Devices That Tell On You: Privacy Trends in Consumer Ubiquitous Computing

T. Scott Saponas
University of Washington

Jonathan Lester
University of Washington

Carl Hartung
University of Washington

Sameer Agarwal
University of Washington

Tadayoshi Kohno
University of Washington

Abstract

We analyze three new consumer electronic gadgets in order to gauge the privacy and security trends in mass-market UbiComp devices. Our study of the *Slingbox Pro* uncovers a new information leakage vector for encrypted streaming multimedia. By exploiting properties of variable bitrate encoding schemes, we show that a passive adversary can determine with high probability the movie that a user is watching via her Slingbox, even when the Slingbox uses encryption. We experimentally evaluated our method against a database of over 100 hours of network traces for 26 distinct movies.

Despite an opportunity to provide significantly more location privacy than existing devices, like RFIDs, we find that an attacker can trivially exploit the *Nike+iPod Sport Kit*'s design to track users; we demonstrate this with a GoogleMaps-based distributed surveillance system. We also uncover security issues with the way Microsoft *Zunes* manage their social relationships.

We show how these products' designers could have significantly raised the bar against some of our attacks. We also use some of our attacks to motivate fundamental security and privacy challenges for future UbiComp devices.

Keywords: Information leakage, variable bitrate (VBR) encoding, encryption, multimedia security, privacy, location privacy, mobile social applications, UbiComp.

1 Introduction

As technology continues to advance, computational devices will increasingly permeate our everyday lives, placing more and more wireless computers into our environment and onto us. Many manufactures have predicted that the increasing capabilities and decreasing costs of wireless radios will enable common electronics in future homes to be predominantly wireless, eliminating the clutter of wires common in today's homes. For example, TVs, cable boxes, speakers, and DVD players could communicate without the proximity restrictions of wires. The changing technological landscape will also lead to new computing devices, such as personal health monitors, for us to wear on our persons as we move around our community. However, despite advances in these areas we have only just begun to see the first examples of such technologies enter the marketplace at a broad scale. While the Ubiquitous Computing (UbiComp) revolution will have many positive aspects, we must be careful to not simultaneously endanger users' privacy or security.

By studying the Sling Media Slingbox Pro, the Nike+iPod Sport Kit, and the Microsoft Zune, we provide a checkpoint of current industrial trends regarding the privacy and security of this new generation of UbiComp devices. (The Slingbox Pro is a video relay system; the Nike+iPod Sport Kit is a wireless exercise accessory for the iPod Nano; and the Zune is a portable wireless media player.) In some cases, such as our techniques for inferring information about what movie a user is watching from 10 minutes of a Slingbox Pro's encrypted transmissions, we present new directions for computer security research. For some of our other results, such as the Nike+iPod's use of a globally unique persistent identifier, the key privacy issues that we uncover are not new; but the ease with which we are able to mount our attacks is surprising. This is particularly true because we show that it would have been technically possible for the Nike+iPod designers to prevent our attacks.

In all cases, we use our results with these devices to paint a set of research challenges that future commercial UbiComp devices should address in order to provide users' with strong levels of privacy and security.

On Our Choice of Devices. The Slingbox Pro, the Nike+iPod Sport Kit, and the Microsoft Zune represent a cross-section of the different classes of UbiComp devices one might encounter in the future: (1) devices that permeate our environment and that stream or exchange

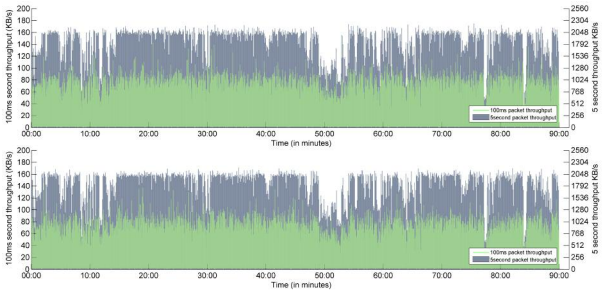


Figure 1: 5 second and 100 millisecond throughput for two traces of Ocean's Eleven played via the Slingbox and captured via a wired connection. Notice the (visual) similarity between the traces.

information; (2) devices that users have on their persons all the time; and (3) devices that promote social interactions. While there is no perfect division between these different classes of devices (e.g., devices that users have on themselves all the time may also exchange content and promote social activity), there are unique aspects to the challenges for each class of devices; we therefore consider each in turn. Specifically, (1) we use the Slingbox Pro as a vehicle to study the issues and challenges affecting next-generation wireless multimedia environments, (2) we use the Nike+iPod Sport Kit as the basis for assessing the issues and challenges with devices that we have on our persons all the time, and (3) we use the Zune as a foothold into understanding the issues and challenges with devices promoting social activity.

Below we survey our results and challenges for each of these scenarios in turn, deferring further details to the body of this paper.

1.1 The Sling Media Slingbox Pro

The Slingbox Pro allows users to remotely view (sling) the contents of their TV over the Internet. The makers of the Slingbox Pro are staged to introduce a new device, the wireless SlingCatcher, which will allow Slingbox users to sling video to other TVs located within the same home, thereby making it one of the first next-generation wireless video multimedia systems for the home [40]. Since the SlingCatcher will not be commercially available until later this year, we choose to study the privacy-preserving properties of a Slingbox streaming encrypted movies to a nearby computer over 802.11 wireless.

We describe in the following sections a technique for monitoring a network connection, wired or wireless, and based on the rate at which data is being sent from one device to the other, predicting the content that is being transferred. Our method consists of two parts. First, we describe a procedure for collecting throughput traces across wired and wireless connections and combining them into a single reference trace per movie. These reference traces are collected into a database for future query use. (Figure 1 shows the raw 5 second and 100 millisecond throughput data for two wired traces of Ocean's Eleven.) Second, we describe a simple Discrete Fourier Transform based matching algorithm for querying this database and predicting the content being transmitted.

We test this algorithm on a dataset consisting of over 100 hours of network throughput data. With only 10 minutes worth of monitoring data, we are able to predict with 62% accuracy the movie that is being watched (on average over all movies); this compares favorably with the less than 4% accuracy that one would achieve by random chance. With 40 minutes worth of monitoring data, we are able to predict the movie with 77% accuracy. For certain movies we can do significantly better; for 15 out of the 26 movies, given a 40 minute trace we are able to predict the correct movie with over 98% accuracy. Given the simplicity of our algorithm, this indicates a significant amount of information leakage — a fact that is not immediately obvious to the users, who likely trust the built in encryption in the device to protect privacy.

Any transmission method whose characteristics depend on the content that is being transmitted is susceptible to the kind of attack we have described. As the world moves towards more advanced multimedia compression methods, and streaming media becomes ubiquitous, variable bitrate encoding is here to stay. Preventing information leakage in variable bitrate streams without a significant performance penalty is an interesting challenge for both the signal processing and the security communities. More broadly, a fundamental challenge that we must address is how to identify, understand, and mitigate information leakage channels in the full range of upcoming UbiComp devices.

