rate [21]. This implies that the token must be configured to generate capacitance variations (at sampling rate f_s) at a rate of at least 120Hz. Let us fix the charging and discharging durations, ξ_1, ξ_2, at 2 ms based on an empirical estimations through measurements.

In the touch sensing module, the bit detection errors (thus BER) will be ideally zero if the number of events sensed by the touch device is exactly equal to the number of events intentionally generated by the token. Therefore, the success rate of detecting the token generated touch events – the ratio between the number of events that the touch sensor receives and the total number of events that are generated by the token – defines the BER curve.

In Figure 18, we show the success rate of event generation on the token, over 10^3 events for 7 touch–enabled devices. We clearly observe that self–calibration helps in generating touch events on the token with high reliability, significantly higher (about 6x in best case) than the strawman approach. We observe a significant difference (3–4 orders of magnitude) in BER of our system compared to the strawman as shown in Figure 20. The experiment confirms that the use of our profiling based self–calibration approach can significantly outperform heuristic approximation through empirical measurements for touch based communication.

8.3 Application-based Evaluation

We evaluate our system using two types of applications. We discuss its ability to associate a token's ID to its touches, and also evaluate the performance of a novel application that allows for 2–factor user authentication in a single step.

8.3.1 *Object Identification Through Its Touches*

We attached 3D printed artifacts (5 artifacts as illustrated in Section 7) with a token of 64 bit ID size and transmission rate of 30–46 bps; depending on its self–calibration output. We customized the pre–installed software for the application on the touch–screen device to identify the token. We conducted the experiment by testing the token identification over different locations on the touch–screen, repeated over 400 trials and tested on 7 touch–screen devices.

Figure 19 reports the object identification accuracy through the token detection rate (fraction of total number of times the token is correctly identified). We observe that it is possible to identify objects with at least 95%. We observed a negligible false detection rate in our experiment. However, we believe that the false detection rate may become non-zero, as the number of trials increase, yet stay low due to the self–calibration process.

8.3.2 *Two–factor Authentication in Single Step*

Use–Case Definition. When a token is worn by a user, such as in a smart glove, two–factor authentication in single step can be enabled. In this application, a user can perform two–factor authentication through a single step process of typing in a password/passcode. When the user touches an alphanumerical on the screen with the smart glove, the token simultaneously transfers the corresponding part of its ID simultaneously. By the time that the user finishes entering the password, the token ID transmission is also completed. This method of authentication eliminates the need to carry multiple physical entities corresponding to each authentication

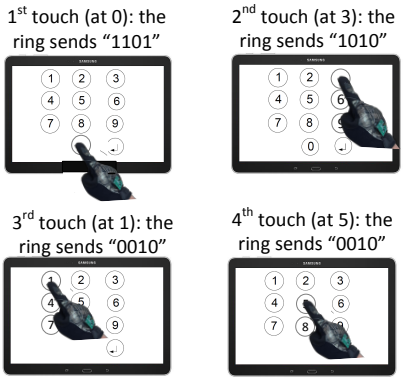


Figure 21: An example workflow of performing two–factor authentication in a single step. Here, a passcode 0315 is being entered.

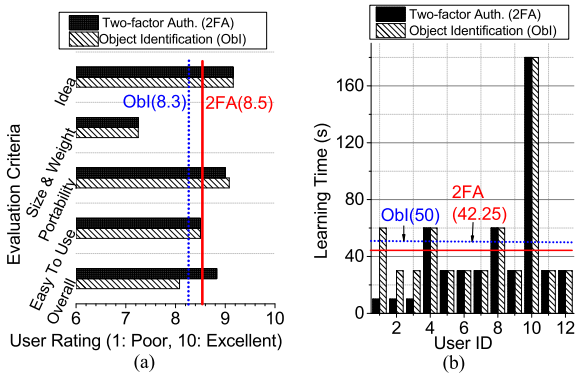


Figure 22: The user study results on 12 participants: (a) The summary of user rating on the technical idea, size, weight portability, easy to use, and overall; and (b) Learning time of users to use the token identification system.

factor as in traditional 2–factor authentication (e.g. password and smartcard). Figure 21 shows an example of two–factor authentication in which the pass code contains 4 digits 0, 3, 5, and 1 and the authorized token ID is “1101 1010 0010 0010". The application allows user to access the device only if the correct pass code is entered and when the token ID is identified correctly.

Detection Rate. We use a 16bit token ID for our evaluation of this application. We conduct the experiment of typing in a 16 bit equivalent password (4 characters); example in Figure 21) and repeat the same for 100 trials. We observed a 92% password identification accuracy (token ID identified). We suspect that the 8 incorrect cases were caused by the users’ typing habits; for example, the finger is lifted from the screen after each touch before the bit sequence gets successfully transmitted. We confirm the impact of such user behaviors through our user–study to be discussed ahead.

