Devices That Tell On You: The Nike+iPod Sport Kit

T. Scott Saponas, Jonathan Lester, Carl Hartung, Tadayoshi Kohno
Department of Computer Science and Engineering
University of Washington, Seattle, WA, 98195
http://www.cs.washington.edu/research/systems/privacy.html

UW CSE Technical Report 2006-12-06

ABSTRACT

Personal sensing devices are becoming more commonplace in everyday life. Unfortunately, radio transmissions from these devices can create unexpected privacy concerns if not carefully designed. We demonstrate these issues with a widely-available commercial product, the Nike+iPod Sport Kit, which contains a sensor that users put in one of their shoes and a receiver that users attach to their iPod Nanos.

We find and technically explore example scenarios, such as stalking, where the Nike+iPod Sport Kit's design can lead to a compromise of personal privacy and safety. Our results exploit the fact that, when a Nike+iPod user walks or runs, the user's Nike+iPod sensor broadcasts a unique identifier that can be detected up to 60 feet away. We implement a prototype surveillance system that can track people wearing Nike+iPod sensors, plotting their location on a GoogleMaps-based website and emailing and text-messaging real-time surveillance data to an attacker. Our surveillance system can track individuals when they are working out, as well as when they are casually walking and do not have their iPods with them. The smallest node in our real-time surveillance system is currently a miniature gumstix computer (8cm x 2.1cm x 1.3cm). We also develop a method to convert a third-generation iPod into a surveillance device. Using a second-generation Intel Mote and a Microsoft SPOT Watch, we develop the means for an attacker to obtain real-time surveillance data on his or her wrist watch. To counterbalance our attacks, we present simple changes to the Nike+iPod Sport Kit's design that, if implemented, would have significantly improved the kit's resistance to the attacks in this paper. This work suggests a greater need for rigorously evaluating the privacy of new technologies before deployment.

1. INTRODUCTION

As technology continues to advance, more and more computers will permeate our everyday lives; while the last computer revolution placed a single computer in front of a vast majority of our population, the next revolution is poised to place many computers into our environment and onto us. While the many-to-one computational revolution will have many positive aspects, our individual privacy is increasingly endangered by this advancing wave of technological gadgetry.

We study one of the latest such consumer gadgets: the *Nike+iPod Sport Kit* from Apple Computer, Inc. Contained within the $29 (USD) kit are two modules: a *sensor* that a person, Alice, can place in her shoe and a *receiver* which Alice can attach to her *iPod Nano*; see Figures 1 and 2. When Alice walks or runs, the sensor in her shoe *senses* information about Alice's movement and wirelessly transmits this information to the iPod Nano through the receiver. The iPod can then provide Alice with audio feedback about her workout, such as the total distance traveled or calories burned. Although the sensor has an on-off button, the Nike+iPod Sport Kit online documentation [26] recommends that most users should leave their sensors in the on position, and we believe this to be the common case in practice.[1] Similarly, the fourth heading in the same online documentation [26] implies that Apple is concerned about trackability issues; however, we find that their design allows for tracking users via their sensors.[2]

We stress, however, that there is no evidence that Apple or Nike intended for these devices to be used in any malicious manner. Additionally, neither Apple nor Nike endorsed this study.

Privacy, Personal Safety, and the Nike+iPod Sport Kit. Despite broad public awareness of the potential privacy risks associated with pre-existing technologies, like concerns over RFID tags in Gillette razors [24] and library books [25], and despite Apple's apparent awareness that trackability can be undesirable, we find that in the common case the Nike+iPod Sport Kit still fails to offer even the most basic level of user privacy to nearby devices: *a Nike+iPod sensor is an active device that continuously broadcasts a unique identifier when a user is walking or running, even when the user's iPod is not nearby.* Moreover, our results show that, compared to some conventional passive RFIDs, the Nike+iPod Sport Kit significantly lowers the bar for an adversary since (1) the receive range for Nike+iPod sensors is greater than the read range for certain classes of conventional passive RFIDs and (2) it is easy and cheap for an attacker to implement some of our attacks — for example, we show how an attacker could use a third-generation iPod together with a Nike+iPod receiver as a surveillance device.

[1]The exact quotation from [26] is, "Most Nike+iPod runners and walkers can just drop the sensor in their Nike+ shoes and forget about it."

[2]To provide the precise quotation, the fourth heading in [26] reads "Does it [the Nike+iPod Sport Kit] use GPS and does this mean you can track my movements?" The stated answer to this question is a single word, "No."

To make this discussion more concrete, we next consider some example scenarios in which an adversary might exploit the Nike+iPod design for nefarious purposes. These examples show that a failure to provide adequate privacy can lead to a compromise of consumers' personal safety. We defer further details to the body of this paper.

Stalking. A malicious person could exploit the Nike+iPod's design and the sensor's wide broadcast radius for stalking purposes. In our first example scenario, Alice is a college student who regularly wears her Nike+ shoes while walking between home, class, the library, the student union, the gym, and her friends' homes. Her ex-boyfriend, Marvin, is unable to come to terms with their separation and still wishes to have some interaction with her. If Marvin places specially-crafted devices (a.k.a. *nodes* or *Nike+iPod detectors*) by each of the above-mentioned locations, then he can remotely detect exactly when Alice enters and leaves a particular location (by detecting the unique identifier associated with Alice's Nike+iPod sensor). Not only is simply collecting this information a potential violation of Alice's privacy, but this information could also enable Marvin to perform some malicious action. At a minimum, Marvin could somehow "accidentally" find himself bumping into Alice at "random" places, as if by coincidence.

Prototype Surveillance System. We implemented a prototype surveillance system, much like the one Marvin would deploy in the above scenario. Our surveillance system consists of multiple Nike+iPod detector nodes (e.g., a $109 *gumstix* with an attached $79 *wifistix* and a Nike+iPod receiver). When a node detects a broadcasting Nike+iPod sensor, the node knows that the sensor is nearby. The node then sends a message using WiFi to a central database; the message contains the location of the node (latitude and longitude), the four-byte unique identifier for the Nike+iPod sensor, and the time the sensor was detected. The central database aggregates all the data from all the nodes and publishes a GoogleMaps overlay showing the locations at which Nike+iPod sensors were recently detected. By looking at this website, Marvin can learn if Alice is near the library, the gym, and so on.

Extensions and Variations. There are several natural extensions to the above scenario. For example, we implemented an extension allowing Marvin to have the system send him SMS messages or emails when Alice changes her location, thereby providing Marvin with a continuous update of Alice's location. Marvin could also correlate Alice's location information with the location information of others; thereby possibly inferring information about Alice's new boyfriend or other associates. In the case where Alice doesn't own a Nike+iPod kit, Marvin could maliciously implant a sensor in one of Alice's shoes, thereby enabling the above attacks.

Other malicious parties could also use the above system to track large populations of individuals simultaneously. They could look for commuting and socializing habits and single out a particular victim based on his or her profile, e.g., by finding the lone jogger who likes to run at 4am, along with his or her running route. Alternatively, after the stalker physically observes and selects a victim, the stalker could access the database of stored logs and immediately know significantly more information about this particular victim's habits, and perhaps even predict where this victim will be in the next hour and intercept him or her there.