1.2 The Nike+iPod Sport Kit

The Nike+iPod Sport Kit is a new wireless accessory for the iPod Nano; see Figure 2. The kit consists of two components — a wireless sensor that a user puts in one of her shoes and a receiver that she attaches to her iPod Nano. When the user walks or runs, the sensor wirelessly transmits information to the receiver; the receiver and iPod will then interpret that information and provide interactive audio feedback to the user about her workout. The Nike+iPod sensor does have an on-off button, but the online documentation suggests that most users should leave their sensors in the on position. Moreover, since the Nike+iPod online documentation encourages users to "just drop the sensor in their Nike+ shoes and forget about it [36]," the Nike+iPod Sport Kit is a prime example of the types of devices that people might even-

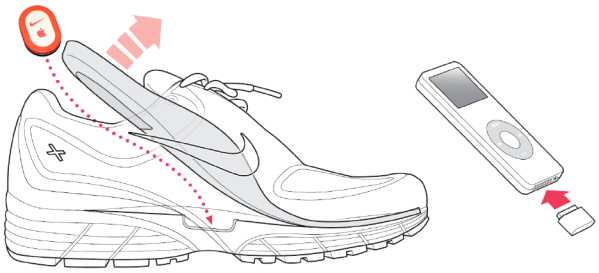


Figure 2: A Nike+iPod sensor in a Nike+ shoe and a Nike+iPod receiver connected to an iPod Nano.

tually find on themselves all the time.

One well-known potential privacy risk with having wireless devices on ourselves all the time is: if the devices use *unique identifiers* when they communicate, and if someone can intercept (sniff) those unique identifiers from the communications, then that someone might learn potentially private information about a user's presence or location. This someone might use that information in ways that are *not* in a user's best interest; e.g., a stalker might use this information to digitally track one or many people, a company might use this information for targeted advertising, and a court might examine this information when debating a contentious case. Location and tracking issues such as these are broadly discussed in the context of RFIDs [27], bluetooth devices [26, 44], and (to a lesser extent) 802.11 wireless devices [15], and there is a large body of UbiComp literature focused on privacy in location-aware systems [5, 11, 12, 19, 20, 25, 22, 29, 34]. Given this broad awareness of the potential trackability issues with wireless devices, and given media reports that the Nike+iPod Sport Kit used a proprietary wireless protocol [35] we set out to determine whether the new Nike+iPod Sport Kit proprietary system "raised the bar" against parties wishing to track users' locations.

We describe the technical process that we went through in order to discover the Nike+iPod Sport Kit protocol in Section 3. The key discovery we found is that not only does each Nike+iPod sensor have a globally unique identifier, but we can cheaply and easily detect the transmissions from the Nike+iPod shoe sensors from 10–20 meters away — an order of magnitude further than what one would expect from a wireless device that only needs to communicate from a user's shoe to the user's iPod (typically strapped around the user's arm), and also significantly further than the conventional passive RFID. The Nike+iPod sensor also broadcasts its unique identifier even when there are no iPods nearby — the user must simply be moving with a Nike+iPod sensor in her shoe. To illustrate the ease with which one could create a Nike+iPod tracking system, we developed a network of Nike+iPod surveillance devices, including a $250 gumstix-based node. The gumstix uses an 802.11 wireless Internet connection to dynamically stream surveillance data to our back-end server, which then displays the surveillance data in a GoogleMaps application in real time.

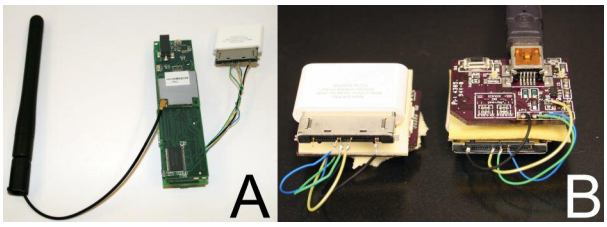


Figure 3: (a) A gumstix-based Nike+iPod surveillance device with wireless Internet capabilities. (b) Our Nike+iPod Receiver to USB adapter

We then describe cryptographic mechanisms that, if implemented, would have significantly improved the Nike+iPod Sport Kit's resistance to our tracking attacks, albeit with the potential drawback of additional resource consumption (e.g., battery life and communication overhead). Our basic approach is to mask the unique identifiers so that only the intended recipient can unmask them. Our solution, however, exploits the fact that the Nike+iPod Sport Kit has a very simplistic communications topology — at any given time a Nike+iPod Sport Kit sensors only needs to be able to communicate with one receiver. The challenge is therefore to lift our privacy-preserving mechanisms (or other mechanisms) to a broader context with heterogeneous devices communicating in an *ad hoc* manner.

1.3 The Microsoft Zune

The Microsoft Zune is a portable digital media player with one (currently) unusual feature: built in 802.11 wireless capabilities. The intended goal is to let users wirelessly share pictures and songs with other nearby Zunes — including Zunes belonging to total strangers. As such, the Zune is arguably the first major commercial device with the design goal of helping catalyze *ad hoc* social interactions in a peer-to-peer wireless environment. (Strictly speaking, we have not read the Zune design documents. Rather, we are inferring this design goal from articles in the popular press and from other publicly available information about the Zune [32].)

Unfortunately, just as it is possible for spammers to send unsolicited or inappropriate emails to users, it is possible for an attacker to beam unsolicited content to a nearby Zune. This unsolicited content may be annoying, such as advertisements or propaganda, or malicious, such as images or songs that might make the recipient

feel uncomfortable or unsafe.

Given the Zune's goal of enabling *ad hoc* interactions, the Zune cannot fall back on traditional mechanisms for preventing unsolicited content, such as buddy lists for instant messaging. Further, much of the research on social interactions for ubiquitous devices is restricted to scenarios where users have a hierarchy of social relationships (e.g., friends and non-friends) [22], which is incompatible with the assumed Zune design goals. Rather, in apparent anticipation of such unsolicited content, the Microsoft Zune allows users to "block" a particular device — a malicious individual might be able to get a user to accept an image or song once, but the recipient should be able to block the originating device from ever sending the user other content in the future. Unfortunately, we find that it is easy for an adversary to subvert this blocking mechanism, thereby allowing the adversary to repeatedly initiate content pushes to the victim until the victim walks out of range or turns off the wireless in her Zune. While we describe techniques that would address the above scenario in the particular case of the Zune, the observations we make underscore two challenges for UbiComp devices designed to enable *ad hoc* social interactions: (1) how to technically implement a blocking procedure or proactively protect against undesired content, especially among a set of heterogeneous devices, and (2) how to balance the blocking mechanisms with our desire to protect location privacy and avoid certain uses of globally unique identifiers.

1.4 Organization and Remarks

We respectively discuss our analyses of the Slingbox Pro, the Nike+iPod Sport Kit, and the Microsoft Zune, as well as the associated research challenges, in Sections 2, 3, and 4. We discuss related work in-line.

We stress that there is no evidence that Sling Media, Apple, Nike, or Microsoft intended for any of these devices to be used in any malicious manner. Neither Sling Media, Apple, Nike, nor Microsoft endorsed this study.

2 The Slingbox Pro: Information Leakage and Variable Bitrate Encoding

Although the future of home entertainment is somewhat fuzzy, many companies have predicted the future home to be a wireless one. Wireless devices tend to be easier to install (though not necessarily easier to setup), provide the user with more flexibility, allow the devices to interoperate with other technologies, and reduce clutter from wires. While it is currently easier to simply plug these devices in once and forget about them, future wireless technologies promise an ever increasing amount of bandwidth, range, and decreasing manufacturing costs, making them more appealing and more likely to be included in future products. Consider, for example, the buzz associated with the upcoming SlingCatcher and the Apple TV; the former is expected to feature integrated wireless support; the latter currently does. In addition to the drive for devices to be connected together, wirelessly, in the home, these devices are often finding themselves networked together and connected to the Internet.

