Acoustic Eavesdropping Attacks on Constrained Wireless Device Pairing

Tzipora Halevi and Nitesh Saxena

***Abstract*—Secure "pairing" of wireless devices based on auxiliary or out-of-band (OOB)—audio, visual, or tactile—communication is a well-established research direction. Specifically, authenticated as well as secret OOB (AS-OOB) channels have been shown to be quite useful for this purpose. Pairing can be achieved by simply transmitting the key or short password over the AS-OOB channel, avoiding potential serious human errors.**

This paper analyzes the security of AS-OOB pairing. Specifically, we take a closer look at three notable prior AS-OOB pairing proposals and challenge the assumptions upon which the security of these proposals relies, i.e., the secrecy of underlying audio channels. The first proposal (IMD Pairing) uses a low frequency audio channel to pair an implanted RFID tag with an external reader. The second proposal (PIN-Vibra) uses an automated vibrational channel to pair a mobile phone with a personal RFID tag. The third proposal (BEDA) uses vibration (or blinking) on one device and manually synchronized button pressing on another device or simultaneous button pressing on two devices. We demonstrate the feasibility of eavesdropping over acoustic emanations associated with these methods and conclude that they provide a weaker level of security than was originally assumed or desired for the pairing operation.

***Index Terms*—Device pairing, authentication, audio emanations, signal processing.**

I. Introduction

SHORT- and medium-range wireless communication—based on technologies such as Bluetooth, WiFi and RFID (Radio Frequency Identification)—is becoming increasingly popular. This surge in popularity, however, brings about various security risks. The Wireless communication channel is easy to eavesdrop upon and to manipulate, and therefore a fundamental security objective is to secure this communication channel. In this paper, we will use the term "pairing" to refer to the operation of bootstrapping secure communication between two wireless devices, resistant against eavesdropping and man-in-the-middle attacks. Examples of pairing include pairing of a WiFi laptop and an access point, a Bluetooth keyboard and a desktop, an RFID tag and reader. Pairing would be easy to achieve, if there existed a global infrastructure enabling devices to share an on- or offline trusted third party, a certification authority, a PKI or any preconfigured secrets. However, such a global infrastructure may not be possible in practice, making pairing a challenging research problem.

A promising and well-established research direction to pairing is to leverage an auxiliary channel, also called an out-of-band (OOB) channel, which is governed by the users operating the devices. Examples of OOB channels include audio, visual, and tactile channels. Unlike the radio communication channels, OOB channels are "human-perceptible", i.e., the underlying transmission/reception can be perceived by one or more of human senses. Due to this property, OOB communication naturally provides (source) authentication and integrity, unlike radio communication. In other words, a user can validate the intended source of an OOB message and an adversary can not manipulate the OOB messages in transit (although he can eavesdrop). We refer to such an authenticated OOB communication as A-OOB.

Using these protocols, a wide-variety of pairing methods—based on visual, audio, tactile and infra-red—A-OOB channels have been proposed. We refer the reader to an exhaustive survey and comparative analysis of various A-OOB pairing methods [14].

The focus of this paper is on pairing *constrained devices*. We define a constrained device as a device that lacks good quality output interfaces (e.g., a speaker, display), input interfaces (e.g., keypads), or receivers (e.g., microphone, camera), and may not be physically accessible. Examples of constrained devices include headsets, access points, and medical implants.[1]

A-OOB pairing of constrained devices can be very complicated due to several reasons (we discuss these in Section II-A). In general, establishing (bidirectional) automated A-OOB channels on constrained devices might be quite difficult. Manual mechanisms for pairing constrained devices can also be prone to *fatal* human errors [31] that eventually translate into man-in-the-middle attacks. A *fatal* human error is defined as an error that violates the security goal of the pairing mechanism. In particular, this would result in one of the pairing devices getting paired with a man-in-the-middle attacker's device, without the device user knowing about it.

A natural work-around to the aforementioned problems is to pair devices based on *secret as well as authenticated* OOB channels (referred to as AS-OOB). In this model, the adversary is not

Manuscript received March 20, 2012; revised November 27, 2012; accepted February 12, 2013. Date of publication February 18, 2013; date of current version March 07, 2013. This work was supported in part by the NSF under Grant 0966187 and Grant CNS-0831397. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any of the sponsors. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang D. Yoo.

T. Halevi is with the Computer Science and Engineering Department, Polytechnic Institute of New York University, Brooklyn, NY 11201 USA (e-mail: thalev01@students.poly.edu).

N. Saxena is with the Computer and Information Sciences Department, University of Alabama, Birmingham, AL 35294 USA (e-mail: saxena@cis.uab.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2013.2247758

[1] Due to economic reasons, such devices may also be constrained in terms of computational resources (e.g., low-cost RFID tags).

only assumed to be incapable of manipulating OOB communication but also can not eavesdrop upon it. Using an AS-OOB channel, pairing can be achieved simply by transmitting—from one device to the other—the key over this channel, avoiding any potential fatal human errors and without having to perform any cryptography. If this channel is low-bandwidth, a short PIN or password can be transferred instead and a password-based authenticated key agreement (PAKA) protocol [5], [9] can be executed to achieve pairing. Several prior proposals, including [11], [24], [27], [28] (reviewed below), have taken this approach to pairing.

A. Motivation: Security of AS-OOB Pairing

In this work, we set out to investigate the security of pairing based on AS-OOB. More specifically, we take a closer look at three notable prior AS-OOB pairing proposals (summarized as follows) and challenge the direct or indirect assumption upon which the security of these proposals relies, i.e., *the secrecy of underlying or associated audio channels*. (We describe the methods in detail in Section II-B.)

- **IMD Pairing:** This method [11] uses a low-frequency audio channel to pair an RFID tag—attached to an IMD (Implanted Medical Device)—with an authorized reader or programmer. Basically, the tag generates a random key and broadcasts it to the reader which listens to it from a close distance (e.g., a microphone is placed in close proximity to the patient's chest in case of a cardiac implant).
- **PIN-Vibra:** This method (also referred to as "Vibrate-to-Unlock") [24], [25] uses an automated vibrational channel to pair a personal RFID tag with a mobile phone. The phone generates a PIN and transmits it to (an accelerometer-equipped) tag through its vibrations, while the user presses the phone against the tag. The same channel is later used by the phone to authenticate to (or activate) the tag.
- **BEDA:** This method (Button-Enabled Device Association) [27], [28] involves one device encoding a short password into vibrations (or blinking of an LED), which is transmitted to the other device by manually synchronized button pressing. Another variant of this method involves a user pressing buttons simultaneously on two devices. The presses and releases of the buttons generate a short password shared by both devices. We refer to the variant that uses vibration as Vibrate-Button, the one that uses blinking as Blink-Button and the third variant as Button-Button.

B. Overview of Contributions

We investigate acoustic eavesdropping attacks on pairing applications geared for constrained devices, including IMD pairing (which uses direct acoustic signals), and PIN-Vibra and BEDA (in which the acoustic signals are a by-product of the vibration/button clicking). To our knowledge, such attacks have not been considered in past research (prior to the conference version of this paper [10]). We also study eavesdropping in a realistic setting (from distances up to a few feet away) and compare the results from different distances using very inexpensive equipment (a PC microphone). Previous research on keyboard and printer acoustic emanations (discussed in Section II-C) concentrated on recordings from a single very close by distance or used special equipment (parabolic microphone) for farther recordings.

We start with IMD pairing, which is set to exchange the key using a relatively low-volume IMD device and is meant to perform the key exchange with an external reader from very close by. As reported in [11], the security of IMD pairing is based on the fact that the sound generated is hard to hear from a distance and is too low to be measured. We examine a realistic setup of eavesdropping from 2–3 ft distance (and farther using a parabolic microphone). This may allow an attacker to, for example, place a microphone next to a PC or other equipment in a medical examination room (and a parabolic microphone at a farther distance). We demonstrate the feasibility of eavesdropping directly over the audio transmissions of a piezo element attached to an implanted RFID. We show that the key can be sniffed upon beyond the standard operating parameters of this setup, i.e., from a farther distance from a beeping piezo.

We then examine the PIN-Vibra and BEDA schemes, and show that even though the acoustic emanations are only a by-product of the phone vibrations and the phone keypress, they can be utilized to successfully recover the exchanged short secret. Specifically, for PIN-Vibra, we consider acoustic emanations associated with a vibrating phone. We show that the PIN can be eavesdropped upon even beyond the standard mechanism used by the tag, i.e., without sensing the vibrations using an accelerometer, and beyond the standard operating parameters of this setup, i.e., from a farther distance from the phone.

For BEDA Vibrate-Button, we again consider acoustic emanations associated with a vibrating phone, and for BEDA Blink-Button and Button-Button, we consider acoustic emanations of button pressings. Similar to PIN-Vibra, we demonstrate that BEDA password can be learned beyond the standard mechanism used by this setup, i.e., without manual sensing of vibrations as in Vibrate-Button and without observing the blinking as in Blink-Button and Button-Button, as well as beyond the standard operating distance in Vibrate-Button.