Authentication Time. The time that it takes for user authentication is comparable to that of NFC and BLE systems. The BLE and NFC P2P approach take about 3 seconds to complete wake up, pairing, and communicating the ID. The dominant time factor in our technique is not from the ID communication process but from the user’s typing behavior. For example, for communicating 1 character (4 bit sequence) on each touch, our communication technique on a Samsung Galaxy S6 phone takes 121 ms which is less than a half of typical typing durations (250ms).

Discussion on Security Level. A common 4 digit PIN pass code on Android or iOS has maximum 13 bit entropy ($10^4 \approx 2^{13}$. A n–character password on iOS has maximum $6.27n$ bit entropy ($77^n \approx 2^{6.27n}$). Android pattern lock is estimated to be 19 bit entropy (2^{19})[9]. The proposed two–factor approach can significantly improve the security level of password based authentication systems as it requires a physical token for authentication. It has 3 $\times m \times n$ bit entropy (i.e. $2^{3\times m\times n}$ possible combinations) in which n is the ID length and the passcode is of length m with each digit in [0–9]. The security levels would be increased further if ASCII passcodes are used.

8.4 User Study

We conducted an user study to evaluate the readiness of users to adopt our technology.

Setup. We conducted the study using 12 participants (seven males and five females) whose within the age group of 18 to 44 years. The participants were all graduate and undergraduate students from computer science and electrical engineering majors. An IRB for the study was approved and qualified for minimum risk exemption. Participants were briefed for 10 minutes about the ID tokens and the underlying technology. This introduction also included demonstration of how to use the ID tokens for object identification and user authentication purposes. We presented two types of prototypes to each participant: a smart glove for two–factor authentication in single step (one for each hand) and a chess piece for on–touch–screen gaming application (quantity = 3).

Study procedure. After the introduction session, participants were provided with the tokens and directed to use them towards the identification and authentication applications. For each participant, we recorded the duration it took each user to confirm (verbally) that they are comfortable with the token usage setup; we will refer to this as learning time. At the end of the study, we provided each participant with a survey form. The survey contained questions that asked users to grade their interest on the current prototypes from 1–10, with 10 being excellent. The ratings were garnered for the idea, size, weight, portability, ease of use, and overall rating. We also asked *what their opinions are about the strengths and weaknesses of the current token; and what we need to improve to make better tokens*.

Survey Results. Figure 22 (a) summarizes the users’ responses on the survey, and Figure 22 (b) summarizes the learning time of users. We can infer that most users require very little time (about 40–50 sec on avg) to familiarize with the tokens. We also can infer that users were typically very positive about the usage of this technology and appreciated its convenience and fundamental idea as a whole.

Participant distribution and behavior. We acknowledge that the results from our study could have a small bias factor

as the participants' background is in either computer science or electrical engineering. However, the participants did not have any prior knowledge about our technology. During the study, we observed that most of them tinkered around with the token for about half hour when the study supervisor left the room after introduction session.

Other feedback from users. We obtained some extra feedback about our system in terms of remarks and questions from users: (i) *"For two–factor authentication tokens, thieves might recognize users if the token is conspicuous. This can push the users into an unsafe situation."*; (ii) *"I don't like to wear a ring or gloves for authentication"*; (iii) *"I would love to have my bio–metric parameter embedded into the pairing code so that I can be exclusive user of the token."*.

Our response for these remarks and questions is that our tokens are currently in prototype phase. We aim to miniaturize these tokens in our future design such that they can be inconspicuously embedded into daily usage accessories such as rings, gloves, toys etc. While we agree that exploring bio–metric signals can help increase the security level of the system, it remains outside the scope of this paper.

9. RELATED WORK

In this section, we discuss related works in four key areas that pertains to the contributions of our work in this paper.

Touch–based Interaction Techniques. Touché [32] and Capacitive Fingerprinting [17] propose to use variants of a technique called Swept Frequency Capacitive Sensing to recognize human hand, body configurations, and bio–signatures. The technique fundamentally involves the touch–sensing hardware customized to transmit signals across a band of frequencies which get reflected back from the human contact surface. The signals are detected by a built–in receiver component and analyzed to recognize human body configurations. The drawback of this approach is the hardware customization required to tweak touch–sensors towards the Swept Frequency Capacitive Sensing.