Figure 1: An un-opened Nike+iPod Sport Kit.

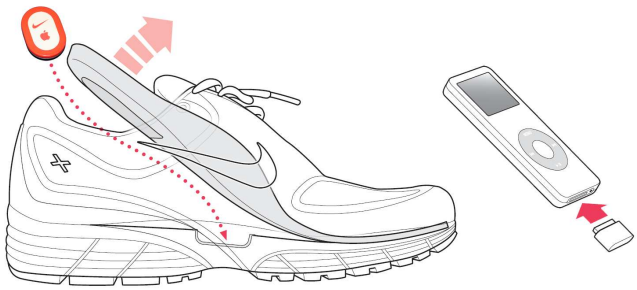

Figure 2: A Nike+iPod sensor in a Nike+ shoe and a Nike+iPod receiver connected to an iPod Nano.

We consider additional scenarios and attack variants in the body of this paper.

Our tools. Toward exploring the potential privacy implications of the Nike+iPod Sport Kit, we developed the following set of attack tools:

- We developed an iPod dock serial-to-USB adaptor, which allows us to plug a Nike+iPod receiver into any device with a USB port.
- We developed a *Nike+iPod Serial Communication Tool* for Windows XP machines. This tool can collect and visually display data about nearby Nike+iPod sensors, and can feed data in real-time to a back-end SQL server as part of a larger surveillance system.
- We created a small Nike+iPod detector (8cm x 2.1cm x 1.3cm) from a $109 gumstix connex 200xm, a $79 wifistix, a $27.50 gumstix breakout board, a $2.95 female iPod dock connector, and a $29 Nike+iPod receiver. This Nike+iPod detector can log data internally and can wirelessly feed data in real-time to a back-end server as part of a larger surveillance system. Without the wifistix, our gumstix Nike+iPod detector can still collect data for offline post-processing.

If others were to make our software (or the equivalent) available on the Internet, then it would require only minimal technical sophistication — the ability to follow online instructions for installing software onto the gumstix and some soldering — for an attacker to create his or her own gumstix-based Nike+iPod detector.

- We created a second small Nike+iPod detector module (5cm x 3.8cm x 2.5cm) using a second-generation Intel Mote (iMote2) running Linux. Our iMote2 module can wirelessly communicate information about nearby Nike+iPod sensors to a paired Microsoft SPOT Watch using bluetooth. An adversary could hide the iMote2 in his or her pocket and observe real-time surveillance data on his or her wrist watch. The iMote2 could also be hidden in an environment and passively record surveillance information for subsequent offline analysis. For example, the iMote2 could be hidden in the bushes near a popular running trail, behind a library book, or inside a restroom paper towel dispenser.

- We also converted a used, third-generation iPod into a Nike+iPod surveillance device. Such iPods are often available on eBay for approximately $100. As with our gumstix-based Nike+iPod detector, creating a third-generation iPod-based surveillance device only requires marginal technical sophistication: an adversary would purchase a Nike+iPod Sport Kit and a few additional parts, would perform a minimal amount of soldering, and would download and install some software that others could make available on the Internet. The converted iPod could serve as a node in some larger surveillance system. Although the iPod's data would not be available to the larger surveillance system in real-time, an adversary could still view real-time surveillance data on the iPod's screen.

- As noted above, we implemented a prototype surveillance system capable of incorporating data from multiple Nike+iPod detectors. The surveillance system can either display a map of real-time Nike+iPod sensor locations, or a historical view of the map. The system can send emails or SMS text messages containing real-time surveillance information. In real-time mode, the current data sources can be Windows XP machines and gumstixs. In historical mode, the current data sources can be Windows XP machines, gumstixs, iMote2s, and third-generation iPods.

We stress that we did not implement our prototype attack systems in order to aid potential stalkers or other adversaries, and we do not plan to distribute our software. Rather, we implemented our systems in order to better understand the capabilities of an attacker and to demonstrate that the attack scenarios that we describe are of practical concern.

While we have used our tools to track ourselves and consenting colleagues, in order to respect the privacy of others, we did not actually deploy our systems to track unsuspecting individuals. Consequently, we do not present the results of a full distributed surveillance experiment in this paper.

Alternate Designs. We consider alternatives to the existing Nike+iPod design that, if implemented, would have significantly improved the privacy-preserving properties of the Nike+iPod kit. While our design alternatives are more privacy-preserving than the current Nike+iPod design, we acknowledge that implementing our alternatives may affect battery life, manufacturing cost, and usability under some circumstances.

Discussion. While the central focus of this study is on understanding, exploiting, and improving the design of the Nike+iPod Sport Kit, the implications of this work are much broader. Namely, while trackability based on personal devices is *not* new — indeed, it is well-known that the possession of traditional passive RFIDs, discoverable bluetooth devices, and WiFi devices can enable tracking by third parties — the key contribution here is showing that new devices are still being introduced without strong privacy-preserving mechanisms. Our hope is that this work will further motivate industry members and the computer science research community to work together to better understand and address the full privacy implications of future devices, as well as to work towards retroactively improving the privacy of existing technologies.

Overview. Section 2 discusses our initial technical exploration into the design of the Nike+iPod system. Next, Section 3 discusses our experiments measuring the various characteristics of the Nike+iPod sensors. We then describe how we instrumented tools for creating surveillance systems using the Nike+iPod receiver in Section 4. Section 5 gives example scenarios where attackers could use these system to stalk victims, and we discuss the implications of our work in Section 6. Section 7 discusses related work. Finally, Section 8 concludes the paper.

2. DISCOVERING THE NIKE + IPOD PROTOCOL

The Nike+iPod Sport Kit. The Nike+iPod Sport Kit allows runners and walkers to hear real time workout progress reports on their iPod Nanos and to view their workouts online at `http://www.nike.com/nikeplus/`. A typical user would purchase an iPod Nano, a Nike+iPod Sport Kit, and either a pair of Nike+ shoes or a special pouch to attach to non-Nike+ shoes. The Nike+iPod kit costs $29 and consists of a *receiver* and a *sensor*; see Figure 1. Users place the sensor from the kit in their left Nike+ shoes and attach the receiver to their iPod Nanos as shown in Figure 2.

The sensor is a 3.5cm x 2.5cm x 0.75cm plastic encased device, and the receiver is a 2.5cm x 2cm x 0.5cm plastic encased device. When a person runs or walks the sensor begins to broadcasts sensor data via a radio transmitter whether or not an iPod Nano is present. When the person stops running or walking for ten seconds, the sensor goes to sleep. When the iPod Nano is in *workout mode* and the receiver's radio receives sensor data from the sensor, the receiver will relay (a function of) that data to the iPod Nano, which will then give feedback to the person on his or her workout.