Based on our results, we conclude that all three approaches provide a weaker level of security compared to what was originally assumed or is desired for the pairing operation.

To the best of the authors' knowledge, this paper explores acoustic emanations in the context of the device pairing application. Since pairing is a fundamental security procedure upon which the security of all subsequent communication between the devices rely, we believe it is important to ascertain to what extent acoustic emanations may undermine the security of pairing. We also remark that the problem we consider in this paper is more challenging than the one considered in [2], [32] (we discuss these in Section II-C). This is predominantly because of the fact that the acoustic emanations in our applications are much more feeble. For example, the piezo transmissions coming from inside of a human body in IMD Pairing are severely dampened; similarly, cell phone vibrations and button pressing on mobile devices (such as phones) in PIN-Vibra and BEDA are not as prominent as pressing keys on traditional PC keyboards.

Organization: The rest of this paper is organized as follows. In Section II, we cover background and prior work. In

Section III, we describe the threat model employed in our work. In Section IV, we give an overview of our experimental setup and techniques. In Sections V, VI and VII, we present our audio eavesdropping attacks on IMD Pairing, PIN-Vibra and BEDA, respectively. Finally, in Section VIII, we discuss the implications of our attacks on the security of the three schemes.

II. Background and Prior Work

A. A-OOB Pairing of Constrained Devices

A-OOB pairing of constrained wireless devices has a number of complications. Several prior pairing methods are based on bidirectional automated device-to-device (d2d) A-OOB channels (e.g., [29], [4], [16]). Such d2d channels require both devices to have transmitters and corresponding receivers (e.g., Infra-Red transceivers), which may not exist on constrained devices. In settings where d2d channel(s) do not exist (i.e., when at least one device does not have a receiver), pairing methods can be based upon device-to-human (d2h) and human-to-device (h2d) channel(s) instead (e.g., based on transfer of numbers [31]). However, establishing such channels on constrained devices may also not be feasible.

One remedy to the above problem is to use only unidirectional communication (from device A to B), but have the user transfer the result of pairing shown on B over to A, as shown in [22]. This, however, may lead to a critical security failure—a user may accept the pairing on A even though B indicates otherwise, as shown via a recent usability study in [14]. (This is referred to as a *fatal* human error [14] which translates into a man-in-the-middle attack).

Another possible approach is based on manual comparison of audiovisual OOB strings over synchronized device-to-human (d2h) channels, as shown in [17], [20]. This would only require the two devices to be equipped with low-cost transmitters, such as LED(s) (and two buttons). However, the security of these approaches rely upon the decision made by the user and is prone to fatal human errors, as demonstrated in [14]. Even worse, a *rushing user* [23][2] may simply “accept” the pairing, without having to correctly take part in the decision process.

B. AS-OOB Pairing Methods

IMD Pairing: Wireless implantable medical devices, such as pacemakers and Implantable Cardiac Defibrillators (ICD), have recently been shown [11] to be vulnerable to a wide variety of serious attacks, ranging from eavesdropping of patient sensitive information to modification of stored information and therapies, and denial-of-service. In [11], the authors suggested zero-power defenses, whereby a passive (and thus zero-power) RFID device is attached to the IMD. A prerequisite to achieving authenticated and confidential communication between an IMD and external reader is key agreement, i.e., pairing, which would allow the IMD to establish a shared secret key with the reader on-the-fly.

A-OOB pairing of an IMD would be problematic because IMD is inherently a constrained device. Since an IMD would be inside a human body, establishing visual channels is not possible. Providing tactile inputs to implanted devices may also not be feasible because of lack of physical access. Due to low-cost and zero-power requirements, establishing bidirectional d2d OOB channels may not be possible. Moreover, computational constraints might prevent a low-cost RFID from performing public-key cryptographic computations involved in A-OOB pairing and limit the use of distance bounding techniques [19].

The pairing approach proposed in [11] is based on an audio AS-OOB channel. Basically, the RFID device attached to the IMD is connected with a piezo element, which simply picks a random key and transmits it over a low-frequency audio channel; this key is recorded and decoded by a microphone attached to the reader near the human body. The experiments presented in [11] seem to indicate that the underlying audio channel is resistant to eavesdropping. In particular, it was shown that transmission of the key was easy to feel with the hand in close contact with the human chest enclosing a cardiac implant (using meat to simulate human chest), but was difficult to hear from a farther distance. In this paper, we set out to further investigate this claim regarding the secrecy of IMD Pairing and demonstrate the feasibility of acoustic eavesdropping even from a distance.

PIN-Vibra: Personal (passive) RFID tags (found, e.g., in access cards, e-passports and licenses) are increasingly becoming ubiquitous. Similar to other personal devices, personal RFID tags often store valuable information privy to their users, and are likely to get lost or stolen. However, unlike other personal wireless devices, such information can be easily subject to eavesdropping, relay attacks and unauthorized “reading”, and can lead to owner tracking.

User authentication to an RFID device would allow a user to control when and where her RFID tag can be accessed and thus help solve some of the aforementioned problems. A fundamental roadblock in developing an RFID user authentication mechanism is the lack of any input or output interfaces on RFID tags (RFID devices were not meant to interact with their users) and a somewhat atypical usage model (users often place RFID tags in their wallets and might not be in direct contact with them).

In [24], the authors present PIN-Vibra, a novel approach for user authentication to RFID tags. PIN-Vibra leverages a pervasive device such as a personal mobile phone, motivated by its ubiquity. It uses the mobile phone as an authentication token, forming a unidirectional AS-OOB tactile communication channel between the user and her (accelerometer-equipped) RFID tags. Pairing of (and later authenticating to) an RFID tag requires the user to touch her vibrating phone with the tag (or wallet carrying the tag); the phone encodes a short PIN into vibrations which are read by the tag’s accelerometer and decoded.

The security of PIN-Vibra relies on the secrecy of the underlying vibrational channel, i.e., an adversary who is not in close physical contact with the phone should not be able to learn the transmitted PIN. We investigate the feasibility of eavesdropping the PIN-Vibra vibrational channel and demonstrate how acoustic emanations from a vibrating mobile phone can be eavesdropped upon from a short distance.

[2] A rushing user is a user who—in a rush to connect her devices—would skip through the pairing process, if possible [23].

BEDA: BEDA [27] suggests pairing devices with the help of manual button pressing, thus utilizing the tactile AS-OOB channel. This method is based on a password-authenticated key exchange protocol [9], and has three variants we study in this work: “Vibrate-Button”, “Blink-Button” and “Button-Button”.

BEDA is geared for devices with constrained interfaces; one device needs a vibration capability or an LED, while the other needs only a button. In the first two BEDA variants, the sending device vibrates (or blinks its LED) and the user presses a button on the receiving device. The short password is encoded as the delay between consecutive vibrations (or blinks). As the sending device vibrates (or blinks), the user synchronously presses the button on the other device thereby transmitting the password from one device to another. The third variant of BEDA belongs to a different class of pairing approach—one where randomness is derived via user inputs. In this method, the user enters the password into both devices by clicking a button on each device simultaneously. As argued in [27], the password input using synchronized button pressing exhibits a uniform distribution (i.e., fully random passwords which span the full n digit vector space, where n being the length of the password). This makes this scheme more appealing, even for devices which are not constrained, compared to traditional PIN or passwords which do not follow a uniform distribution and are prone to small-space dictionary attacks.

The security of BEDA is clearly based on the secrecy of the password which is being transmitted via vibration (or blinking) on one device and synchronized button-pressing on the other device or the secrecy of button-pressing on both devices. We show, in this paper, that the three BEDA variants are subject to acoustic eavesdropping. More precisely, we demonstrate that Vibrate-Button is susceptible to acoustic eavesdropping of phone vibrations, and Blink-Button and Button-Button methods are susceptible to acoustic eavesdropping of button-pressing.

C. Acoustic Emanations

Prior work has considered the problem of eavesdropping over acoustic emanations as a side channel. Asonov and Agrawal [2] were the first to investigate the feasibility of eavesdropping over acoustic emanations associated with typing on computer keyboards. They demonstrated that pressing each key on a keyboard produces a unique sound using which an eavesdropper can learn the characters typed by a user. The authors used signal processing techniques, machine learning classifiers and an off-the-shelf PC microphone for eavesdropping from a distance of up to 1 meter.

Zhuang *et al.* [32] examined the same problem and improved upon the work of [2]. In particular, they showed that using Mel Frequency Cepstrum Coefficients (MFCC) features [18] yield better classification accuracies compared to the Fast Fourier Transform (FFT) features used in [2].

In a proof-of-concept work published on the web [26], Shamir and Tromer explore inferring of CPU activities (e.g., patterns of CPU operations and memory access) via acoustic emanations and how they can be used to learn RSA private key.

Acoustic emanations were also utilized for eavesdropping on dot matrix printers. In [6], Briol showed that significant information can be extracted about the printed text, using acoustic emanations to distinguish between the letters ‘W’ and ‘J’. In [3], Backes *et al.* presented an attack which recovers English printed text from the printer audio sounds, using dictionary and language-based models attack.