Capacitive sensing and coupling. The idea of using *capacitive coupling* for very short–range communication has been explored extensively in both, academia and industry. Sample works in this space include Bioamp from Yahoo [18], Microchip Bodycomm [6], Ishin-Den-Shin [35], Sony's TouchNet [23], Ericsson's Connected Me [37], and KAIST Semiconductor System Lab research [44]. This approach involves using the *capacitive coupling* concept to couple electrical signal, pertaining to the information to be communicated, generated by a external transmitter with a receiver integrated on the mobile device. This mechanism uses the human body as a medium for conducting the signals. The main drawback of this approach is the need for designing a custom receiver as the electrodes and controllers for capacitive coupling are not integrated defacto in mobile devices.

Capacitive proximity sensing kits have become prevalent in recent times; in particular, OpenCapSense [14], CapToolKit [42], and CapNFC [15] provide capacitive receivers to measure capacitance changes caused by human body or object movements. It is notable that this fundamental idea was used for designing short–range communication systems through near–field electro–static coupling, proposed by Zimmerman in 1996 [45]. While these kits provide excellent tools for quick prototyping of capacitive sensing systems, they do not an encapsulate and end–to–end system for identification. HumanAtenna [11] explored the idea of coupling an electric field with human body creating a virtual antenna for sensing body gestures. it requires the transmitter has wall–to–ground connection, making it not suitable for mobile devices.

Object Identification and Localization. There have been recent works on radio based radar–type tracking systems for precise localization and identification [1, 43, 29]. Objects identification can also be achieved through radio tomography and imaging techniques [31, 41] , which primarily require a large array of sensors to localize an object. While these techniques are effective in their respective domain, the application to identifying smart tokens may be very challenging considering the deployment cost and energy consumption challenges.

Energy Harvesting. Recent works have proposed energy harvesting from radio signals [36, 27]. While these systems require the present of radio signals they are constrained with a minimum size requirement to match the wavelength (order of cm to mm) of the radio signal. Energy can be also harvested from light emitted from a touch screen [40]. However, not all touch sensing devices are light emitters (e.g touch pads). Moreover, the effectiveness of the harvesting largely depends on the screen display's brightness which can vary at large; depending on the application and/or users.

10. CONCLUSION

We explored the idea of associating identities to touch events on touch–enabled devices. We proposed a token design that incorporates a low–energy and high reliability mechanism to communicate information to touch–sensing devices. We realized that by passively modifying the capacitance on touch–sensing surfaces and with help of a touch–sensor profiling mechanism to characterize the touch device, we can communicate IDs through touch at significantly low–energy. We also learned that it is possible to harvest energy from touch–sensing surface, upto significant amounts that can help operate such touch based ID tokens. Through a user study using a small group we understand that users are typically positive about our proposal however few challenges regarding intricate design details such as miniaturization, theft protection and bio–signal integration remain for future design considerations.

11. ACKNOWLEDGMENTS

We would like to thank the anonymous ACM SenSys reviewers for their helpful comments. We also thank Inworks for their training courses on 3D printing, and thank Duran RJ for his support on prototype development. This research is partially supported by Google Faculty Awards 2013-R2-634 and the US National Science Foundation Award (CNS 1452628).

12. REFERENCES

[1] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller. 3D tracking via body radio reflections. In *Proc. of the USENIX Conference on Networked Systems Design and Implementation*, NSDI'14, pages 317–329.

[2] Arduino. Arduino Pro Mini. http://tinyurl.com/zhk5mgh.

[3] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 10 2010.

[4] Avago. HSMS-285C. http://tinyurl.com/j4kcdzn, 2016.

[5] Bhalla. Comparative study of various touchscreen technologies. *Journal of Computer Apps, 2010*.

[6] Microchip bodycomm technology. http://tinyurl.com/cbpzarw.

[7] Britt & Jules. Bluetooth Low Energy HM-10. http://tinyurl.com/jrfgopv.

[8] Britt & Jules. Snowshoe Stamp. https://snowshoestamp.com/.

[9] Y. Chen, J. Sun, R. Zhang, and Y. Zhang. Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, pages 2686–2694, 2015.

[10] Chitiz Mathema and Christiana Wu, Cypress Semiconductor Corporation. Projected Capacitance Touchscreens Dominate Market. http://tinyurl.com/jd99wk7, 2014.

[11] G. Cohn, D. Morris, S. Patel, and D. Tan. Humantenna: Using the body as an antenna for real-time whole-body interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1901–1910. ACM, 2012.

[12] B. Evans. Patent Application US 4806709: Method of and apparatus for sensing the location, such as coordinates, of designated points on an electrically sensitive touch-screen surface, 1989.

[13] Google Androi. MotionEvent class. http://tinyurl.com/yjlenkw.

[14] T. Grosse-Puppendahl, Y. Berghoefer, A. Braun, R. Wimmer, and A. Kuijper. OpenCapSense: A rapid prototyping toolkit for pervasive interaction using capacitive sensing. In *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 152–159, 2013.