Protecting our private information becomes increasingly difficult as we begin to continually use more wireless devices. Devices in our homes could leak private information to wireless eavesdroppers or, when using home devices over the Internet, wired eavesdroppers. We have investigated one such new wireless/remote TV viewing application — the Slingbox Pro — from a privacy standpoint. In doing so we have uncovered a new information leakage vector for encrypted multimedia systems via variable bitrate encoding.

2.1 Slingbox Pro Description

The Slingbox Pro is a networked video streaming device built by Sling Media, Inc. It is capable of streaming video using its built in TV tuner or one of four inputs connected to DVD players, cable TV, personal video recorders, built in TV tuner, etc. and controlling these devices using an IR emitter. The device itself has no hard drive and cannot store media locally, relying on the connected devices to provide the video and audio content. Paired with player software, called SlingPlayer, the user can watch video streamed by the Slingbox Pro on their laptop, desktop, or PDA anywhere they have Internet access. To accommodate limited network connections when watching videos over a wireless network or away from home, the Slingbox Pro re-encodes the video stream using a variable bitrate encoder, likely a optimized version of Windows Media 9s VC-1 implementation [41]. The Slingbox Pro provides encryption for its data stream (regardless of any transport encryption like WPA). To avoid any problems caused by latency or network interruption the SlingPlayer will cache a buffer of several seconds worth of video. Because of this caching behavior and commonly used packet sizes for TCP packets, the data packets from the Slingbox Pro tend to always be large data packets of similar size or small (seemingly control) packets.

Sling Media recently announced a new device, the wireless SlingCatcher, which users can attach to their TVs. The SlingCatcher would allow users to wirelessly stream content from a Slingbox Pro to their TVs, thereby taking us one step further to a wireless multimedia home. Since the SlingCatcher is not yet commercially available, we choose to study the Slingbox Pro in isolation.

Index	Movie	Index	Movie
A	Bad Boys	B	Bad Boys II
C	Bourne Supremacy	D	Break-Up
E	Harry Potter 1	F	Harry Potter 3
G	Incredibles	H	Men in Black II
I	Ocean's Eleven	J	Short Circuit
K	X2	L	X-Men
M	Air Force One	N	Bourne Identity
O	Caddyshack	P	Clueless
Q	Happy Gilmore	R	Jurassic Park
S	Nightmare Before	T	Office Space
U	Red October	V	Austin Powers 1
W	Austin Powers 2	X	Bruce Almighty
Y	Hurricane	Z	Short Circuit 2

Table 1: Mapping from movie names to movie indices.

2.2 Experimental Setup

We ask whether Slingbox's use of encryption prevents an eavesdropper from discovering what content is being transmitted. This private information could be potentially sensitive if the content is illegal (e.g., pirated), embarrassing, or is otherwise associated with some social stigma. Toward answering this question, we conducted the following experiments.

We streamed a total of 26 movies from a Slingbox Pro to laptop and desktop Windows XP computers running the Slingmedia SlingPlayer. See Table 1. For each movie we streamed the first hour of the movie twice over a wired connection and twice over an 802.11G WPA-PSK TKIP wireless connection. Each time we used the Wireshark protocol analyzer [43] to capture all of the Slingbox encrypted packets to a file. We split each of these traces into 100-millisecond segments and calculate the data throughput for each segment. We use these 100-millisecond throughput traces as the basis for our eavesdropping analysis. See Figure 1 for two examples of these 100-millisecond traces, as well as two example 5-second throughput traces.

2.3 Throughput Analyses

Our eavesdropping algorithm consists of two parts. In the first part, we construct a database of reference traces. Each movie was represented by exactly one reference trace obtained by combining all the throughput traces corresponding to it. Each reference trace requires approximately 600 kilobytes of storage per hour of video. The second part of our algorithm uses this database of reference traces to match against a previously unseen trace. In the following we describe each of these two stages in detail.

Building a Database of Reference Traces. While it is possible to use our matching algorithm against individual raw traces, combining the raw traces for a movie into one reference trace, reduces the time complexity of the matching process and increases the statistical robustness of the matching procedure by eliminating noise and network effects peculiar to a particular trace.

For each movie, all its traces were temporally aligned with each other. This is needed because the trace capturing process was started manually and the traces could be offset in time by 0 to 20 seconds. The alignment was done by looking at the maximum of the normalized cross correlation between smoothed versions of the traces. The smoothing was performed using Savitzky-Golay filtering of degree 2 and windowsize 300. These filters perform smoothing while preserving high frequency content better than standard averaging filters [38]. The reference trace was obtained by averaging over the aligned raw signals.

Matching a Query Trace to the Database. Given a database of reference traces and a short throughput trace, we are now faced with the task of finding the best matching reference trace. This is an instance of the problem of subsequence matching in databases, which has been widely studied in both discrete and continuous domains. Our algorithm is inspired by the work of Faloutsos *et al.* [13].

The simplest approach to subsequence matching in timeseries is to calculate the Euclidean distance between the query sequence and all contiguous subsequences of the same size in the database. Due to the amount of noise present in these traces, this method does not perform well in practice. Following Faloutsos *et al.*, instead of comparing raw throughput values, we first extract noise tolerant features from the traces and then compare subsequences based on these features.

A number of feature extraction schemes have been proposed for this task in the literature, including the Discrete Fourier Transform (DFT) and the Discrete Wavelet Transform. We use the DFT in our experiments. Each point in a throughput sequence was replaced by the first f DFT coefficients of window size w centered on that point. Thus each reference trace in the database was a sequence of non-negative throughput values was replaced by a sequence of f-dimensional Fourier coefficients. The low order Fourier coefficients capture the dominant low frequency behavior in each window. We treat the higher frequency components as noise and ignore them. The same transformation is applied to the query trace. The resulting f-dimensional query trace is compared with all subsequences of the same in the database. The movie with the closest matching subsequence is declared a match. Figure 4 illustrates the database construction and matching process.