III. Threat Model

The threat model employed in our work, and in the work on which this paper is based (i.e., IMD Pairing [11], PIN-Vibra [24] and BEDA [27]), follow the “open design” principle, which is a well accepted approach in cryptography in general and key agreement in particular. In this case, the algorithms and the associated parameters (such as the bit length and the beginning sequence values) of the code are publicly known to everyone (including the devices being paired as well as any attacker). The attacker’s goal will be to violate the security of the system based on the knowledge of these algorithms and parameters. We note that, in the case of pairing, the assumption is that the parties do not share any “secrets” in advance (since they do not have any prior context with each other); establishing such secrets is the goal for pairing. Only thing that the parties share are the algorithms and public parameters. Thus, the “closed design” will not work in this setting.

The need for following the above threat model can be further illustrated taking the example of Implantable Medical Devices (IMD). In this case, it is imperative that any valid reader will be able to work with any IMD (e.g., in an emergency situation where the two devices may be complete strangers to each other). Therefore, to be able to exchange the secret keys, both the reader and the IMD need to know in advance the parameters of the code.

Eavesdropping Attacks: Execution of eavesdropping attacks from different distances can be achieved, for example, by hiding a (remotely controlled) wireless microphone near a user’s workspace and hoping that the user pairs his/her devices (e.g., a phone and headset). Another possibility may be when a device suffers a software compromise. For example, in a recent attack, researchers at McAfee [1] managed to activate remotely microphones in a variety of test devices. This shows the threat of eavesdropping is growing due to the fact that microphones are becoming ubiquitous in many devices.

IV. Overview of Our Attacks

In the following sections, we demonstrate the feasibility of acoustic eavesdropping on IMD Pairing, PIN-Vibra and BEDA Vibrate-Button, Blink-Button and Button-Button schemes. We implemented (or used existing prototypes) for each of these methods and recorded the resulting audio signals. We then used signal processing algorithms and, if needed, machine learning classifiers to detect the beginning of signals and decode the transmitted secret (key or a short PIN/password).

In the first two schemes (IMD Pairing and PIN-Vibra), the secret is transmitted as a binary code. The code includes a beginning sequence that helps the receiver (honest decoder) detect the beginning of the key. In the original IMD defense [11], the authors do not specify how the beginning of the secret key is detected. However, detecting the sequence beginning is an essential step for the (authentic) decoder to allow for proper decoding. Adding a beginning sequence is a well-known approach in coding that facilitates a (valid) decoder to detect the signal beginning. An alternative is to add a different frequency to mark the beginning. However, this would be harder to implement with a piezo (a very simple device) and would require changing the original scheme of the IMD paper (which used 2-FSK encoding). For PIN-Vibra, the beginning sequence was included in the original proposal.

We attempt to eavesdrop over the key in two phases: first, we detect the beginning sequence in the key using signal processing algorithms. Then, we extract spectrum features from each consecutive bit and use these features as input to machine learning algorithms that classify each bit value.

The Vibrate-Button, Blink-Button and Button-Button methods differ from the first two in that there is no beginning sequence or a constant bit size in the signal. For these methods, we detect each event (vibration or key press) using signal processing techniques and calculate the key from the time differences between the events.

We note that we did not have any control over the piezo beep volume. Similarly, the phone vibration (as well as the phone key press) volume could not be controlled and is a function of the system. Therefore, the volume was not a parameter of our tests but was a result of the systems' default design.

Our experimental setup, for the three schemes, consisted of the following common components:

- *PC Microphone*: We used a \$20 commodity PC microphone (Logitech model 981-000246).
- *Software and System:* We used the Windows sound recorder (with sampling rate of 22.05 kHz) and the Matlab software for all signal processing and decoding, on an IBM Thinkpad X60 laptop.

V. Eavesdropping IMD Pairing

A. Eavesdropping Challenges and Goals

There are two prior research projects that relate to our work on IMD eavesdropping. The first project [2], [32], (Section II-C) involves eavesdropping over keyboard acoustic emanations. Here, the keyboard audio signals were found to be at least 100 ms apart. This enabled detecting the beginning of each key using spectrum analysis and extracting its signal prior to its classification.

The second project [15] explored device-to-device proximity communications using audible sound. The proposed audio codec uses Amplitude Shift Keying (ASK) and Frequency Shift Keying (FSK) modulation techniques to transmit information between two devices. A specific 'hail' frequency is sent at the beginning of the message which signals the receiver to start decoding. This work does not consider an adversarial setting, and the communicating parties are assumed to be honest and very close by.

One of the main challenges in our work is the fact that, unlike a modem, a piezo can not be programmed to send a specific frequency. Rather, the piezo acts as an electric capacitor which contracts and expands as the voltage across it fluctuates. Since IMD Pairing suggests 2-FSK decoding [11], the main problem in eavesdropping this setup is differentiating between the two resulting frequency ranges of the piezo vibrations used to transmit the key.

In addition, since 2-FSK decoding utilizes only two frequencies, the protocol does not allow the use of a 'hail' signal (unlike the codec application, which is a protocol implemented on a computer or devices that can generate many frequencies [15]). This limits the piezo output to two frequencies that mark each bit value as '0' or '1'. Instead, we use a beginning sequence of "01111110" to mark the beginning of the key. In addition, adding a 'hail' frequency would require the piezo, which is a simple device, to generate yet another distinctive frequency which may not be possible for some piezo devices (we found that our piezo would not produce one distinctive frequency but rather only a combination of frequencies which needed to be detected). Therefore, adding a beginning sequence would be possible in any 2-FSK implementation and would be essential for the valid decoder to detect correctly the beginning bit.

Furthermore, our symbols are short (67 samples per bit), and they are consecutive with no interval/delay between them (unlike the audio signals of keyboard emanations [2], [32]) and sometimes overlap each other. Therefore, we can not detect separately the beginning of each bit but rather use a constant bit length to locate each following bit in the key. Thus, an inaccurate detection of the start of the first bit will cause a shift in all consecutive bit locations (from their true locations) and reduce the decoding rate.

Another issue is that when the piezo is inserted in a human body (simulated by meat), the sound amplitude further subdues (see sound level measurements in Section IV).

We set out to study the weaknesses of the IMD system. We found that even though the piezo generates a string of audio signals that have very low amplitude and which sound very similar, the system is vulnerable to attacks using signal-processing based algorithms. We further attempt to show that even when using a simple off-the-shelf PC microphone and recording a few feet away (outside of a typical PC microphone's optimal recording range), an attacker may still decode the secret key sent.

B. Setup

In addition to the components described in Section IV, we used PUI Audio piezo model AT-2310-T-LW100-R connected to a WISP tag [21] (similar to [11]). For distant recording (12 ft), we also used the Educational Insights Sonic Sleuth, model 5200.

We took the following steps for our eavesdropping experiments:

- We encoded a random 144-bit (128-bit key + 8-bit start & stop sequence "01111110") binary key with 2-FSK modulation and a baud rate of 341 bps as indicated in [11].
- We inserted the piezo within a combination of beef and bacon to emulate a system inside a human chest exactly as described in [11]. The meat-bacon combination included 1 cm of bacon on top of 4 cm of 85% lean ground beef (see Fig. 2). The piezo was attached to the WISP (Fig. 3).

Sound Level Measurements: We measured the level of the piezo sound from different distances and compared it to the readings reported in [11]. We found the piezo audio measured

67 dB SPL when inserted inside the meat (just outside the surface of the meat) and 62 dB from a distance of 3 feet. This was quieter than the piezo described in [11] which measured 84 dB as the piezo buzzing volume just outside the meat surface and 67 dB SPL from 1 meter away. Therefore, although our system is using a quieter piezo than the one originally used in [11] we attempt to show we can eavesdrop upon it.

C. General Approach

Since the piezo is encoded to produce 2-FSK based encoding, we started by characterizing the piezo beep spectrum and tried to detect the "mark" frequencies (binary one) and the "space" frequencies (binary zero). To do this, we first took recordings of the piezo in air, examined its spectrum and detected the main signal characteristics for both binary bits. Then, we took recordings of the piezo inside meat (simulated IMD scenario), examined the spectrum and adjusted the new "characteristic frequencies" according to the updated signals. Example of the signal (inside meat recorded from 3 ft away) appears in Fig. 1(a).

We perform the full key detection in two steps. We first find the key beginning sequence using the frequency characteristics and a specialized procedure that utilizes frequency analysis. We then decode the key with the help of a machine learning classifier that uses frequency-based features extracted from the key bits. The input features are created for each consecutive 3 ms bit in the signal and a classifier is used to decode the each bit. A full diagram of the attack stages appears in Fig. 4.

All of our recordings were taken in a regular office (a graduate student's room). The main sources for background noise were people walking outside the room in the university corridors.