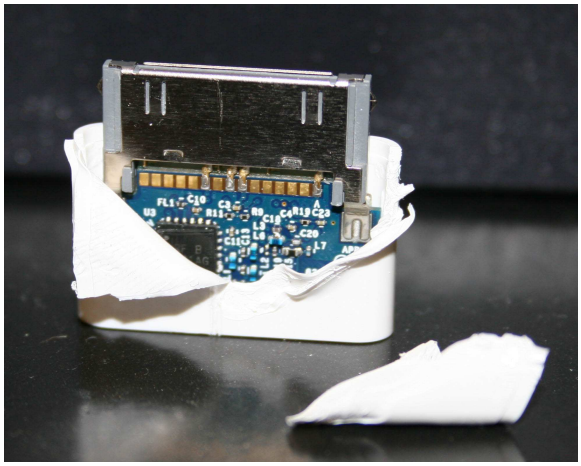

Figure 3: A broken-open Nike+iPod receiver.

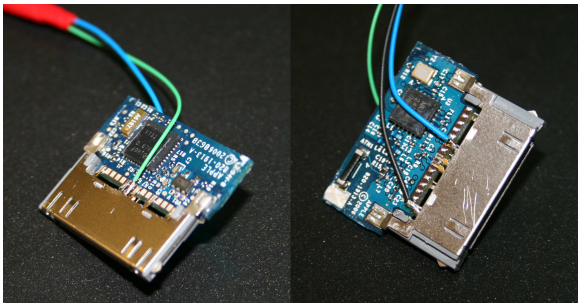

Figure 4: A Nike+iPod receiver fully removed from its protective case. The blue wire is attached to the iTXD pin, the green wire to the iRXD pin, and the black wire is attached to ground.

The iPod software offers a variety of workout modes: *Basic*, *Time*, *Distance*, and *Calories*. The Basic mode allows one to select music to listen to during a workout and monitors distance, running pace, and calories burned. At anytime during the workout, the user may press the center button of the iPod and spoken feedback over the headphones announces how much time has elapsed since beginning the workout, the distance run so far, and the current pace (in terms of minutes per mile or km). The Time, Distance, and Calories modes are similar to the Basic mode, except they allow the user to set a target workout duration, distance to run, or calories to burn, respectively. At the conclusion of a workout users sync their iPod with iTunes and the workout information is sent to `NikePlus.com`. The NikePlus web site gives users several visualizations of their workouts, the ability to challenge others to workout competitions, as well as a forum for discussing running.

Initial Analysis. Our first goal was to learn how the Nike+iPod sensor communicates with the receiver. According to the Nike+iPod documentation, a sensor and receiver need to be *linked* together before use; this linking process involves user participation. Once linked, the receiver will only report data from that specific sensor, eliminating readings from other users' sensors. The receiver can also remember the last sensor to which it was linked so that users do not need to perform the linking step every time they turn on their iPods. The receiver can also later be linked to a different sensor (for a replacement sensor or different user), but under the standard user interface the receiver can only be linked to one sensor at any given time.

We observed, however, that a single sensor could be linked to two receivers simultaneously, meaning that two people could use their iPod Nanos and the standard user interface to read the data from a single Nike+iPod sensor at the same time. Further investigation revealed that the sensor was a transmitter only, meaning that it was incapable of knowing what iPod or receiver it was associated with. This observation provides the underlying foundation for our work since it concretely shows that a Nike+iPod Sport Kit does not enforce a strong, exclusive, one-to-one binding between a sensor and a receiver. Having made this observation, we then commenced to uncover more details about the Nike+iPod protocol.

The Hardware. The Nike+iPod Sport Kit receiver communicates with the iPod Nano through the standard iPod connector. Examining which pins are present on the receiver's connector and comparing those pins with online third-party pin documentation [17], we determined communication was most likely being done over a serial connection.

Opening the white plastic case of the receiver reveals a component board and the pin connections to the iPod connector. There are ten pins in use; three of these pins are used in serial communication: ground, iPod transmit (iTXD), and iPod receive (iRXD); see Figures 3 and 4. We verified that digital data was being sent across this serial connection by connecting an oscilloscope over the iRXD and ground while the open receiver was connected to the iPod. This also allowed us to measure the bit width and establish that the serial connection was using the data rate of 57.6 Kbps. We then soldered wires onto the ground, iTXD, and iRXD pins and connected them to the serial port of our computer.

With the receiver connected to the iPod we turned on the iPod and saw data sent in both directions over the serial connection. In the data transmitted by the iPod, the serial number of the iPod is sent in ASCII. Similarly, in the data transmitted by the receiver, the receiver's serial number and the serial number of the last sensor that was used in a workout with the receiver is sent in ASCII.

Serial Communications. As noted above, before the receiver can be used with a new sensor, the sensor must be *linked* with the receiver. This is initiated by the user through menus in the iPod interface. The user is asked to walk around so that the sensor can be detected by the receiver. When the link process is started, the iPod sends some data to the receiver. Then, the receiver begins sending data until the new sensor is discovered and linked by the receiver. Finally, the iPod sends some more data back to the receiver. In this last chunk of data from the iPod, the serial number of the new sensor is sent in ASCII. A transcript of the communications is show in Figure 5.

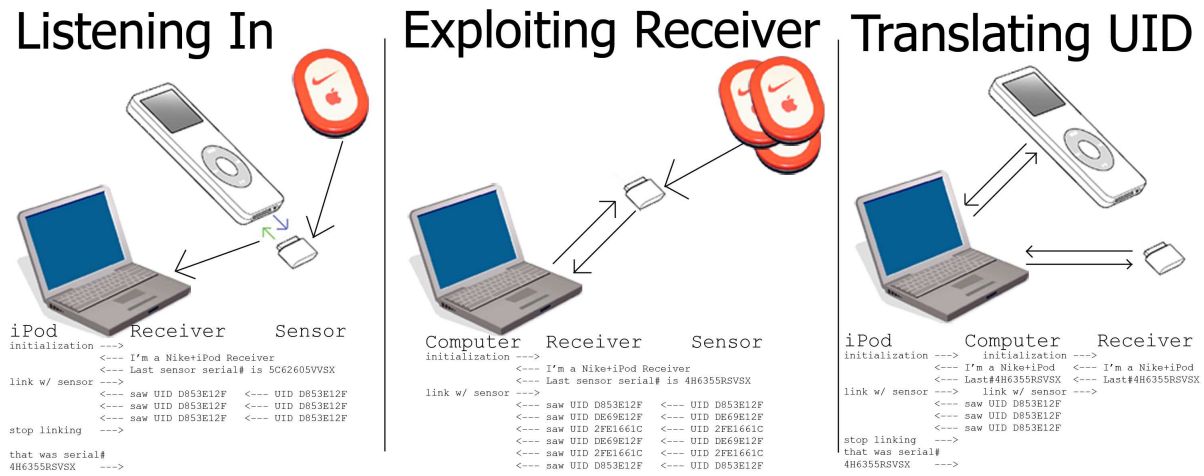

Figure 5: The figure on the left shows our approach for passively monitoring the serial communications between an iPod and the Nike+iPod receiver; the communications between the iPod and the receiver are over a physical, serial connection, and the communication from the sensor to the receiver is via a radio. The figure in the middle shows our approach for directly controlling a Nike+iPod receiver from a computer; the communication from the computer to the Nike+iPod receiver is over a physical serial connection. The figure on the right shows our approach for translating between a sensor's UID and the sensor's serial number.