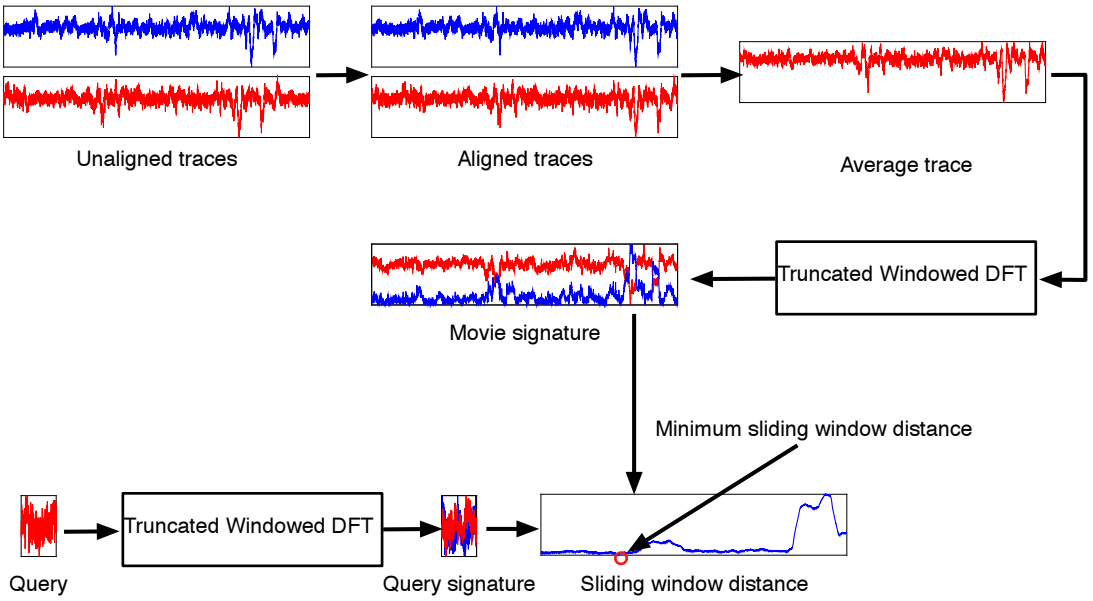


Figure 4: Database construction and query matching. The raw throughput traces corresponding to a movie are aligned and averaged to produce a single composite trace. A windowed Fourier transform is performed on the composite trace and the first $f = 2$ coefficients are kept. A database of movie signatures is constructed in this manner. A query trace is transformed similarly into a signature, and the minimum sliding window distance between the movie signatures and the query signature is calculated. The movie with the minimum distance is declared a match.

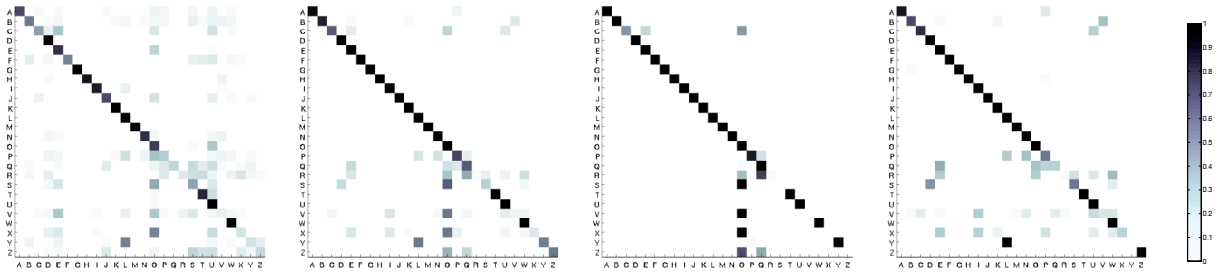
We note that exhaustive matching of all subsequences would not be computationally feasible in a production environment with thousands of references traces. Methods based on approximate nearest neighbor searching can be used to substantially accelerate the matching process without a significant loss in accuracy [13].

Experiments. The above algorithm has two parameters. The size w of the sliding window used to extract the features and the number of Fourier features f, extracted from each window. Both affect the recognition performance of the algorithm. Small values of w and f result in high noise sensitivity, and large values result in over-smoothing of the data. The other factor that affects recognition performance is the length l of the query trace. To choose a good parameter setting, we studied the behavior of the algorithm described above for varying values of $w = [100, 300, 600], f = [1, 2, 4]$. For each setting of the parameters, a random query trace of length $l = 6000$ was extracted from one of the raw throughput traces and compared using the matching algorithm described above. This procedure was repeated 100 times for every parameter setting. The highest accuracy was obtained for $w = 100$ and $f = 2$, or a sliding window of 10 seconds with two Fourier coefficients per window.

We now fix $w = 100$ and $f = 2$ parameters, vary $l = [6000, 12000, 18000, 24000]$ (10, 20, 30, and 40 minutes), and estimate the prediction accuracy of the eavesdropping algorithm. This is done by choosing one throughput trace at a time, constructing the reference trace database using the rest of the throughput traces and then counting how many times random subsequences from the chosen trace result in an incorrect prediction. The average number of incorrect matches over all traces is the leave one out error [18]. In our experiment, 50 random subsequences were chosen from each trace. Sometimes a good shortlist of possible matches is also useful, where the list can be further trimmed with side information, for example, the cable schedule for the area. To account for this possibility, not only do we count the number of times we get the best match right, we also count for varying values of $k = 1, \ldots, 5$, when the algorithm correctly ranks the movie amongst the top k matches.

Table 2 reports the overall accuracy (1-error) of the algorithm, where the accuracy (true positive rate) was computed over all 26 movies. (We define the true positive rate of a movie M as the rate at which a random query trace for movie M is correctly identified as movie M; we define the false positive rate of a movie M as the rate at which a random query trace for a movie $M' \neq M$ is incorrectly identified as movie M.)

For 10- and 40-minute queries, the overall accuracy rates are respectively 62% and 77%. Table 3 and Figures 5 and 6 show that the accuracy rate for individual movies can be significantly higher. From Table 3, 15 of our 26 movies had $\geq 98\%$ true positive rates for 40-



(a) (b) (c) (d)

Figure 5: Confusion matrices for: (a) 10 minute probes from both wired and wireless traces; (b) 40 minute probes from both wired and wireless traces; (c) 40 minute probes from wired traces; (d) 40 minute probes from wireless traces. The color scale is on the right; black corresponds to 1.0 and white corresponds to 0.0.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
10 mins	0.62	0.66	0.69	0.71	0.73
20 mins	0.71	0.75	0.78	0.80	0.82
30 mins	0.74	0.79	0.81	0.84	0.85
40 mins	0.77	0.81	0.84	0.86	0.89
chance	0.04	0.08	0.12	0.15	0.19

Table 2: Overall accuracy of the eavesdropping algorithm. The rows correspond to 10, 20, 30, and 40 minute query traces, and the columns report the success with which the algorithm correctly placed the movie in the top k matches. The bottom row correspond to the probability of a match by random chance.

minute traces with $k = 1$, and 22 of our 26 movies had $\leq 1\%$ false positive rates for our 40-minute traces with $k = 1$.

Figures 5 (a) and (b) show the confusion matrices for 10 and 40 minute query traces with $k = 1$. The shade of the cell in row i, column j denotes the rate at which the i-th movie is identified as the j-th movie; the cells on the diagonal correspond to correct identifications. Contrasting Figures 5 (a) and (b) visually show the increase in accuracy as the length of the query trace increases. Our wireless traces have a higher level of noise as compared to our wired traces. Figures 5 (c) and (d) therefore show the confusion matrices for when the query is restricted to (c) wired and (d) wireless traces. Note that a few movies were misidentified as Caddyshack, as represented by the vertical band most visible in Figure 5 (c); this is likely due to the fact that the bitrate for Caddyshack was fairly constant and the misidentified movies had significant noise (e.g., the wireless traces for Austin Powers 1 had significant noise, which influenced the composite reference trace and therefore the ability of the Austin Power query trace to match to the reference trace).