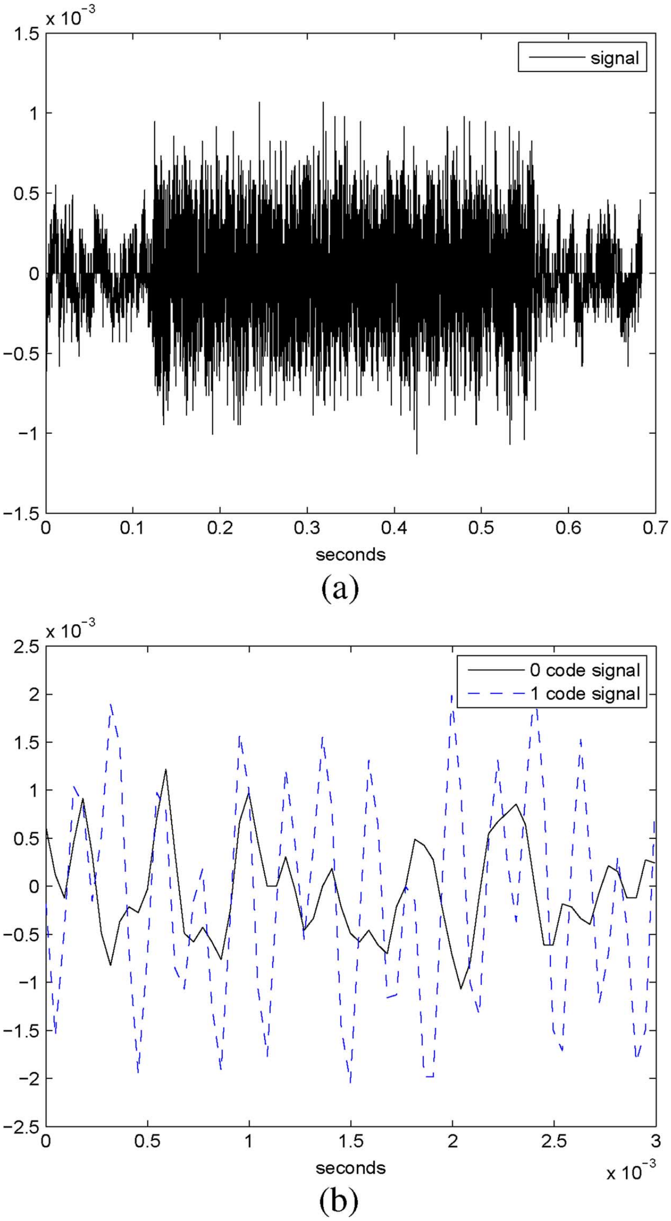
D. Audio Signal Decoding Algorithm

We started by choosing the proper input for our signal decoding algorithm. Our original recording was in the time domain. However, the amplitude of the signal is affected by background noise, microphone characteristics and the distance from the microphone. To overcome such amplitude variations and since the piezo encoding is frequency-based, we transformed our signal into the frequency domain.

Next, we examined the signal to determine the correct window size for which to create the spectrum. We compared using the whole bit lengths (shown in Fig. 1(b)) against using only the middle parts of each bit. We found that due to the short duration of the bit signal (3 ms, 67 samples per bit), we got the best results when we extracted features from the whole bit signal.

1) Frequency Characteristics—Recording With and Without Meat: We first create the bit spectrum of the open-environment acoustic signal by performing Fast Fourier Transform (FFT) on each of the bit signals sent by the piezo (using one full bit duration). We obtained a spectrum with 34 frequency intervals of 335 Hz each (Fig. 5(a)). We observed that the '0' bit spectrum has two peaks in the 1.67–2.68 kHz interval while the '1' bit has a peak at the 2.68–3.35 kHz interval.

We then recorded the piezo beeping inside meat and reviewed the changes in the signal. We found that the audio signal was much more faint and the spectrum was degraded (Fig. 5(b)) which resulted in less noticeable '0' bit peak frequencies. We note that both bit spectrums contained an additional peak around the 2.9 kHz frequency band, but it was more pronounced in the '1' bit spectrum. Therefore, this frequency is later used to detect the existence of the "mark" (binary '1') in the key.

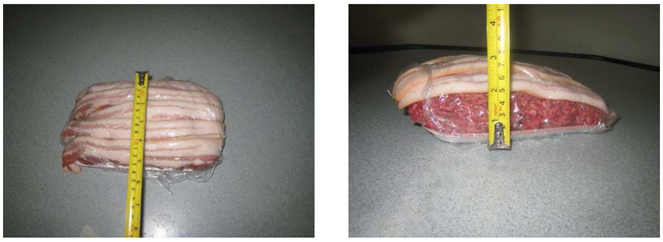


(a)

(b)

Fig. 1. Piezo audio signals. (a) Audio signal for the full key. (b) Acoustic signal (in meat).

2) Valid Bit Detection and Bit Decoding: We detect the beginning of the piezo beep in the signal using signal-processing tools. In particular, to determine if a certain signal region is a potential piezo beep, we examine the signal using a window size of 67 samples and perform an FFT to produce the spectrum of each signal region. We then calculate the energy of the main frequencies intervals (1.67–2.68 kHz and 2.68–3.35 kHz). If either of the energies (which are equal to the square sum of the FFT coefficients during this interval) is above a certain threshold, we consider this signal a valid piezo beep. (We set the threshold to be 40% of the energy of the whole bit.)



(a) (b)

Fig. 2. IMD setup. (a) Meat (top). (b) Meat (side).

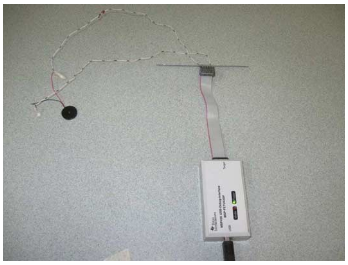
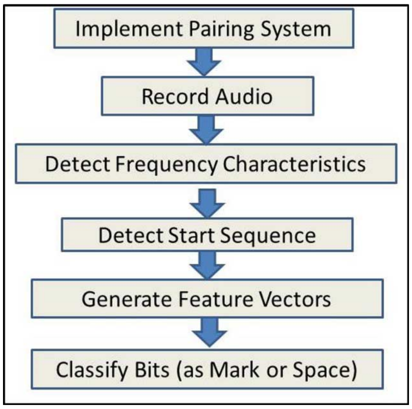


Fig. 3. Piezo-WISP setup.



Implement Pairing System

Record Audio

Detect Frequency Characteristics

Detect Start Sequence

Generate Feature Vectors

Classify Bits (as Mark or Space)

Fig. 4. The stages involved in the attack.

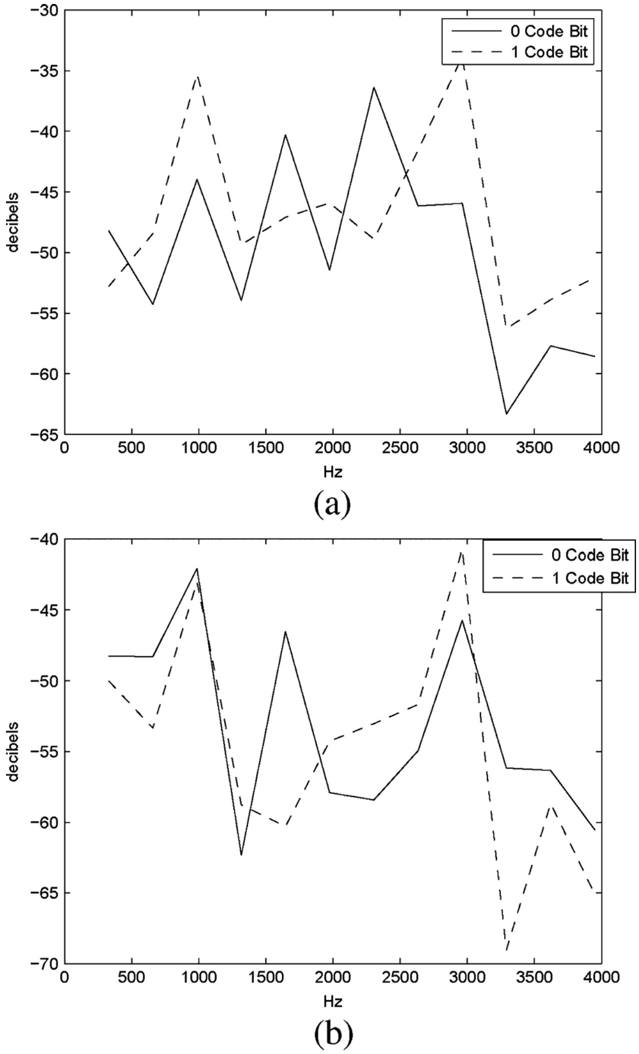


Fig. 5. Spectrum. (a) Open environment. (b) Piezo inside meat.

To further classify each beep in the beginning sequence to the correct digital binary bit, we calculate the ratio between the main piezo frequencies (2.3 kHz/2.9 kHz FFT values). We compare the ratio to a threshold (which is set to 0.5 in our case, as the 2.9 kHz frequency is more pronounced for the ‘1’ bit than the 2.3 kHz frequency is for the ‘0’ bit) and classify the signal.

3) Start Sequence Detection: We perform the start sequence detection in three steps: Finding the first potential bit, synchronizing the bit beginning (which is essential for correct bit decoding) and detecting the rest of the bits in the start sequence.