[15] T. Grosse-Puppendahl, S. Herber, R. Wimmer, F. Englert, S. Beck, J. von Wilmsdorff, R. Wichert, and A. Kuijper. Capacitive near-field communication for ubiquitous interaction and perception. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 231–242. ACM, 2014.

[16] J. Gummeson, B. Priyantha, and J. Liu. An energy harvesting wearable ring platform for gestureinput on surfaces. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 162–175. ACM.

[17] C. Harrison, M. Sato, and I. Poupyrev. Capacitive fingerprinting: Exploring user differentiation by sensing electrical properties of the human body. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 537–544. ACM.

[18] C. Holz and M. Knaust. Biometric touch sensing: Seamlessly augmenting each touch with continuous authentication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 303–312. ACM.

[19] C. Hsieh. Touch-panel market analysis reports 2008-2014. Technical report, DisplaySearch, 2014.

[20] M. S. Inc. Power Monitor. http://tinyurl.com/hfla5fw.

[21] S. Kolokowsky and T. Davis. Not all touch screens are created equal: how to ensure you are developing a world-class capacitive touch product. http://tinyurl.com/zpn3chb.

[22] L. L. L. Patent Application US 3885173: Apparatus and method for coupling an acoustical surface wave device to an electronic circuit, 1975.

[23] N. Matsushita, S. Tajima, Y. Ayatsuka, and J. Rekimoto. Wearable key: device for personalizing nearby environment. In *The Fourth International Symposium on Wearable Computers*, pages 119–126, 2000-10.

[24] Mechanix Glove. Tactical Gloves. http://tinyurl.com/zf3v3mn.

[25] MEDER. Relay Reed. http://tinyurl.com/zug399o, 2016.

[26] Microchip. PIC12F1571 Datasheet. http://tinyurl.com/jykj2hv/.

[27] D. Mishra, S. De, and K. R. Chowdhury. Charging time characterization for wireless RF energy transfer. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(4):362–366, 2015-04.

[28] OMRON Relay. http://tinyurl.com/zkqn6tj.

[29] J. Nanzer. *Microwave and Millimeter-wave Remote Sensing for Security Applications*. Artech House, 2012.

[30] P. Nintanavongsa, U. Muncuk, D. R. Lewis, and K. R. Chowdhury. Design optimization and implementation for rf energy harvesting circuits. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(1):24–33, March 2012.

[31] N. Patwari, L. Brewer, Q. Tate, O. Kaltiokallio, and M. Bocca. Breathfinding: A wireless network that monitors and locates breathing in a home. 8(1):30–42, 02 2014.

[32] M. Sato, I. Poupyrev, and C. Harrison. TouchÃl': Enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, CHI '12, pages 483–492. ACM, 2012.

[33] E. So, H. Zhang, and Y.-s. Guan. Sensing contact with analog resistive technology. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 806–811 vol.2, 1999.

[34] Sparkfun. RFID Module. https://www.sparkfun.com/products/10126/.

[35] Y. Suzuki et al. Ishin-Den-Shin: Transmitting Sound Through Touch. http://tinyurl.com/nu7jxav.

[36] V. Talla et al. Powering the Next Billion Devices with Wi-Fi. In *Prof. of ACM CoNext*, 2015.

[37] D. K. Vajravelu. Connected Me-Proof of Concept. http://tinyurl.com/gwcsejs, 2013.

[38] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic, and J. Walling. Distinguishing users with capacitive touch communication. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 197–208. ACM, 2012.

[39] T. Wang and T. Blankenship. Projected capacitive touch systems from the controller point of view. *Information Display*, 3(11):8–11, 2011.

[40] W. S. Wang, T. O'Donnell, N. Wang, M. Hayes, B. O'Flynn, and C. O'Mathuna. Design considerations of sub-mW indoor light energy harvesting for wireless sensor systems. *J. Emerg. Technol. Comput. Syst.*, 6(2):6:1–6:26, 2008.

[41] B. Wei, A. Varshney, N. Patwari, W. Hu, T. Voigt, and C. T. Chou. dRTI: Directional radio tomographic imaging. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, pages 166–177. ACM, 2015.

[42] R. Wimmer, M. Kranz, S. Boring, and A. Schmidt. A capacitive sensing toolkit for pervasive activity detection and recognition. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications, 2007.*, pages 171–180, 2007.

[43] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 237–248. ACM, 2014.

[44] P. H.-J. Yoo, S.-J. Song, N. Cho, and H.-J. Kim. Low energy on-body communication for BSN. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, IFMBE Proceedings, pages 15–20. Springer Berlin Heidelberg, 2007. DOI: 10.1007/978-3-540-70994-7_3.

[45] T. G. Zimmerman. Personal area networks: Near-field intrabody communication. *IBM Syst. J.*, 35(3-4):609–617, Sept. 1996.