Movie Index	True positives n minute probes $n=10$	True positives n minute probes $n=40$	False positives n minute probes $n=10$	False positives n minute probes $n=40$
A	0.67	0.95	0.00	0.00
B	0.51	0.86	0.01	0.00
C	0.38	0.60	0.01	0.00
D	1.00	1.00	0.02	0.01
E	0.78	1.00	0.04	0.02
F	0.47	0.99	0.00	0.00
G	0.99	0.98	0.00	0.00
H	0.90	0.99	0.00	0.00
I	0.87	1.00	0.00	0.01
J	0.66	1.00	0.01	0.00
K	0.99	0.99	0.00	0.00
L	0.99	1.00	0.03	0.02
M	0.98	0.99	0.00	0.00
N	0.79	1.00	0.00	0.01
O	0.72	1.00	0.08	0.10
P	0.22	0.68	0.01	0.01
Q	0.18	0.59	0.00	0.02
R	0.10	0.02	0.00	0.00
S	0.37	0.20	0.03	0.00
T	0.83	1.00	0.02	0.00
U	0.97	1.00	0.06	0.01
V	0.07	0.06	0.01	0.01
W	0.99	1.00	0.01	0.01
X	0.14	0.10	0.01	0.00
Y	0.12	0.49	0.01	0.00
Z	0.18	0.50	0.01	0.00

Table 3: True and false positive rates for 10 and 40 minute probes of both wired and wireless traces. The true positive rate of a movie M is the rate at which an n-minute query of that movie is correctly identified as movie M. The false positive rate of a movie M is the rate at which an n-minute query of some other movie $M' \neq M$ is incorrectly identified as M.

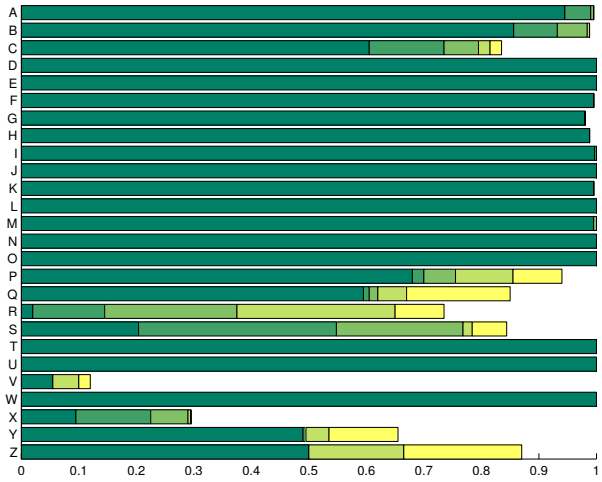


Figure 6: Accuracy per movie for 40 minute query traces; $k = 1$ through $k = 5$.

2.4 Limitations, Implications, and Challenges

While our experiments were conducted in a laboratory setting, they do reflect some possible configurations that one might encounter in a future home equipped with many wireless multimedia devices. The implications of our results are, therefore, that an adversary in close proximity to a users' home might be able to infer information about what videos a user is watching. This adversary might be a nosy neighbor. Or the adversary might be someone sitting outside in a van, looking to collect forensics evidence about those viewing "illegal" (e.g., censored or pirated) content. Moreover, a content producer (such as the creator of a movie) could intentionally construct its movies to have stronger, more distinctive fingerprints. This situation would seem to violate the user's perception of privacy within their own home, especially given the Slingbox Pro's use of encryption.

More broadly, our Slingbox results provide further evidence that encryption alone cannot fully conceal the contents of encrypted data. Other results show that one can infer the origins of encrypted web traffic or infer application protocol behaviors from encrypted data [30, 45]. Concurrent with this work, Wright *et al.* show how variable bitrate encodings can reveal the language spoken through an encrypted VoIP connection [46]. Protecting against such information leakage vectors for all possible applications seems to be a fundamental challenge. Indeed, it may be difficult to simultaneously preserve desirable properties like low-latency and low bandwidth consumption while also allowing for applications with bursty or otherwise data-dependent communication properties. As a concrete example, while it may be possible to significantly raise the bar against information leakage through the Slingbox by having the Slingbox push data at a constant rate while a user is watching a movie, a passive eavesdropper may still be able to learn when a user watches movies, and for how long. The challenge, therefore, is to first determine the possible information leakage vectors, understand their implications, and develop technical means for mitigating them.

3 The Nike+iPod Sport Kit: Devices that Reveal Your Presence

The Nike+iPod Sport Kit foreshadows the types of application-specific UbiComp devices that we might soon find ourselves wearing as part of our daily routine. Indeed, based on publicly available information about the intended usage of the Nike+iPod Sport Kit, as well as our own personal observations, we expect that many Nike+iPod users will always leave their Nike+iPod sensors turned on and in their shoes.

We describe here the steps we took to discover the Nike+iPod protocol; our goal was to assess whether the Nike+iPod Sport Kit provides protection mechanisms against an adversary who wishes to track users' locations. Having uncovered no such protection mechanisms, we then describe our subsequent steps to gauge how easy and cheap it might be for an adversary to implement our attacks. Finally we consider fixes to the Nike+iPod protocol as well as some broader research challenges that our results raise.

3.1 Nike+iPod Description

The Nike+iPod Sport Kit allows runners and walkers to hear real time workout progress reports on their iPod Nanos. 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 kit consists of a receiver and a sensor. Users place the sensor in their left Nike+ shoe and attach the receiver to their iPod Nano 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 audio feedback (via the iPod headphones) to the person about his or her workout. As of September 2006, Apple has sold more than 450,000 of the $29 (USD) Nike+iPod Sport Kits [1].

3.2 Discovering the Nike+iPod Protocol

Initial Analysis. The first step 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 the readings from other nearby 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 results 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, Serial Communications, and Unique Identifiers. 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 [24], we determined that 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, and iPod receive. We verified that digital data was being sent across this serial connection using an oscilloscope and soldered wires connecting them to the serial port of our computer. With the receiver connected to the iPod we turned on the iPod and observed data sent in both directions over the serial connection.

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 back to the iPod until the new sensor is discovered and linked by the receiver. Finally, the iPod sends some more data back to the receiver.

After collecting and 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 same four bytes; however, the next four bytes were different depending on which sensor we used. In all our experiments these four bytes appear to be consistent and unique for a single sensor, and therefore we refer to these four bytes as the sensor's *unique identifier* or *UID*. As further corroboration for the uniqueness of these UIDs, we find that we can use the iPod Nano as an oracle for translating between the UIDs and the Nike+iPod sensor's serial number as it appears on the back of the sensor; we omit details but instead refer the reader to Figure 7 for a sketch of how one might use an iPod Nano as a UID to serial number oracle. As suggested above, the Nike+iPod Sport Kit appears to use these UIDs for addressing purposes — after linking, a receiver will only report packets containing the specified UID.

Automatically Discovering UIDs. Our next step was to use the Nike+iPod receiver to listen for sensor UIDs 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.

Implications. Our observations here immediately imply that the Nike+iPod Sport Kit may leak private information about a user's location. Namely, as is well known in the context of other devices (like RFIDs and discoverable bluetooth devices [26, 27, 44]), if a wireless devices broadcasts a persistent globally unique identifier, an attacker with multiple wireless sniffers can correlate the location of that device (and by inference the user) across different physical spaces and over time.