Finding first potential bit: To detect a potential key beginning, we processed bit-length signal regions until a potential valid bit was found. Then, we reduced the step size to 1 sample and searched for the first bit in the signal. Since we know that the first bit in our preamble sequence is the ‘0’ bit we chose the region with the highest frequencies related to this bit (in the 1.67-2.68 kHz interval).

Bit synchronization: To further perfect our start-bit detection, we used signal energy analysis when detecting the first bit with higher energy level (which corresponds to the ‘1’ bit value). Specifically, we chose a window size of 0.75 ms and a step size of 1 sample and calculated the signal energy within these regions. If the energy is higher than a specific threshold, (we set the threshold to 20% of the energy in the whole 3 ms bit) we mark the first sample in this region as the beginning of the bit.

Detecting following bits: Since the piezo emits the bits continuously with no gap/delay between them, we use a constant window length that starts right at the end of the previous bit region to extract the signal for each consecutive bit. We classify each consecutive bit until we locate the start sequence. It is expected that the start sequence would be the same for all piezo elements (unlike the key, which is random). Therefore, the eavesdropper may know its value ahead of time. Alternatively, the eavesdropper can detect the characteristic frequencies for the two binary bits and use energy analysis on the first high-frequency bit to detect its exact bit start.

4) Feature Vectors Generation: For decoding the full key, we explore the use of machine learning classifiers. To utilize these classifiers, we create two feature files that can be used separately: FFT based features and MFCC features. The FFT-based features are extracted by using a constant bit-size window of 67 samples for each bit and performing an FFT on the bit signal. We also create a separate MFCC feature for each bit. We use a 40-channel filter bank and generate 13 MFCC values for each bit. We use these features as input to our classifier to distinguish between the ‘0’ and ‘1’ bits.

TABLE I
CORRECTNESS OF SUPERVISED AND UNSUPERVISED METHODS BOTH FOR FFT AND MFCC FEATURES IN 3-FEET DISTANCE BETWEEN MICROPHONE AND PIEZO

	Supervised Methods			**Unsupervised Methods**			
Features	FFBP	PNN	LC	KMeans	EM	FF	MDB
FFT	97.22%	56.94%	97.34%	76.50%	97.80%	57.06%	76.74%
MFCC	99.88%	99.54%	99.77%	98.38%	99.65%	67.36%	99.42%

5) Classifiers: As discussed in Section V-D, each recording had 144-bit long keys. For each bit, there were 34 FFT features and 13 MFCC features. The resulting feature vectors are then used with two different types of classifiers—supervised [13] and unsupervised [7].

Supervised Classifiers: In a supervised learning method, the classifier is built based on training data; the target of the classifier is to predict the output of test data. In the context of IMD eavesdropping, the adversary may learn the key corresponding to some of the transmission sessions (e.g., by using the same transmitting device or a similar setup), create the training data set and build the classifier. On future sessions, the adversary can simply sniff upon the audio channel and decode the key using the classifier.

We labeled each feature vector with corresponding bit values (‘0’ or ‘1’) and built the training data set using half of the total recordings. We implemented the classifiers in Matlab.

We calculate the average correctness (%) of the classifier output from the five recordings as follows:

$$\text{Average correctness} = \frac{\#\text{ of correct bits}}{\#\text{ of bits transmitted}} \times 100\%$$

The decoding result of supervised learning algorithms for both FFT and MFCC features (for 3 feet distance between microphone and piezo) are depicted in Table I.

The results show that MFCC features always performed better than FFT features, which is also in line with the findings of [32]. Most methods yielded an accuracy of 99–100% for MFCC features. The Linear Classifier had the highest accuracy for the FFT features, resulting in 97.34%. This suggests the feature vectors location in the space can be separated with a hyperplane which divides the space accurately. For more details, please refer to the conference version of this paper [10].

We speculate that the reason that MFCC features provide better results relative to FFT features is that MFCC use the Mel Scale [30] of frequencies. In this scale, the contribution of the lower frequencies is maximized relatively to the higher frequencies. Since the frequencies of interest in our case lie in the 0–3 kHz range, their contribution is maximized which will improve classification.

Unsupervised Classifiers: Unsupervised classifiers can be used in situations where training data is not available or possible to generate. Since the bits are binary, the classifiers divide the test data into two clusters (‘0’ or ‘1’) and each cluster is assigned a label. Therefore, in the IMD eavesdropping setting, the adversary may decode the key using the unsupervised clustering methods (without previously labeled training data). We used KMeans, Expectation-Maximization (EM), Farthest First (FF), and Make Density Based (MDB) clustering algorithms implemented in Weka [8]. A total of 5 recordings were used to calculate the accuracy of each of the clustering algorithms. The results of unsupervised learning algorithms are depicted in Table I. They show that MFCC features have better performance than FFT features, similar to our results using supervised learning. For FFT features, EM performs better than other clustering algorithms, providing 97.8% accuracy. For MFCC features, all methods provide good results (99%–100% correct detection).

6) Effect of Distance: We experimented with IMD eavesdropping using a PC microphone from different reasonable distances between the piezo and microphone. We took 10 recordings for each of the 8 distances—close by (less than 1 foot), 1–2 feet, 3 feet and 4–6 feet. Half of the recordings were used for building training data set when using supervised learning algorithms and the rest of the recordings were used for testing both supervised and unsupervised algorithms.

Overall, supervised classifiers seem to have better performance up to 4 feet distances (>90% correctness). However, there is high degradation of correctness between 5–6 feet, which prompts us to consider a parabolic microphone. (The full results appear in [10]).

E. Eavesdropping Using Parabolic Microphone

We further investigated if our techniques will work even from farther distances (up to 12 ft). Since parabolic microphones are currently widely available and have become less expensive (we used a $28 microphone which is sold in toy stores), we believe this is a realistic threat that may increase the vulnerability of IMD Pairing. We further explored the vulnerability of the system to an eavesdropping attack using only signal processing methods (without utilizing classifiers). To this end, we took recordings using a parabolic microphone with the same setup (piezo beeping inside meat) from a few distances up to 12 feet. We examined the signal spectrum and found that while the lower frequencies got blurred, we were able to use the spectrum in the higher frequencies band (6.5 kHz–7.5 kHz) instead for detecting the ‘1’ bit accurately. We created a curve with the sum of the frequencies in this interval and threshold it (we set the threshold to 0.5 of the maximum value of that curve over the key recording) to detect the ‘1’ bits. We found that even at 12 feet, we were able to distinguish between the ‘0’ and ‘1’ bits with a probability of over 80%. This emphasizes the vulnerability of IMD Pairing from farther distances.

VI. EAVESDROPPING PIN-VIBRA

A. PIN-Vibra Encoding

In our eavesdropping experiments, we used the original prototype implementation of the PIN-Vibra method [24]. For encoding a PIN into vibrations, a simple time interval based

ON-OFF encoding was employed that used a four-digit PIN which is equivalent to 14 bits of binary data. Three additional bits ("110") were used as a start sequence to indicate (to a valid decoder) the beginning of the transmission. Each '1' bit was converted into a vibration that lasts for 200 ms and each '0' bit was converted to a 200 ms interval of stillness (i.e., no vibration). Thus the PIN was transmitted using 17 bits resulting in a total transmission time of $17 \times 200\ \mathrm{ms} = 3.4$ seconds.

B. Eavesdropping Challenges

As discussed above, PIN-Vibra is based on a constant bit length of 0.2 sec. In each bit duration, the phone either vibrates or there is a sleep period. The phone vibration has very low sound amplitude which makes it harder to distinguish the vibrations from random noise. Furthermore the vibration might last for a few consecutive bit periods with no gap between the periods which makes it impossible to detect the beginning of each vibration separately. Therefore, once we detect the beginning of the phone vibration, we use a constant bit length to extract the signal of each consecutive bit and decode it. We found that, similar to the IMD setup, accurate detection of the first bit is essential to correctly detect the PIN.

We do note that the fact that the signal is longer (200 ms versus 3 ms) and that the '0' bit is marked by sleep (as opposed to a different frequency in IMD Pairing) makes it somewhat easier to eavesdrop upon PIN-Vibra. However, unlike the piezo, which is intended to generate audio and has specific noticeable frequency peaks for each bit, the phone vibration audio signal is a by-product (of vibration) and is not centered around one specific frequency. This makes it harder to distinguish the signal from random background audio sounds.

To solve this problem, we start by recording vibration from a close range and characterizing the audio signal by finding the audio frequencies associated with vibrations. Then, we take recordings from farther and try to locate regions in which these frequencies are more obvious. Unlike IMD eavesdropping, we found that we need to examine two wider frequency intervals to be able to detect the vibration. This allows us to detect with high probability the existence of a vibration while eliminating random noise.

Another challenge in PIN-Vibra eavesdropping arises from attempting to eavesdrop using a standard (inexpensive) PC microphone. Since noise cancellation algorithms are commonly used today on standard PC microphones (such as the one we used for our experiments), it makes it harder to eavesdrop from a distance. Noise cancellation mechanisms in microphones attempt to distinguish between audio coming from close by and audio coming from a distance. In this case, frequencies of audio coming from a distance will be filtered out (making it easier to hear the close-by audio). Therefore, the further our phone is from the microphone, the higher the likelihood the system may attempt to cancel it out.