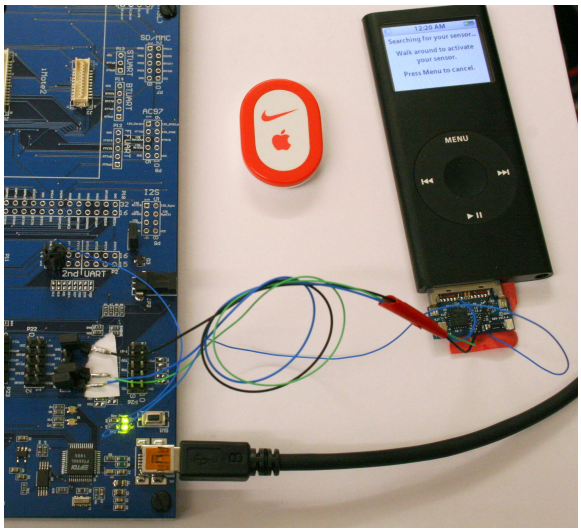

Figure 6: Our setup for deriving a serial number from a UID.

Online third party resources document the iPod accessory serial packet format to be a three byte header, a payload, and a one byte checksum [14]. The first two bytes of the header are `FF55` and the third byte is the size of the payload in bytes. We also found that sometimes there are additional `00` bytes between the `FF` and `55` bytes in the header.

After comparing several traces of the link process with several different sensors we noticed that linking seemed to complete when the third occurrence of a certain packet came from the receiver. These packets' payload started with the four bytes `090D0D01`; however, the next four bytes were different depending on which sensor we used (for example `37625122`). We assumed this to be a unique identifier per sensor that the iPod Nano software somehow maps to the ASCII serial number of the sensor in order to send the serial number back to the receiver after linking. We refer to these four bytes as the sensor's *UID*.

Verifying UID to Serial Number Translation. We attempted to verify that the iPod translates these identifiers to the sensor's serial number by having our computer act as a man-in-the-middle between the receiver and the iPod Nano. We further modified the receiver by disconnecting the iRXD pin from the receiver board. We then connected a second serial port from our computer between the receiver and the iRXD pin. This allowed us to listen to iTXD with our first serial port, listen to the receiver with our second serial port, and repeat what the receiver sends to the iPod via iTXD using the transmit on the second serial port. See Figure 6. Using this new configuration, we again tried the link process; except this time, instead of walking around with a sensor, we sent to the iPod a packet consisting of the bytes `090D0D01`, the four bytes corresponding to the UID of one of our sensors, and a string of zeros to pad the payload to the appropriate length. The iPod returned the correct serial number for this sensor. This method therefore appears to allow one to translate the four-byte identifiers seen in the packets from the receiver to actual sensor serial numbers.

Since UIDs appear unique, and since there is a straightforward way to exploit an existing iPod Nano to map from UIDs to serial numbers, we will hereafter focus only on the UIDs. While we suspect that most adversaries will never need to map from UIDs to serial numbers, if an adversary wishes to do so, the adversary could re-implement the steps we describe above.

Controlling the Nike+iPod Receivers Directly. Our next step was to use the Nike+iPod receiver to listen for sensor identifiers in an automated fashion *without* the iPod Nano. To do this we modified an iPod female connector by soldering wires from the serial pins on the iPod connector to our adapter, adjusted the voltage accordingly, and attached 3.3V power to the power pin. We then plugged an unmodified Nike+iPod receiver into our female connector and replayed the data that we saw coming from the iPod when the iPod is turned on and then when the iPod enters link mode. This process caused the receiver to start sending packets over the serial connection to our computer with the identifiers of the broadcasting sensors in range. However, because our computer never responds to the receiver's packets, the link process never ends and the receiver continues to send to our computer the identifiers of transmitting sensors until power is removed.

3. MEASUREMENTS

In this section we discuss our preliminary measurements of: when a sensor transmits; how often it transmits; the range at which the receiver hears the sensor; and the collision behavior of multiple sensors. For this exploration, we are only interested in the unique identifier transmitted by each sensor. We do not investigate the "payload" of the sensor packets.

As described in the documentation of the Nike+iPod Sport Kit, when the sensor is still, it is "sleeping" to save battery. When one begins to walk or run with the sensor in their shoe, the sensor begins transmitting. It is also possible to wake up the sensor without putting it in a shoe. For example, shaking the sensor while still in the sealed package from the store will cause it to transmit its UID. Sensors can also be awakened by tapping them against a hard surface or shaking them sharply. Similarly, if a sensor is in the pocket of one's pants, backpack, or purse, it will wake up occasionally. Once walking, running, shaking, and the likes ceases, the sensor goes to sleep after approximately ten seconds.

While the sensor is awake and nearby we observed it transmit one packet every second (containing the UID). When the sensor is more distant or around a corner the receiver heard packets intermittently, but still on second intervals. When multiple sensors are awake near one another some packets get corrupted (their checksums do not match). As the number of awake sensors increase so does the number of corrupt packet. However, our tests with seven sensors indicated the receiver still hears every sensor UID at least once in a ten second window.

From examining the Nike+iPod receiver's components, as shown in Figure 4, we surmise that the Nike+iPod Sport Kit uses the ANT wireless radio and protocol. ANT radios generally have a range of 1–30 meters [1]. During our experiments with the Nike+iPod sensors we observed approximately a 10 meter range indoors and a 10–20 meter range outdoors. Sensors are also detectable while moving quickly. Running by a receiver at approximately 10 MPH, the sensor is reliably received. Driving by someone walking with a sensor in their shoe, the sensor can be reliably detected at 30 MPH. We have not tested faster speeds.

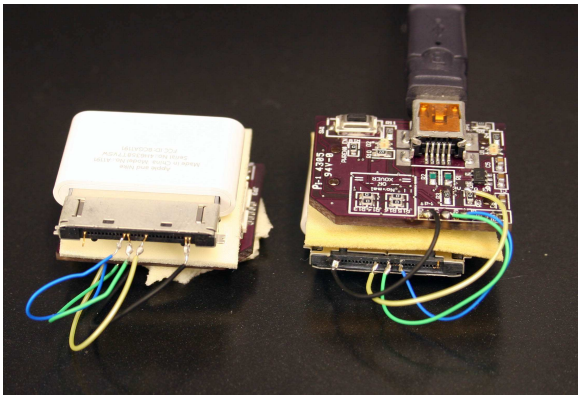

Figure 7: Our Nike+iPod receiver to USB adaptor.

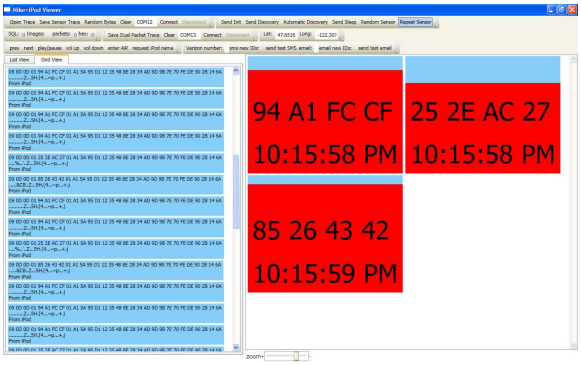

Figure 8: A screenshot of our Nike+iPod Serial Communication Tool.