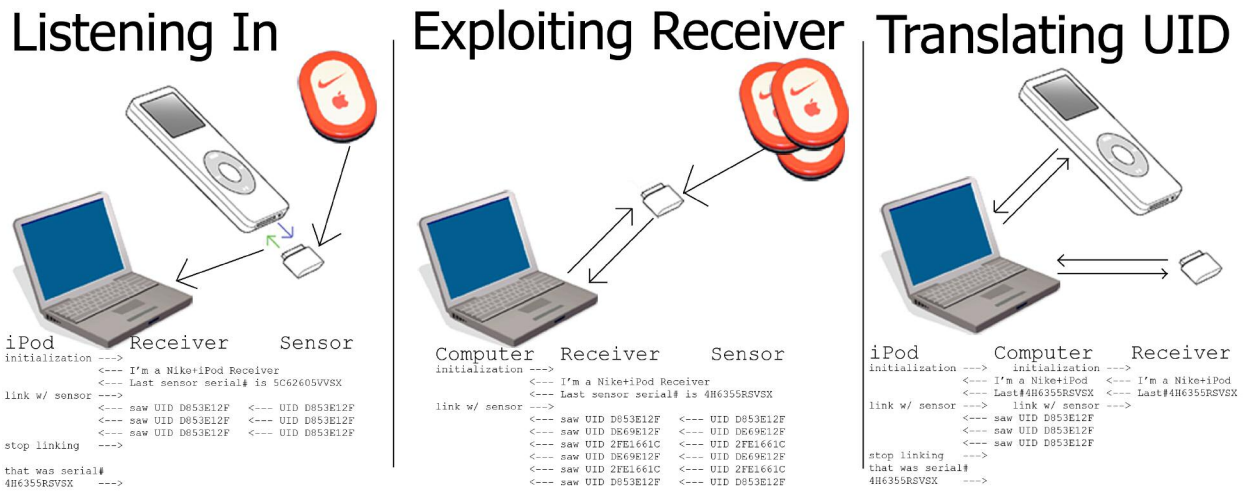


Figure 7: The figure on the left shows our approach for passively monitoring the serial communications between an iPod and the Nike+iPod receiver; the communications between the iPod and the receiver are over a physical, serial connection, and the communication from the sensor to the receiver is via 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.

3.3 Measurements

To understand the implications of our observations in Section 3.2, we must understand the following properties of a Nike+iPod sensor: when it transmits; how often it transmits; the range at which the receiver hears the sensor's UID; and the collision behavior of multiple sensors. We have already partially addressed some of these properties, but elaborate on our observations here.

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 occasionally wake up and start transmitting. Once walking, running, or shaking ceases, the sensor goes to sleep after approximately ten seconds.

While the sensor is awake and nearby we observed that it transmits 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 packets. However, our tests with seven sensors indicated the receiver still hears every sensor UID at least once in a ten second window. 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.

3.4 Instrumenting Attacks

Section 3.2 shows that it is possible for an adversary to extract a Nike+iPod sensor's UIDs from sniffed radio transmissions, and Section 3.3 qualifies the circumstances under which the receiver might be able to sniff those transmissions. These results already enable us to conclude that, despite broad awareness about the trackability concerns with unique identifiers in other technologies (e..g, RFIDs, discoverable bluetooth), new commercial products are still entering the market without any strong protection mechanisms for ensuring users' location privacy.

We now seek to explore just how easy — in terms of cost and technical sophistication — it might be for an adversary to exploit the Nike+iPod Sport Kit's lack of location privacy protection and, at the same time, to explore the types of applications that an adversary might build. For example, one application that we built is a GoogleMaps-based system that pools data from multiple

Nike+iPod sniffers and displays the resulting tracking information on a map in real-time. When assessing the ease with which an attacker might be able to implement a Nike+iPod-based surveillance system, it is worth noting that the attacker may not need to write source code him or herself, but may instead download the necessary software from somewhere on the Internet. We built the following components and systems:

- **Receiver to USB Adaptor.** We created a compact USB receiver module for connecting the Nike+iPod receiver to a computer via USB. Our module does not require any modification to the Nike+iPod receiver; see Figure 3b, and consists of a female iPod connector [23] and a serial-to-USB board utilizing the FTDI FT2232C chipset [14]. We connected the serial pins and power pins of the iPod connector to the appropriate pins of the FT2232C board. When this module is connected to a computer, 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.

  We also created a windows serial communications tool for interfacing with the Nike+iPod Receiver using our adapter. Our tool can detect the UIDs of nearby Nike+iPod sensors and transmit those UID readings, a timestamp, and latitude and longitude information to a back-end SQL server for post-processing; the latitude and longitude are currently set manually. Optionally, when a sensor is detected, this application can take photographs with a USB camera and upload those photographs to the SQL server along with the UID information. This application can also SMS or email sensor information to pre-specified phone numbers or email addresses.

- **Gumstixs.** We 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 Nike+iPod receiver is connected directly to the gumstix's serial port, thereby eliminating the need for our serial-to-USB adaptor. The assembled package is 8cm x 2.1cm x 1.3cm and weighs 1.1 ounces; see Figure 3a.

  Our gumstix-based module runs a 280 line C program that communicates with the Nike+iPod receiver over a serial port and that uses the wifistix 802.11 wireless module to wirelessly transmit real-time surveillance data to a centralized back-end server. The real-time reporting capability allows the gumstix module to be part of a larger real-time surveillance system. If an adversary does not need this real-time capability, then the adversary can reduce the cost of this module by omitting the wifistix.

- **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. The data sources may be our serial communication tool or our gumstix application.

  In real-time mode, sensors' UIDs are overlayed 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. 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 Nike+iPod sensors. Among the related research, demonstration, and commercial bluetooth- and 802.11 wireless-based tracking systems (e.g., [6, 8, 10, 17, 31, 37, 39]), we are unaware of any other location-based surveillance system that goes as far as plotting subjects' locations on a map in real-time.

We also developed two other surveillance devices — one which uses a third-generation iPod and iPod Linux to detect nearby Nike+iPod sensors, and the other of which uses a second-generation Intel Mote (iMote2) to detect nearby Nike+iPod sensors and beams the recorded information to a paired Microsoft SPOT watch via bluetooth. For brevity, and since the above applications provide a survey of the applications that we developed, we omit discussion of our iPod Linux- and iMote2-based applications here.

3.5 Privacy-Preserving Alternatives

Our results show that, despite public awareness of the importance of location privacy and untrackability, major new products are still being introduced without strong privacy guards. We consider this situation unfortunate since in many cases it is technically possible to signifi-

cantly improve consumer privacy.

Exploiting (Largely) Static Associations. Consider 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 would have to be dealt with separately. If the manufacturer decides 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.

Un-Sniffable Unique Identifiers. Assume now that both the sensor and the receiver in a Nike+iPod Sport Kit are preprogrammed 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. This construction shares commonality with the randomized hash lock protocol for anonymous authorization [42] in which an RFID tag reader must try all tag keys in order to determine the identity of an RFID tag; in our case a receiver must attempt to decrypt all received messages, even when the messages are intended for other receivers. 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 [4] 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. 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.

Use an On-Off Switch. One 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. Unfortunately, this approach by itself will not protect consumers' privacy while the devices are in operation. Additionally, 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.

3.6 Challenges

While the above discussion clearly shows that it is possible to significantly improve upon the privacy properties of the current Nike+iPod Sport Kits, from a broader perspective the solutions advocated above are somewhat unsatisfying. For example, how does one generalize the above recommendations (or derive new recommendations) for wireless devices that do not have largely static pairings, such as commercial 802.11 wireless hot spots or the dynamic peer-to-peer pairings of the Zune, where one may wish to allow for *ad hoc* network formations but still restrict access to only authorized devices? And how does one reduce the extra costs (e.g., battery lifetime, packet size, the need to decrypt packets intended to other parties), to environments that cannot afford the extra resource requirements? If we wish to provide a strong level of location privacy for future UbiComp devices, we need to develop mechanisms for handling such broad classes of situations.