Our experiments showed that the vibration spectrum indeed became very blurred when we took recordings from a few feet away (versus the close by recordings). When comparing the phone vibration with the IMD sound, we find that the amplitude is lower and the spectrum stretches over two wide frequency intervals. Therefore, we suspect that the phone vibration is more vulnerable to the effects of the noise cancellation mechanism which causes larger signal blurring when captured from a few feet away.

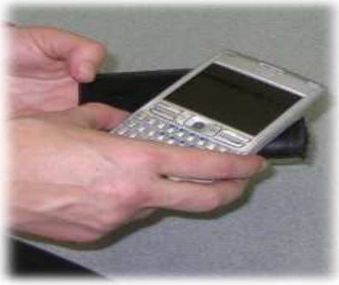


Fig. 6. Pin-Vibra setup.

C. Setup

For our experiments, we used the same components discussed in Section V-B. Additionally, we used a Nokia mobile phone model E61[3], the same model used in the experiments reported in [24].

The following steps were followed to capture the recordings:

- The phone was programmed to produce a random 14-bit value ("01000111010010" or 4562 decimal PIN) prefixed with the beginning sequence "110" (using the original PIN-Vibra prototype [24]).
- The phone was held next to a wallet (the two touched each other) and set to send the PIN (Fig. 6). This was done to emulate RFID authentication as described in [24]).

Sound Level Measurements: We measured the audio intensity of our mobile phone vibrations. We found that when measuring very close to the phone (a few cm away) the reading was 70 dB SPL and reduced to 62 db SPL from 3 ft away. We also found that touching the phone with a wallet produced no difference in the measurements (which indicates that there was no dampening effect due to touching the phone with the wallet). The measured SPL of the phone vibration is equivalent to quiet conversation (60 dB) and can be heard by the human ear. However, we observed that the overall vibration key signal sounds like one continuous vibration and due to the short duration of each bit, it is not possible to distinguish between a vibration bit period and a "sleep" period just by manually listening to the signal. Therefore, we attempt to utilize signal processing methods to detect the beginning of the key.

D. General Approach

The PIN-Vibra algorithm is similar to the IMD scheme in that they both use a beginning sequence to mark the start of the key (adding a beginning sequence is part of the original PIN-Vibra algorithm). Both schemes use a constant bit length to send each consecutive bit with no gap between the bits. Therefore, we utilize an algorithm similar to the one we used for the IMD eavesdropping (Fig. 4).

E. Audio Signal Decoding Algorithm

1) Frequency Characteristics: We start by characterizing the phone vibrations as recorded from a close distance (a few cm

[3] Specifications are available at: http://www.forum.nokia.com/devices/E61/

TABLE II
CORRECTNESS OF SUPERVISED AND UNSUPERVISED METHODS BOTH FOR FFT AND MFCC FEATURES IN 3-FEET DISTANCE FOR PIN-VIBRA ATTACK

	Supervised Methods			**Unsupervised Methods**			
Features	FFBP	PNN	LC	KMeans	EM	FF	MDB
FFT	90.76%	94.96%	94.96%	92.44%	99.16%	83.19%	92.44%
MFCC	100.00%	100.00%	100.00%	100.00%	100.00%	99.16%	100.00%

away from the phone). We examined the vibration spectrum for our phone and found that the frequencies stretch over two intervals: 125–250 Hz and 1.1–1.5 kHz. Therefore, in order to detect the vibration accurately, our algorithm can not rely on detecting one specific frequency but rather has to look at a wider range of frequencies.

To correctly decode the signal, we need to first determine the period which gives the best frequency spectrum within one vibration. Each bit is 0.2 seconds in duration. However, careful examination of the recorded bit shows that the main vibrations occur in the middle three quarters of the bit. Therefore, we use a window size of 150 ms when searching for the first bit vibration.

2) Start Sequence Detection: As mentioned previously, to allow for correct detection of the start of the PIN transmission (by a valid decoder, i.e., an RFID tag with an accelerometer), the PIN-Vibra method [24] includes a 'start' sequence equal to "110" as a prefix to the PIN. We begin by looking for this start sequence. Since the sound emitted during the phone vibrations is of very low amplitude, detecting the beginning sequence ensures that the PIN is decoded from its beginning and helps distinguish the PIN vibrations from other sounds that may be emitted by the phone.

Finding first potential bit: We calculated the spectrum using a 150 ms window and a step size of 25 ms between the beginning of the consecutive signal regions. We performed a Fast Fourier Transform (FFT) for each region and calculated the sum of the FFT values over the two vibration frequency intervals (125–250 Hz and 1.1–1.5 kHz). We compared these sums against set threshold levels to detect the vibration for the frequency spectrum.

Bit synchronization: After the first vibration was detected, we used energy calculations to improve the detection of the beginning of the key. To this end we examined all the periods of 0.1 seconds within the discovered vibration and chose the part with the highest energy as the middle of our positive bit (we subtracted a quarter size bit length from the start of the region to mark the beginning of the first bit).

Detecting following bits: We note that the phone vibration bits are sent in a consecutive order, and that the vibrations last for 0.2 seconds (with a "slack" period of 5 ms between the vibrations). Therefore, once the first potential vibration is found we continue by decoding the two following bits as either '1' (vibration) or '0' (sleep). This is done by calculating the FFT for the 0.2 second window for each consecutive bit and repeating the same decoding procedure (utilizing two curves of FFT sums in the 130–25 Hz and 1.1–1.5 kHz frequency regions) until the beginning sequence is found.

The frequency sum curve threshold was initially set to the maximum sum of FFT coefficients for the recording, and than lowered by 10% in each iteration until the start sequence was found. For the two curves, the threshold for the second curve was set to twice the threshold for the first curve. We note that the vibration is very low and hard to distinguish from the background noise. Therefore, setting a threshold that's too low will detect random noise as vibration.

3) Feature Vectors Generation: Upon detecting the start sequence, we create both MFCC and FFT feature files for each following bit. We feed these input vectors through machine learning classifiers (we will discuss these in Section VI-F). As a result we construct a 17-bit binary data. We extract the beginning sequence and convert the 14 bits into a 4-digit decimal PIN.

F. Classifiers

PIN-Vibra eavesdropping yields feature vectors corresponding to a 17-bit PIN/key both for FFT and MFCC. FFT feature vectors have 12 columns and MFCC feature vectors have 13 columns, and both of them have 17 rows as per the length of the key. We apply supervised and unsupervised learning algorithms and decode the key from the feature vectors (similar to the methods used in Section V-D5).

Result of supervised and unsupervised algorithms for compromising the key by audio eavesdropping on PIN-Vibra method are depicted in Table II. We found that MFCC works as a better feature than FFT and almost all algorithms work perfectly (with 100% correctness) except the unsupervised FF algorithm. Among all of them, unsupervised EM seems to be a winner for both FFT and MFCC features.

VII. EAVESDROPPING BEDA

A. Encoding and Decoding

In the BEDA scheme [27], one device vibrates (or blinks) for 0.5 seconds. The user is required to press the button on the other device synchronously whenever the first device vibrates (or blinks). When the protocol starts, the first device generates a short (21-bit) random secret key (a password or PIN) and provides a total of eight signals. Each signal is generated by the idle time determined by the i-th 3-bit segment of the secret. Therefore, the time between each consecutive vibrations (or blinks) is equal to the value of these 3-bits segment in seconds. The receiving device measures the intervals between successive button presses in milliseconds and rounds it to the closest full second. Each of those rounded integers is translated into 3-bit segment to reconstruct the full key.

B. Challenges and General Approach

We attempt to eavesdrop on the Blink-Button, Vibrate-Button and Button-Button BEDA methods. We note that eavesdropping over button presses in the Blink-Button and Button-Button

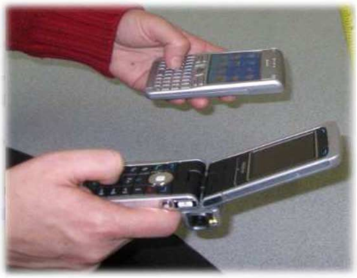


Fig. 7. BEDA setup.

schemes is somewhat similar to keyboard eavesdropping as discussed in [2], [32]. However, when we examine the audio signal, we find that the mobile phone button pressing is much quieter than the keyboard on the laptop computer we used and therefore detecting the click may be a harder task.

We note that the BEDA method is different from IMD Paring and PIN-Vibra in that the key is not sent in a binary form. Instead, the key is constructed from the time differences between vibrations and button presses. Therefore, unlike IMD Pairing and PIN-Vibra we do not have a constant "bit length" which defines each bit and therefore we can not classify each signal window. Rather, the BEDA method requires the eavesdropper to detect each vibration and button press separately and calculate the duration between them. Therefore, we only use signal processing methods to detect the beginning of each time period and decode the key from it *without the need for using a classifier* in this case.