When someone is engaged in a workout with a sensor using a receiver attached to an iPod, a second receiver can detect the sensor transmitting its UID. This situation is natural since the sensor can only transmit information; the sensor does not have receive capabilities.

4. INSTRUMENTING ATTACKS

We now describe our systems for exploiting the design of the Nike+iPod Sport Kit.

4.1 Receiver to USB Adaptor

We created a compact USB receiver module for detecting Nike+iPod sensor UIDs. Our module does not require any modification to the Nike+iPod receiver; see Figure 7. Our USB module consists of a female iPod connector [15] and a serial-to-USB board utilizing the FTDI FT2232C chipset [7]. We connected the three serial pins and the 3.3V power pin of the iPod connector to the appropriate pins of the FT2232C. When this module is connected to a computer via USB, the receiver is then powered and a USB serial port is made available for our software to communicate with the receiver. With the receiver attached, this package is approximately 3cm x 3cm x 2cm.

4.2 Nike+iPod Serial Communication Tool

Our Nike+iPod Serial Communication Tool provides support for: logging serial traces on up to two serial ports simultaneously; sending Nike+iPod receiver initialization and

link commands; logging Nike+iPod receiver serial data at the packet level (including checksum verification); and logging what sensors have been seen and when. See Figure 8. Logs for multiple serial port traces are interleaved so that one may see the data exchange or protocol of two devices (in our case, the iPod and the receiver).

Our tool also provides a graphical interface for: sending and receiving binary data in hex format over up to two serial ports; viewing hex and ASCII representation of incoming packets; and viewing what sensors have been seen and when new packets for those sensors arrive. Our sensor visualization consists of a blue rectangle for each seen sensor with its UID in hex and the last time it was seen. Each time the sensor is seen the box for that sensor becomes red and slowly becomes blue (from top to bottom) over the following five seconds. This allows one to get a sense of which awake sensors are in range and how many of their packets are arriving uncorrupted.

Optionally, our tool can take a picture whenever a new sensor is discovered using most USB cameras and make that photo the background of the blue box in the sensor visualization. This application can also serve as a data collection node in a larger surveillance network. To support this task, the tool can upload to a SQL server sensor events including UID, optional photo, timestamp, latitude, and longitude. (Latitude and longitude are currently set manually by the user; one could, however, imagine linking a bluetooth GPS receiver so that a mobile receiver on a car or bike could do accurate data collection.)

The tool can also SMS or email sensor information to users. This tool is implemented in approximately 2000 lines of C# and XAML on Microsoft .NET 3.0 for Microsoft Windows XP or later.

4.3 Intel Motes

In addition to our serial communication tool for Windows we have also created an embedded module for logging and tracking Nike+iPod sensors using version 2 of the Intel Motes (iMote2). This module consists of an iMote2, an unmodified Nike+iPod receiver, female iPod connector, and an iMote2 utility daughter board with bluetooth. The assembled package is 5cm x 3.8cm x 2.5cm, weighs 2.3 ounces, and has a storage capacity of 2GB via a Mini Secure Digital card.

The iMote2 runs the Linux operating system and our software is written in C. The iMote2 communicates with the receiver using a serial port. On boot, the iMote2 sends initialization and link commands to the receiver and begins logging sensor events to a file. Optionally, the software can turn on an LED when the iMote2 detects that one or more prespecified sensors are nearby. The set of target sensors is specified in a configuration file. One can imagine using the LED-based alarm as a discrete mechanism for visually notifying a user when a target victim's sensor is nearby. There are obvious audio (buzzer) and physical (vibrate) extensions.

We have instrumented our iMote2 to communicate the UIDs of sensors in range to a Microsoft SPOT Watch over bluetooth. Using our system, an adversary could put the iMote2 and receiver in his or her backpack, purse, or pocket, and still continuously monitor information about nearby sensors on his or her watch. See Figure 9.

Figure 9: A Microsoft SPOT Watch receiving Nike+iPod sensor IDs from an iMote2 over wireless bluetooth.

The sensor events logged to file can also be manually uploaded to a central server for aggregation with sensor information from the Serial Communication Tool. A straightforward extension to this device would be for the iMote2 to obtain real-time location information from a bluetooth GPS sensor; this would enable an attacker to collect accurate sensor location information while the attacker is mobile. Another straightforward extension would be to create a large, distributed surveillance sensor network consisting of multiple iMote2 nodes, and to upload surveillance data in real-time to some central SQL server. Rather than implement this latter capability in our iMote2s, we do so with our gumstix in Section 4.4.

4.4 Gumstixs

We have also implemented a cheap Nike+iPod surveillance device using the Linux-based gumstix computers. This module consists of an unmodified $29 Nike+iPod receiver, a $109 gumstix connex 200xm motherboard, a $79 wifistix, a $27.50 gumstix breakout board, and a $2.95 female iPod connector. The assembled package is 8cm x 2.1cm x 1.3cm and weighs 1.1 ounces; see Figure 10.

Our gumstix-based module runs the same surveillance software that our iMote2s run, except that (1) the software on the gumstix module uses WiFi to wirelessly transmit real-time surveillance data to a centralized back-end server and (2) our gumstixs do not pair with a Microsoft SPOT Watch. The real-time reporting capability allows the gumstix module to be part of a larger real-time surveillance system. If an

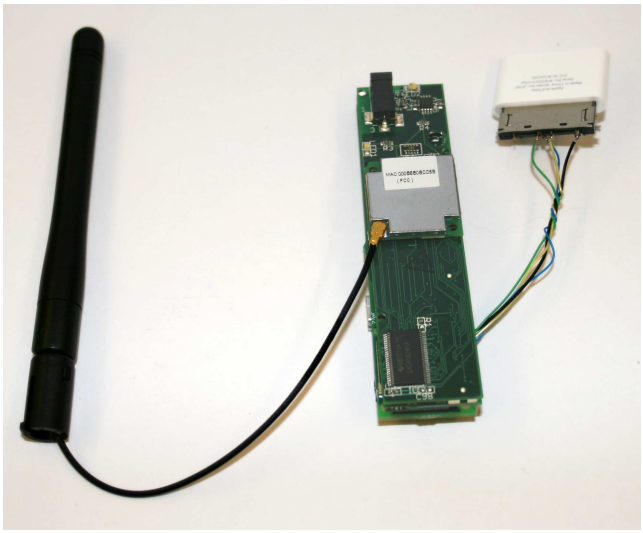

Figure 10: A gumstix-based Nike+iPod surveillance device with WiFi wireless capabilities.

adversary does not need this real-time capability, then the adversary can reduce the cost of this module by omitting the wifistix.

4.5 Exploiting the iPod

Using off the shelf hardware, we created another surveillance device that requires little hardware modification to log and view Nike+iPod sensor activity. We used a desktop computer to install iPod Linux [18] on a third-generation iPod. We then recompiled our iMote2 logging software using the iPod Linux toolchain. The only hardware modification is that the serial send and receive lines on the remote control port at the top of the iPod must be connected to the serial lines at the bottom of the iPod in the dock connector. This can be achieved with a dock connector break-out board [16]. The breakout board allows one to plug in the Nike+iPod receiver (or any iPod accessory) to the iPod though the break-out board while exposing all of the connector pins, including serial send and receive, for soldering.