The challenge, therefore, is to provide anonymous communications for wireless devices in more diverse and potentially *ad hoc* environments. This challenge is not unique to us — indeed, others have also considered this problem in other restricted contexts [16, 21, 33, 42, 28, 33, 44] — but bears repeating given the potential complexities; e.g., while we have focused this discussion on unique identifiers, which by themselves are not trivial to address, application characteristics and other side channel information, which can survive encryption [30, 45], might facilitate the tracking and identification of individuals.

4 Zunes: Challenges with Managing *Ad Hoc* Mobile Social Interactions

The Microsoft Zune portable media player is one of the first portable media devices to include wireless capability for the purpose of sharing media. Zune owners can enter a coffee shop, turn on their Zune, and discover nearby Zunes. Once a nearby Zune is discovered, users can send music or photos to the nearby Zune. Discovery and sharing are meant to facilitate social interaction; hence the Zune slogan: "Welcome to the Social." Like the Nike+iPod Sport Kit and SlingBox, the Zune represents a gadget pioneering a new application space and represents a central example of our third class of UbiComp devices geared toward catalyzing new social interactions. However, we demonstrate that there are challenges with protecting users' privacy and safety while simultaneously providing ad hoc communications with strangers.

4.1 Zune Description

We focus this description on how the Zune media player allows users to control their social interactions. Consider a scenario consisting of two users, Alice and Bob, and assume that Alice and Bob respectively name their Zunes AliceZune and BobZune; Alice and Bob choose these names when they configure their Zune. If Bob wishes to utilize the Zune social system, to see who's around, he would first use the Zune interface navigate to the "community – nearby devices" menu. He will then see the names of all discoverable nearby Zunes and, depending on the options chosen by the owners of the other Zunes, the names of the songs that his neighbors are listening to or their state (online/busy). If Bob wishes to share a song or picture with his neighbors, he must first select the song or picture and then select the "send" option. The Zune will then show Bob the names of nearby Zunes, and Bob can then send the song or picture to a neighbor of his choosing, in this case AliceZune. The interface on Alice's Zune asks whether Alice wishes to accept a song from BobZune; no additional information about the song or picture is included in the prompt. Alice has two choices: to accept the content or to not accept the content. If Alice accepts the song and later decides that she would like to prevent Bob from ever sending her a song in the future, she can navigate to her Zune's "community – nearby devices" menu, select BobZune, and then select the "block" option.

4.2 Circumventing the Zune Blocking Mechanism

Microsoft appears to envision a world where Zune owners wish to receive interesting content from people they have never met before. Of course, these users also wish to avoid being bothered by people or companies that send inappropriate or annoying content, hence the Zune's blocking feature. Such a situation is not purely hypothetical; indeed, there has recently been media reports about advertisers beaming unsolicited content to users with discoverable Bluetooth devices [7].

Unfortunately, we find that a malicious adversary could circumvent the Zune blocking feature, and we have verified this in practice. The critical issue revolves around how blocking is actually implemented on the Zunes. When Bob sends a song or image to Alice, Alice is only given the option of accepting or denying the song or image; she is not given the option of blocking the sender. Then, after playing the song or viewing the image, if Alice wishes to block Bob's Zune in the future, she must navigate to the "community – nearby devices" menu and actively choose to block BobZune.

The crux of the problem is that Alice will not be able to block Bob's Zune if BobZune is no longer nearby or discoverable.

Disappearing attack Zune. A simple method to circumvent the Zune block feature is, after beaming an inappropriate image, to turn the wireless on the originating Zune off. Since Alice may remember the name of Bob's Zune, and thereby simply deny messages from BobZune in the future, Bob can change the name of his Zune before trying to beam Alice additional content. Also, before beaming Alice the inappropriate content in the first place, Bob could scan his nearby community, find a nearby Zune named CharlieZune, and then name his Zune CharlieZune. If Bob sends inappropriate content to Alice and then turns off his wireless, he might trick Alice into blocking the real CharlieZune.

Fake MAC addresses. Upon further investigation, we find that the Zune neighbor discovery process and blocking mechanism is based on 802.11 probe-responses and MAC addresses. Bob could therefore use a Linux laptop to fool Alice into thinking that she has blocked BobZune when in fact she has not; unlike the observation in the previous paragraph, our attack here works even when there are no other nearby Zunes.

Building on the scenario above, where Bob sends inappropriate content to Alice, disables his Zune's wireless, and changes his Zune's name. Suppose Alice does not like the content she received from Bob and navigates to the nearby list on her Zune. Bob can use his laptop to send out Zune 802.11 probe-responses with the same

name that his Zune was using but with a different MAC address. Alice will then see the previous name of Bob's Zune in her nearby list and select the block command. It will now appear to Alice that she has blocked Bob's Zune. Conversely, what has actually occurred is Alice has blocked a different MAC address. The next time Bob enables his Zune's wireless and attempts to send inappropriate content to Alice, it will appear to Alice that Bob is sending content from a third BobZune that Alice has never seen before. We have implemented a C application for Linux that uses the MadWiFi drivers and an Atheros Chipset-based wireless card to listen to 802.11 probe-requests from Zunes and send a Zune probe-response with whatever name and MAC address the user desires.

Post-blocking privacy. Lastly, even when the blocking mechanism is used successfully, it only stops Alice from receiving new content pushes from BobZune; the malicious user, Bob, can still detect Alice's presence unless she turns off her Zune's wireless capability all together; this has the negative side effect of preventing Alice from sharing any media at all if she doesn't want to be detectable by Bob.

4.3 Improving User Control

Perhaps the most natural method for protecting against such unsolicited content is to adopt what is now common practice in other social applications, such as instant messaging: create a "buddy list" and only accept connections from known buddies. One might populate the buddy list using some interactions that require two Zunes to be in close proximity [2]. Such a buddy list is, however, in direct conflict with the Zune's intended goal of initiating *ad hoc* interactions with total strangers.

Therefore, the goal is to improve the resistance of the Zune blocking mechanisms to attacks like those we present above. One simple solution to Bob's blocking circumvention is to record which Zune sent the specific media and allow the user to block the *sender* of media even if they are not currently nearby and active. We note, however, that there are some subtleties that one must consider. For example, since the Zune blocking mechanism described above seems to be based on the Zune's MAC address (recall that our C program in Section 4.2 created 802.11 probe-responses with forged MAC addresses to trick the Zune blocking feature) Bob might still be able to circumvent this improved blocking mechanism by mounting a MAC-rewriting man-in-the-middle attack between his Zune and Alice's. Since the Zune's communicate using encryption, MAC rewriting of this form will not, however, be successful if the Zunes' MAC addresses are used as input to the encryption key derivation process. We have currently not successfully determined whether or not this is actually the case, but argue below that the use of MAC addresses for this purpose is fundamentally problematic if one also wishes to protect information about a user's presence to outsiders (recall Section 3).

4.4 Challenges

While there has been significant research on providing control over private information in social networks in ubiquitous social applications, much of the work focuses on situations with hierarchical or other complex relationships, such as boss/spouse/friend or buddies/non-buddies [22]. While there is still much work to be done in this space, the Zunes suggest another scenario in which a key target application is to share content with strangers. Blocking individuals in such a scenario can be very challenging when users have complete control over the information that their devices present to others.