C. Setup

For our experiments, we used the same components we discussed in Section V-B. Additionally, we used one Nokia mobile phone model E61 (as also used in the PIN-Vibra setup), and one Nokia N90[4]. Both of these models were used in the experiments reported in [27].

Vibrate-Button and Blink-Button: In the implementation of these methods, the E61 served as the server (the one that vibrates or blinks) and the N90 as the client (used for button pressing).

The following steps were performed as part of our experiments:

- The server phone was programmed with a randomly generated 5-digit (or 6-event) secret key. Each digit specified the difference between every two vibrations (or blinks) in units of half a second.
- The server phone was set to transmit the secret key. Each time the server system vibrated (or blinked), the user clicked on a button on the client system (Fig. 7).

Button-Button: For the implementation of this scheme, both the E61 and the N90 were used together by the user to exchange the code, by pressing and releasing a button on both phones simultaneously. The buttons were pressed and then released three times producing a total of 6 events.

The phone application calculates each key digit by measuring the time difference between each two events in units of 0.5 second (generating a random 5-digit key).

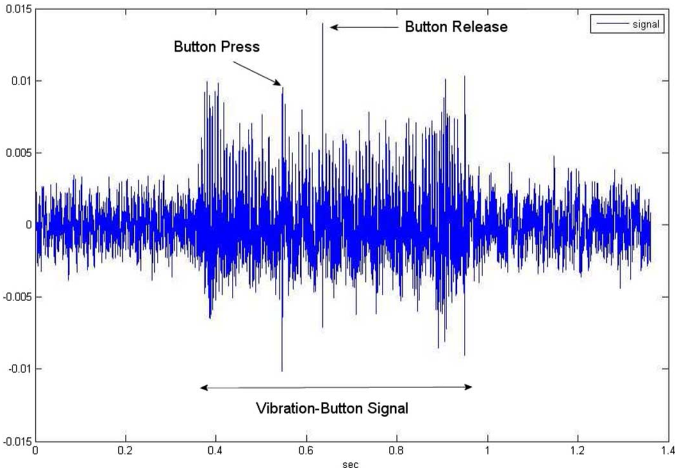


Fig. 8. Audio signal (vibrate-button).

Sound Level Measurements: We measured the signal SPL volume and found that the button pressing on our N90 phone measures 64 db SPL from a close by (a few cm away). When attempting to measure the volume from 2 ft distance, we found that the clicks were too low to be registered by the sound level meter (Radio Shack model 33-2055). The E61 sound levels appear in Section VI-C.

D. Vibrate-Button

In Vibrate-Button, the user needs to press a button on one device when the other device vibrates. Both button pressing and vibration produce a very low amplitude sound that makes eavesdropping challenging. As discussed in Section VII-B, the sound that the button emits is very short in duration relative to the vibration and overlaps it, which makes it hard to distinguish from the vibration, depending on the location of the eavesdropping device. The vibration eavesdropping challenges are similar to the ones described in the PIN-Vibra scheme (Section VI-B). The main problems arise from the fact that the mobile phone audio frequencies stretch over two intervals and the attempt to eavesdrop from a distance with a standard PC microphone (which regards low amplitude sounds as noise and attempts to cancel them).

When analyzing the Vibrate-Button audio signal (shown in Fig. 8), we note that the vibration lasts around 0.5 seconds while the button click has one main observable peak which lasts only 2–3 ms and overlaps the vibration. Since the code is determined by the time differences between the vibrations, our techniques concentrated on detecting the server vibration duration (which subsumes the button pressing).

Since we found that typically the recording spectrum (Fig. 10) is not even throughout the duration of the vibration, we divided the test interval into smaller parts. Specifically, we found that using a window size of 125 ms and calculating the spectrum for these windows produced better results than using a window size of 0.5 seconds. We create the signal spectrum by calculating the FFT for the signal using a step size of 62.5 ms (and a 125 ms window). We noticed that the vibration spectrum is higher over the range 1–7 kHz. We therefore calculated the sum of the frequencies over this range and used a threshold to determine potential vibration regions. We plot the resulting curve in Fig. 9.

[4] Specification are available at: http://www.forum.nokia.com/devices/N90/

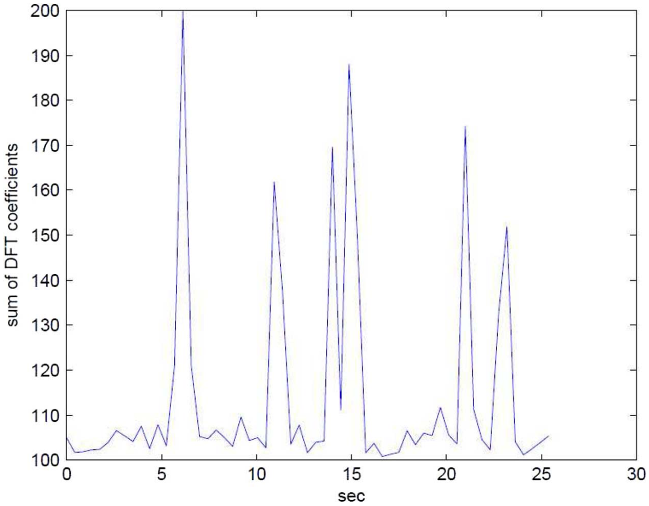


Fig. 9. FFT sum (blink-button 5-digit key).

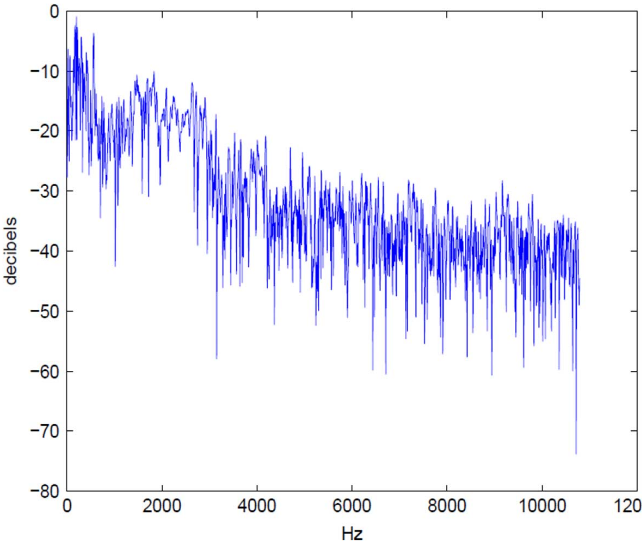


Fig. 10. Spectrum (vibrate-button).

For each test signal, we confirm the vibration only if at least two windows within a range of five were positive. This resulted in good vibration detection and removal of "random noise" in the recording. The code was extracted by computing the difference between the discovered vibrations in 500 ms units.

To find a proper threshold for the FFT coefficients curve, we start from a high value (equal to the maximum curve for the recording) and reduce the threshold by 10% each time until we detect 6 events.

E. Blink-Button

For the Blink-Button scheme, we recorded the sounds of button pressing on the client phone. When examining the button click period, we observed that each click on the mobile phone typically produced a sharp vibration over a short period (about 2–3 ms) and a second spike (lower), less than half a second apart. This corresponds to the press and release of the button. To detect the button click, we chose a window size of 10 ms and an overlap of 2.5 ms. We "windowed" our signal with a Hamming window [12] and performed FFT on the resulting signal. This method is similar to the one described in [32] used to decode keyboard presses, and our observation of the signal confirmed it is also suitable for our mobile phone key pressings. We summed up the FFT values over the 1–11 kHz frequencies and "thresholdized" this sum to detect the recorded vibrations (an example of this curve is shown in Fig. 9).

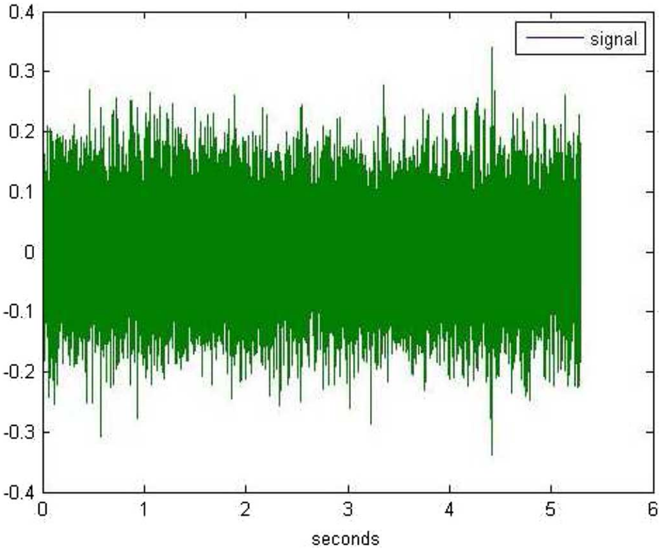


Fig. 11. Button-button signal.

To verify the button click and eliminate background sounds, the program first confirms the existence of an actual vibration. The code was then calculated by computing the difference between the verified button clicks (in 500 ms units).

To find a proper threshold, we start from a high value (equal to the maximum curve for the recording) and reduce the threshold by 10% each time until we find 6 events (a 5-digit code).