Running our application under iPod Linux with the receiver plugged into the iPod, we can now display on the screen of the iPod what sensors are nearby and log sensor events (UID and timestamp) to the harddrive of the iPod for later synchronization with a central database. This allows an attacker to use an older third-generation iPod as a surveillance device; the older iPod can be obtained at a discount from places like eBay. A natural extension of this application would be to use a text-to-speech software package so that one can wear headphones connected to the iPod and be notified by audio of what Nike+iPod sensors or people are nearby.

If the iPod Linux community figures out how to use the serial port in the dock connector of the third-generation iPod, or any other iPod with a dock connector running iPod Linux, an attacker could track Nike+iPod sensors without any hardware modification at all.

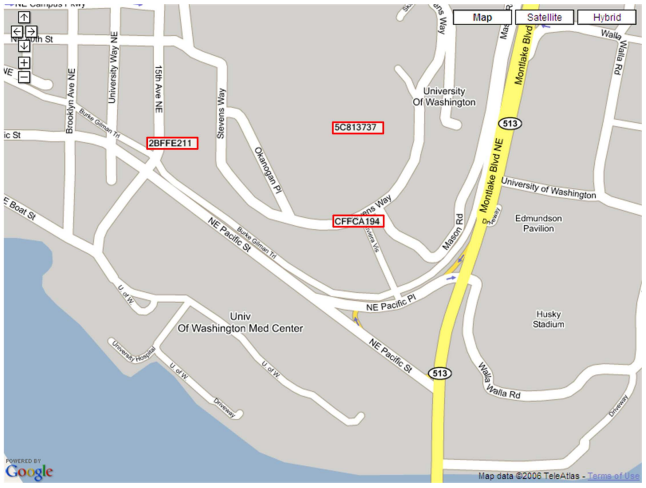

Figure 11: A screenshot of our GoogleMaps-based surveillance web application.

4.6 A Distributed Surveillance System

To illustrate the power of aggregating sensor information from multiple physical locations, we created a GoogleMaps-based web application. Our web application uses and displays the sensor event data uploaded to a central SQL server from multiple data sources. Currently, the data sources may be our Serial Communication Tool, iMote2 application, gumstix application, or iPod Linux application. See Figure 11.

In real-time mode, sensors' UIDs are overlayed in hex on a GoogleMaps map at the location the sensor is seen. When the sensor is no longer present at that location, the UID disappears. Optionally, digital pictures taken by a laptop when the sensor is first seen can be overlayed instead of the UID. Currently, the Serial Communication Tool and the gumstixs can upload data in real-time, but it is also possible to create a network of multiple iMote2 nodes that are also capable of uploading data to the server in real-time. In history mode, the web application allows the user to select a timespan and show all sensors recorded in that timespan. For example, one could select the timespan between noon and 6pm on a given day; all sensors seen that afternoon will be overlayed on the map at the appropriate location.

This application would allow many individuals to track people of interest. An attacker might also use this tool to establish patterns of presence. If many attackers with receivers cooperated, this software and website would allow the tracking and correlation of many people with sensors.

5. EXAMPLE SCENARIOS

Having described our basic attack tools in Section 4, we are now in a position to discuss how one might use or refine our attack tools for particular surveillance and adversarial applications. We do not intend for the following to be an exhaustive list of all possible attack scenarios. Rather, the goal of the following discussion is to highlight the breadth of the types of scenarios that exist.

While the attacker may choose from any of our technologies (a laptop running the Serial Communication Tool, an iMote2, a gumstix, or a third-generation iPod itself), we refer to the adversary's equipment generically as a *Nike+iPod detector*.

The Jealous Boyfriend. Marvin is a jealous boyfriend who suspects that his girlfriend, Alice, is cheating on him with his best friend Bob. Alice wears Nike+ shoes and uses a Nike+iPod Sport Kit. We assume that Marvin knows the UID of the Nike+iPod sensor in Alice's shoe; Marvin could easily learn this UID by, for example, shaking Alice's shoe in front of a Nike+iPod detector or by turning his Nike+iPod detector on while walking Alice to her car. Alternately, suppose that, unbeknownst to Alice, Marvin maliciously implants a Nike+iPod sensor in one of Alice's shoes, or hides a sensor in Alice's jacket or purse.

If Marvin is able to place a Nike+iPod detector near Bob's house, then Marvin will be able to infer whether Alice visited Bob, and for how long. We stress that, rather than deploy our full prototype surveillance system, for this application Marvin may only need a single Windows XP laptop or palmtop running our Serial Communication Tool, a single Intel Mote, a single gumstix, or a single third-generation iPod.

If Marvin knows that Bob also regularly wears his Nike+ shoes, and if Marvin knows the UID of Bob's Nike+iPod sensor, then Marvin could place a Nike+iPod detector near Alice's home, thereby allowing Marvin to infer how regularly Bob (or anyone else with Nike+ shoes and a Nike+iPod sensor) visits Alice. As a final twist to this example, Marvin could also deploy a Nike+iPod sensor on Alice's preferred jogging path, with the goal of inferring whether Alice is starting to acquire a new jogging partner.

While we leave to the reader's imagination what Marvin might do with the information he gains from implementing any of these attacks, it should be clear that these attacks compromise Alice's privacy.

The Ex-Boyfriend. After realizing that Marvin is no good for her, Alice has chosen to terminate their relationship. Unable to handle rejection, Marvin is committed to finding ways to keep bumping into Alice.

In order to track Alice, Marvin might deploy a distributed surveillance system like the one we describe in Section 4.6. Namely, Marvin might place Nike+iPod detectors at key locations on campus, including in front of the library, the gym, the dorms, and the student union, as well as at key locations off campus, such as Alice's place of employment, and Alice's home. By viewing the GoogleMaps-based website, Marvin can at any time detect whether Alice is currently near one of his deployed Nike+iPod detectors or when she was last detected. Marvin can then "coincidentally" find himself bumping into Alice near that location. Rather than finding Alice's location by looking at the GoogleMaps-based website, Marvin could also configure the surveillance system to send him an email message or an SMS text message when Alice enters the receive range of a surveillance node.

Marvin could also use our GoogleMaps-based surveillance system to learn the UIDs of the Nike+iPod sensors that are consistently near Alice, thereby learning information about Alice's peers. Marvin could also exploit another feature of our surveillance system and look at historical views of the map, thereby allowing Marvin to develop a more complete understanding of Alice's daily schedules over time.

The Stalker. We refer the reader to Section 1 for further discussions about stalking and, in particular, for a discussion of how a stalker might exploit a distributed surveillance system.

The Professional Thief. Unlike Marvin, who targets only a signal individual, the professional thief might be interested in targeting *any* person bearing a certain profile or with certain schedules, such as a person who is not at home during certain times of the day. Currently, a thief might visually case homes for robbery; however, doing so is time consuming and conspicuous. Unfortunately, a Nike+iPod-based distributed surveillance system, like the one we describe in Section 4.6, would enable a professional thief to monitor many people simultaneously, until he determines which victim or victims to focus on.