When all the devices are homogeneous and incorporate a secure hardware module, one possibility is to let that secure hardware control what information is shared with the user and other devices, and to ensure that some information (such as a unique identifier) is not mutable by the user. The secure hardware might then use this non-mutable information to control blocking. Coupled with the discussion in Section 3, one must ensure that these unique identifiers do not reveal private information about a user's presence. For example, this unique identifier should not be an 802.11 MAC address, which the Zunes currently appear to use for blocking purposes. While there might be approaches for addressing this problem in the case of homogeneous devices with secure hardware from the same manufacturer (e.g., restricted behavior on the secure hardware and symmetric key agreement using the exchange of anonymous public keys [3] signed using a group signature scheme [9]), solving this problem in the case of a heterogeneous environment appears to be a challenge.

5 Conclusions

We technically explore privacy and security properties of several commercial UbiComp products. We find that despite research and public awareness, these products do not provide strong levels of privacy protection and do not put the user in control of their private information.

Our analysis of the encrypted SlingBox stream suggests that transmission characteristics from variable data rate encoding can cause information leakage even when such a stream is encrypted. This puts users privacy at risk because one might assume encryption is enough to thwart an eavesdropper from learning what media one is watching. Our first attempt at recognizing movies via their variable throughput in a 26 movie database yielded

an overall accuracy of approximately 62% for the best match and 73% for ranking in the top 5 matches when a 10 minute query trace was used, and 77% and 89% respectively when a 40 minute query trace was used; which compared with the 4% and 21% respectively that one can expect with random guessing, shows much information leakage. For certain movies our accuracy rates are significantly higher; for example, for 15 out of our 26 movies, a 40-minute query trace will match with the correct movie over 98% of the time. When a variable data rate encoding is used, a content provider could potentially increase this accuracy by using a throughput-based watermarking scheme.

Persistent identifiers in the Nike+iPod Sport Kit and Zune potentially reveal presence and, in the Nike+iPod case, we demonstrate how a tracking system can be built using the Nike+iPod Sport Kit sensors and receivers. We argue that these persistent identifiers should not be used in future devices and should instead be replaced with other privacy preserving mechanisms.

Finally, our evaluation of the Zune blocking scheme shows that an interface design choice coupled with a technology choice can take control away from the consumer and put it in the hands of malicious users. Together, the results from this paper demonstrate with new classes of devices come new privacy and security challenges; privacy must be designed in at all levels of the protocol stack.

Acknowledgments

We thank Yaw Anokwa, Kate Everitt, Kevin Fu, J. Alex Halderman, Karl Koscher, Ed Lazowska, David Molnar, Fabian Monrose, Lincoln Ritter, Avi Rubin, Jason Schultz, Adam Stubblefield, Dan Wallach, and David Wetherall. S. Agarwal was funded by NSF EIA-0321235, University of Washington Animation Research Labs, Washington Research Foundation, Adobe and Microsoft. T. Kohno thanks Cisco Systems Inc. for a gift supporting his research on information leakage and the interactions between compression and encryption.

References

[1] Apple 'It's Showtime!' event – live coverage. `http://www.macworld.com/news/2006/09/12/showtime/index.php`.

[2] D. Balfanz, G. Durfee, R. Grinter, D. Smetters, and P. Stewart. Network-in-a-Box: How to set up a secure wireless network in under a minute. In *13th USENIX Security Symposium*, 2004.

[3] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer-Verlag, Dec. 2001.

[4] 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.

[5] A. Beresford and F. Stajano. Location privacy in pervasive computing. In *IEEE Pervasive Computing*, January 2003.

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

[7] Bluetooth Spam in Public Places. `http://it.slashdot.org/it/07/01/28/2114253.shtml`.

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

[9] D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265, 1991.

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

[11] J. Cornwell, I. Fette, G. Hsieh, M. Prabaker, J. Rao, K. Tang, K. Vaniea, L. Bauer, L. Cranor, J. Hong, B. McLaren, M. Reiter, and N. Sadeh. User-controllable security and privacy for pervsaive computing. In *8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile 2007)*, 2007.

[12] Y. Duan and J. Canny. Protecting user data in ubiquitous computing environments: Towards trustworthy environments. In *Workshop on Privacy Enhancing Technology*, 2004.

[13] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. *ACM SIGMOD Record*, 23(2):419–429, 1994.

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

[15] M. Gruteser and D. Grunwald. A methodological assessment of location privacy risks in wireless hotspot networks. In *First International Conference on Security in Pervasive Computing*, 2003.

[16] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis. In *ACM Mobile Networks and Applications (MONET)*, volume 10, pages 315–325, Hingham, MA, USA, 2005. Kluwer Academic Publishers.

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

[18] T. Hastie, R. Tibshirani, J. Friedman, et al. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2001.

[19] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Second International Conference on Mobile Systems, Applications, and Services (Mobisys 2004)*, 2004.

[20] J. I. Hong and J. A. Landay. Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *Designing Interactive Systems (DIS2004)*, 2004.

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

[22] G. Iachello, I. Smith, S. Consolvo, M. chen, and G. Abowd. Developing privacy guidelines for social location disclosure applications and services. In *Symposium on Usable Privacy and Security*, 2005.

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

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

[25] A. Jacobs and G. Abowd. A framework for comparing perspectives on privacy and pervasive technologies. In *IEEE Pervasive Computing Magazine. Volume 2, Number 4, October-December*, 2003.

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

[27] A. Juels. RFID security and privacy: A research survey. In *IEEE Journal on Selected Areas in Communications*, 2006.

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

[29] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, timothy Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Third International Conference on Pervasive Computing*, 2005.

[30] M. Liberatore and B. N. Levine. Inferring the Source of Encrypted HTTP Connections. In *Proc. ACM conference on Computer and Communications Security (CCS)*, October 2006.

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

[32] Microsofts Zune Delivers Connected Music and Entertainment Experience. `http://www.microsoft.com/presspass/press/2006/sep06/09-14ZuneUnveiling%PR.mspx`.

[33] 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.

[34] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. In *IEEE Pervasive Computing*, 2003.

[35] Nike and Apple Lanch Nike+iPod Sport Kit (for real). `http://www.engadget.com/2006/07/13/nike-and-apple-launch-nike-ipod-sport-kit-for-real/`.

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

[37] 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.

[38] S. Orfanidis. *Introduction to Signal Processing. Inglewood Cliffs*. NJ: Prentice-Hall, 1996.

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

[40] Sling Media Announces SlingCatcher. `http://us.slingmedia.com/object/io_1168286861787.html`.

[41] Sling Media's Newly-Released SlingBox Uses Microsoft Windows Media and Texas Instruments Digital Media Technology to Deliver On-the-Go Entertainment. `http://us.slingmedia.com/object/io_1157566629962.html`.

[42] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Security in Pervasive Computing*, 2004.

[43] WireShark. `http://www.wireshark.org`.

[44] 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.

[45] C. Wright, F. Monrose, and G. Masson. On Inferring Application Protocol Behaviors in Encrypted Network Traffic. In *Journal of Machine Learning Research (JMLR): Special issue on Machine Learning for Computer Security*, 2006.

[46] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted VoIP traffic: Alejandra y Roberto or Alice and Bob? In *16th Usenix Security Symposium*, 2007.