The Unethical Organization. An unethical organization could use a distributed Nike+iPod-based surveillance system to track their members, or the members of a competitor organization. As an example of the former, an organization might employ a Nike+iPod-based surveillance system to determine whether its members attend the other organization's rallies, events, or offices. As an example of the latter, rather than hire a large number of private investigators, the first organization could place Nike+iPod detectors near all of its competitor's homes, as well as near questionable venues on the wrong side of town. After cheaply performing such a broad initial surveillance step, the first organization could now hire one or two private investigators to follow the individuals that likely participate in blackmailable activities.

Customer Tracking. Companies could use the Nike+iPod sensors to track customers throughout their stores and over multiple visits. For example, a store could create a binding between the consumer's Nike+iPod sensor's UID and the customer's purchasing history (as derived through loyalty programs or credit card numbers). Then, the next time the customer enters the store, the store attendants could immediately know the types of purchases the customer might be interested in making and target the sales pitch accordingly.

As background, we remark that there have been several high-profile examples of stores (neither Apple nor Nike) violating customers' privacy by embedding RFID tags in consumer products and then observing customer handling those tagged objects [8, 24, 29].

Muggers. Since iPod Nanos are attractive to thieves, and since owners of iPod Nanos might have other attractive gadgets on them, muggers could benefit from knowing whether a potential victim might be in possession of an iPod Nano. Indeed, following a string of muggings in Manhattan tar-

geting owners of iPods, New York University students were advised to not wear the tell-tale white ear-buds commonly associated with iPods [2, 9].

Our results show that even if a person does not wear the tell-tale ear-buds, a mugger can detect whether a person might be in possession of an iPod by deploying a Nike+iPod detector (under the assumption that if a person is wearing a shoe with a Nike+iPod sensor, then that person might be in possession of an iPod, even if he or she is not actively using it). In more detail, the mugger could use a generic Nike+iPod detector to determine whether a Nike+iPod sensor is nearby. If there are multiple people in the vicinity, the mugger could localize on the particular victim by using a directional antenna.

While one might initially suspect that the above attack may be beyond the technical capabilities of most muggers, we disagree. First, although cumbersome for the mugger, the mugger could actually implement a crude version of this attack using a standard iPod Nano, a standard Nike+iPod receiver, and the standard user interface and linking process to detect whether a Nike+iPod sensor is nearby. Second, we believe that some of our methods for automating an attack require only minimal technical sophistication, provided that someone else makes the software for the attack available on the Internet. Moreover, we observe that a group of criminals working together would only need a single individual with the appropriate technical capabilities, or the means to purchase those capabilities. Lastly, if in the future there are many more Nike+iPod-like devices that fail to provide strong privacy properties, there may be additional motivation for criminals to acquire the appropriate gadget-detecting technologies.

Combining Tracking Technologies. It is possible to combine our Nike+iPod based surveillance techniques with other techniques, thereby creating more complete profiles of people and further eroding their privacy. For example, it is possible to incorporate the surveillance of discoverable bluetooth devices into our prototype surveillance system. By binding Nike+iPod sensors to bluetooth devices, an adversary can now track individuals even if they are only carrying one of those devices. Perhaps, even worse, the sometimes discoverable bluetooth devices could give a meaningful name to the often transmitting Nike+iPod sensor. While our prototype Serial Communications Tool is capable of taking photographs when it detects a sensor enter into range, one could employ more sophisticated computer vision, as well as face, gait, or license plate recognition techniques, to extract further information from these pictures and build broader profiles of individuals. If an attacker can place his or her device close enough to an individual, an attacker could also bind a person's Nike+iPod sensor data to the individual's RFID credit card information [11], passport information [20], or library book information [25].

6. DISCUSSION

One of the key contributions of this work is to highlight the fact that, despite broad public awareness of the privacy concerns with RFID tags and discoverable bluetooth devices, and even despite some industry awareness of the importance of untrackability, major companies are still introducing popular new technologies without strong privacy guards.

We consider this situation unfortunate since in many cases it is technically possible to significantly improve consumer privacy. Consider, for example, the typical usage scenario for the Nike+iPod Sport Kit. In the common case, we expect that once a user purchases a Nike+iPod Sport Kit, he or she will rarely use the sensor from that kit with the receiver from a different kit. This means that the sensor and the receiver could have been pre-programmed at the factory with a shared secret cryptographic key. By having the sensor encrypt each broadcast message with this shared key, the Nike+iPod designers could have addressed most of our privacy concerns about the Nike+iPod application protocol; there may still be information leakage through the underlying radio hardware, which could be dealt with separately. If Apple and Nike decide that a sensor from one kit should be used with the receiver from a separate kit, then several options still remain. For example, under the assumption that one will only rarely want to use a sensor from one kit with a receiver from another, the cryptographic key could be written on the backs of the sensors, and a user could manually enter that key into their iPods or computers before using that new sensor. Alternately, the sensor could have a special button on it that, when pressed, causes the sensor to actually broadcasts a cryptographic key for some short duration of time.

In a bit more technical detail, assume that both the sensor and the receiver in a Nike+iPod Sport Kit are pre-programmed with the same shared 128-bit cryptographic key K. One design approach would be for the sensor to pre-generate a new pseudorandom 128-bit value X during the one-second idle time between broadcasts. Although the sensor could generate X using physical processes, we suggest generating X by using AES in CTR mode with a second, non-shared 128-bit AES key K'. Also during this one-second idle time between broadcast, the sensor could pre-generate a keystream S using AES in CTR mode, this time with the initial counter X and the shared key K. Finally, when the sensor wishes to send a message M to the corresponding receiver, the sensor would actually send the pair $(X, M \oplus S)$, where "$\oplus$" denotes the exclusive-or operation. Upon receiving a message (X, Y), the receiver would re-generate S from X and the shared key K, recover M as $Y \oplus S$, and then accept M as coming from the paired sensor if M contains the desired UID. While it is rather straightforward to argue that this construction provides privacy at the application level against passive adversaries (by leveraging Bellare *et al.*'s [3] provable security results for CTR mode encryption), we do acknowledge that this construction may not fully provide all desired target security properties against active adversaries. Also, there may be some identifying information transmitted by the radio hardware in the Nike+iPod sensor. Furthermore, we acknowledge that there are ways of optimizing the approach outlined above, and that the above approach may affect the battery life, manufacturing costs, and usability of the Nike+iPod Sport Kit. Nevertheless, this discussion clearly shows that it is possible to significantly improve upon the privacy properties of the current Nike+iPod Sport Kits.

Other Devices. The reader might ask why it is important for new technologies to protect consumer privacy when existing technologies, like traditional RFIDs and discoverable bluetooth devices, may not. Our opinion is that if one adopts such a belief system, then one has already conceded to an eventual loss of control of one's personal privacy. We believe that protecting consumer privacy is critical, as exemplified in part by the example scenarios in Section 5. Therefore, rather than concede defeat, our hope is that the computer science research community and industry partners will work even more closely together to proactively understand and address the privacy of portable consumer products, both by ensuring that new technologies do not further erode our privacy, and by working to retroactively improve the privacy of existing technologies.

Is an On-Off Switch Enough? Another natural question to ask is whether a sufficient privacy-protection mechanism might simply be to place on-off switches directly on all mobile personal devices, like the Nike+iPod Sport Kit sensors. We do not believe this proposal to be sufficient for several reasons. First, this approach by itself will not protect consumers' privacy while the devices are in operation. Second, we believe that it is unrealistic to assume that most users will actually turn their devices off when not in use, especially as the number of such personal devices increases over time. To further support our belief, we quote from the Nike+iPod online documentation: "most Nike+iPod runners and walkers can just drop the sensor in their Nike+ shoes and forget about it [26]." This quote suggests that Apple and Nike have already realized that, given the choice between simplicity (not using the on-off switch) and cost (not consuming energy from the battery when not working out), consumers would choose simplicity. Unfortunately, making or following this recommendation also favors simplicity over privacy and personal safety. (As a slight aside, we remark that assuming that all users will turn off all their mobile devices when not in use is somewhat akin to assuming that all users will choose strong passwords without external prompting, an assumption which folklore knowledge suggests to be unreasonable in practice.)

7. RELATED WORK

There is an immense body of related work; we only provide a brief survey here. Most closely related to this paper are the research results highlighting potential privacy concerns with RFID tags (e.g., [20, 25]) and discoverable bluetooth devices (e.g., [19]), as well as many popular press articles on the subject (e.g., [8, 24, 29]). Moreover, others have created bluetooth- and WiFi-based tracking systems for research, demonstration, or commercial purposes (e.g., [4, 5, 6, 10, 23, 27, 28]). We are, however, unaware of any other location-based surveillance system that goes as far as plotting subjects' locations on a map in real-time. Further afield, there has been significant research on face recognition and gait recognition; see [12] for a survey.

On the defensive side, researchers have proposed a *blocker tag* for helping protect the privacy of RFID devices [21], as well as algorithmic changes to portions of the RFID communication protocols (e.g., [22, 25]) and other wireless protocols (e.g., [13, 30]). Despite these advances, and although it is possible to significantly improve the privacy of some devices (e.g., our changes to the Nike+iPod protocol in Section 6), there are still fundamental open research questions to address.

8. CONCLUSIONS

As personal sensing devices begin to pervade our daily lives, it becomes increasingly important for our gadgets to preserve privacy. Despite historic consumer concerns and previous literature on potential privacy issues, our results show that companies continue to release devices that do not provide strong privacy guarantees. Since privacy failures with these devices can have serious consequences, including compromises to personal safety, we hope that this work will motivate industry and research partners to further pursue proactive measures of ensuring consumer privacy.

9. ACKNOWLEDGEMENTS

We thank Yaw Anokwa, Kate Everitt, Kevin Fu, J. Alex Halderman, Ed Lazowska, David Molnar, Lincoln Ritter, Avi Rubin, Jason Schultz, Adam Stubblefield, Dan Wallach, and David Wetherall.

10. REFERENCES

[1] ANT comparison sheet. `http://www.thisisant.com/index.php?section=36`.

[2] AppleInsider. NYU Warns Students Against Wearing iPod Earbuds, 2005. `http://www.appleinsider.com/article.php?id=930`.

[3] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 394–403. IEEE Computer Society Press, 1997.

[4] BlueTags to install world's first Bluetooth tracking system, 2003. `http://www.geekzone.co.nz/content.asp?contentid=1070`.

[5] Braces – A Bluetooth Tracking Utility. `http://braces.shmoo.com/`.

[6] J. Collins. Lost and Found in Legoland, RFID Journal, 2004. `http://www.rfidjournal.com/article/articleview/921/1/1/`.

[7] FT2232C Dual USB UARF/FIFO IC. `http://www.ftdichip.com/Products/FT2232C.htm`.

[8] A. Gilbert. 'Secret' RFID test draws consumer ire, ZDNet.co.uk, 2003. `http://news.zdnet.co.uk/emergingtech/0,1000000183,39117924,00.htm`.

[9] gothamist. Gang of iPod Thieves Arrested, 2005. `http://www.gothamist.com/archives/2005/09/09/gang_of_ipod_thieves_arrested.php`.

[10] M. Haase and M. Handy. BlueTrack – Imperceptible tracking of bluetooth devices. In *Ubicomp Poster Proceedings*, 2004.

[11] T. S. Heydt-Benjamin, D. V. Bailey, K. Fu, A. Juels, and T. O'Hare. Vulnerabilities in first-generation RFID-enabled credit cards, 2006. Manuscript.

[12] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. In *IEEE Transactions on Systems, Man, and Cybernetics*, August 2004.

[13] Y.-C. Hu and H. J. Wang. A framework for location privacy in wireless networks. In *ACM SIGCOMM Asia Workshop*, 2005.

[14] iPod 3G Accessory Protocol. `http://stud3.tuwien.ac.at/~e0026607/ipod_remote/ipod_ap.html`.

[15] iPod Connector Female SMD. `http://www.sparkfun.com/commerce/product_info.php?products_id=8035`.

[16] iPod Dock Connector. `http://home.swipnet.se/ridax/connector.htm`.

[17] iPod Linux. `http://ipodlinux.org/Dock_Connector`.

[18] iPod Linux. `http://ipodlinux.org`.

[19] M. Jakobsson and S. Wetzel. Security weaknesses in bluetooth. In *2001 Conference on Topics in Cryptography*, 2001.

[20] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in e-passports. In *IEEE SecureComm*, 2005.

[21] A. Juels, R. Rivest, and M. Szydlo. The blocker tag: Selective blocking of rfid tags for consumer privacy. In *10th Annual ACM CCS*, 2003.

[22] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *25th Annual International Cryptography Conference*, August 2005.

[23] Loca – About Loca. `http://www.loca-lab.org/`.

[24] A. McCue. Privacy Groups Protest RFID Tagging of Razors, ZDNet.co.uk, 2003. `http://news.zdnet.co.uk/emergingtech/0,1000000183,39115718,00.htm`.

[25] D. Molnar and D. Wagner. Privacy and security in library RFID issues, practices, and architectures. In *11th ACM Conference on Computer and Communications Security (CCS 2004)*, 2004.

[26] Nike + iPod Frequently Asked Questions (Technical). `http://docs.info.apple.com/article.html?artnum=303934`. Last accessed on November 12, 2006.

[27] E. O'Neill, V. Kostakos, T. Kindberg, A. F. gen. Schieck, A. Penn, D. S. Fraser, and T. Jones. Instrumenting the city: Developing methods for observing and understanding the digital cityscape. In *Ubicomp*, 2006.

[28] M. Pels, J. Barhorst, M. Michels, R. Hobo, and J. Barendse. Tracking people using bluetooth: Implications of enabling bluetooth discoverable mode, 2005. Manuscript.

[29] Radio Frequency Identification – Wikipedia. `http://en.wikipedia.org/wiki/RFID`.

[30] F.-L. Wong and F. Stajano. Location privacy in bluetooth. In *2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, 2005.