F. Button-Button

For the Button-Button scheme, we record the sound of button pressing on both phones. The scheme is similar to the Blink-Button method since only the phone key clicks are emanating audio signals. However, unlike the Blink-Button scheme, in this case the attacker needs to detect accurately both the press and release events in order to calculate the code. Since the release is significantly quieter than the press, the code detection becomes even more challenging. In addition, we get two button clicks (from two devices) that either overlap or are very close to each other, which further complicates the ability to detect accurately the time of each press/release.

To implement the attack, we eavesdrop on the communication from 3 ft distance (Fig. 11). To detect the button click, we choose a window size of 10 ms and an overlap of 2.5 ms. We window our recorded signal with a Hamming window and perform FFT on the resulting signal (Fig. 12).

Since the Button-Button method requires that we distinguish between both a press event and a release event, we further need to locate frequency features which can be used to detect both successfully. Examination of the signal spectrums shows that the press is best detected by summing the frequency features over the range 0.4–11 kHz (as done for the previous BEDA variants). However, the audio frequencies of the button release signal are concentrated in the 1–3 kHz range. Our tests further showed that summing only the frequencies in the latter range

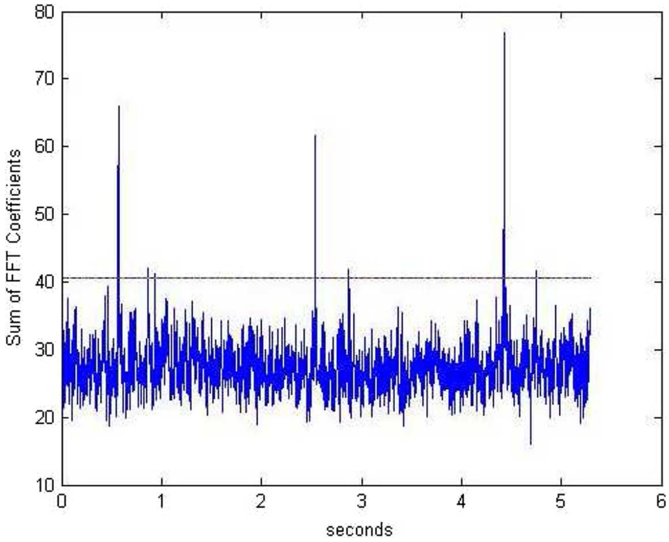


Fig. 12. FFT sum (button-button 5 digit key).

for both the press and release produced better overall results. We therefore calculate this sum for each window spectrum and compare it to a threshold to detect the existence of an event (button press or release).

To further pinpoint the exact location of the start of the button event, for each event of length 10 ms, we further examine signal windows of 2.5 ms within this period (using step size of 1 sample) and locate the region which has the maximum energy. We choose the beginning of this region as the beginning of each Button-Button event.

To detect the sent code, we calculate the time difference between each two consecutive events, divide it by units of 0.5 sec and round it to the nearest integer. However, to further improve the detection, in cases where the resulting time difference is in the middle between two integers (specifically, when the resulting numbers has the digits 4 or 5 after the decimal point), a second option is chosen. For example, if the resulting difference is 2.51, then as the first option the integer 2 is chosen, and as a second option the integer 3 is chosen for the code. Since this would only be considered in cases where the first code is tested and found incorrect, this further raises the detection capability while only requiring part of the codes to be retested.

To set the threshold for the FFT coefficients curve, we start from a large value (equal to the largest sum of the FFT coefficients over the recording). We then lower the threshold by 5% in each iteration until we detect 6 events (a 5-digit code).

G. Results

The Vibrate-Button recordings were made from 3 ft distance from the vibrating phone (around 4 ft distance from the client phone). For Vibrate-Button eavesdropping tests, we took 20 recordings of the phone vibrations using a PC microphone. For 19 of the recordings, we succeeded in fully decoding the key. In one of the recordings, only three of the five digits were decoded correctly. Therefore, our overall success rate was 98%.

For Blink-Button eavesdropping tests, we took 20 recordings from 3 ft away from the client phone. We received results similar to the Vibrate-Button test. Only one of our recording was not fully decoded (with three of the 5 digits decoded correctly) and our overall decoding rate was 98%.

For Button-Button experiments, we took a total of 60 recordings from 3 ft distance. Of these recordings, 44 were detected correctly fully. For the rest of the recordings, part of the bits were detected correctly, resulting in an overall success rate of 82% (the success rate is calculated as described in Section V-D5). These results show that even in cases where the event is a single button press or release (which is inherently harder to detect than the button press), the code can be detected with high probability.

VIII. Discussion and Conclusion

Technical Novelty of Our Paper: First of all, this paper is the first to study acoustic eavesdropping and side channel attacks against device pairing mechanisms. In so doing, we overcome a few technical challenges that were not encountered before, to the best of our knowledge. For the IMD case, we are classifying continuous data where there is no separation between the bits. This makes accurate synchronization of the beginning of the first bit critical for correct decoding. We also study eavesdropping in a realistic setting (from distances up to a few feet away) and compare the results from different distances using very inexpensive equipment (PC microphone). Previous research on keyboard acoustic emanations concentrated on recordings from a single close by distance or used special equipment (parabolic microphone) for farther recordings.

Our research also uses classifiers and signal processing tools for binary classification of audio data (i.e., detect the mark and spaces). Our results show that binary classification may be easier, which suggests choosing nonbinary coding schemes may provide higher security.

Classifiers and Neural Networks: For our IMD and PIN-Vibra decoding, we use signal-processing based methods to decode the beginning sequence and neural networks to decode the rest of the signal. Neural networks provide higher-level analysis than regular processing (which is based on energy or sum of features). Therefore, we utilized them in cases of known bit length. However, in order to be able to correctly create the input features for the neural networks, the accurate signal beginning location needs to be found. Therefore, signal-processing methods are needed to pinpoint the exact sequence beginning.

Implications of Our Attacks: The attacks we demonstrated on IMD Pairing, PIN-Vibra and the BEDA variants can be accomplished with a high accuracy by using inexpensive off-the-shelf equipment, such as PC microphones, and existing signal processing techniques and/or machine learning classifiers. We successfully executed our attacks from a distance of up to 5–6 ft for IMD Pairing and 3 ft for PIN-Vibra and BEDA. Our overall accuracy was 97–100% for IMD Pairing, 100% for PIN-Vibra, 98% for the Vibrate-Button and Blink-Button BEDA variants and 82% for the Button-Button method (for eavesdropping up to 3 ft). We summarize the techniques used in Table III.

As described in III, execution attacks can occur in many scenarios[5]. Moreover, for the IMD setup, we also explored eaves-

[5] The adversary can also eavesdrop over the wireless radio channel to detect as to when the pairing process is initiated. Note that pairing protocols would typically precede with a certain negotiation phase, as is customary for key exchange protocols (e.g., IKE).

TABLE III
METHODS USED TO DETECT AND DECODE BITS

Scheme	Signal Processing	Window Size (ms)	Classifiers Used
IMD Pairing	Energy+Frequency Ratio	3	Supervised/Unsupervised
PIN-Vibra	FFT Coefficients sum	150	Supervised/Unsupervised
Vibrate-Button	FFT Coefficients sum	125	None
Blink-Button Button-Button	FFT Coefficients sum	10	None

dropping using a parabolic microphone and were able to achieve reasonable accuracies from a distance of 12 ft; we anticipate similar results when working with a parabolic microphone for distant eavesdropping on PIN-Vibra and BEDA variants.

We remark that compromising IMD Pairing and PIN-Vibra is an easier task compared to attacking BEDA. This is because the former schemes transmit the key over the underlying OOB channel, whereas the latter only transmits a password using which the two devices derive the key via a PAKA protocol. This implies that even after eavesdropping over the password in BEDA, the adversary would still need to act as a man-in-the-middle (and fast enough) to be able to compromise the security of the protocol.

In the IMD setup, the adversary can always verify the correctness of the key that was eavesdropped once equipped with a known plaintext-ciphertext pair. For PIN-Vibra and BEDA, the adversary can try to use the PIN/password that was eavesdropped to unlock the RFID tag or the phone, and launch the man-in-the-middle attack, respectively. The adversary can compromise the security of these approaches with a high probability (as shown by our high accuracy rates), much higher than the original success probability of 2^{-k} for a k-bit password. We note, however, that learning the PIN only undermines the security of PIN-Vibra against impersonation attacks (e.g., in case of the tag theft); the method still provides strong protection against unauthorized reading and some relay attacks. This is because the attacker needs direct physical access to the tag in order to unlock it (by "touching" the phone with the tag), and will not be able to read the tag otherwise, even if it knows the PIN.

Hardware Variations and Attack Techniques: The attacks we developed included general signal processing based algorithms and/or classifiers and were not hardware specific. For IMD eavesdropping, we used spectrum analysis and energy calculations to differentiate between two piezo frequencies and machine learning methods to further classify all the bits automatically. These attacks can be used on any piezo hardware without being limited to specific FSK frequencies or piezo amplitude. Furthermore, since the method is based on the piezo sending the key via audio signal, an attacker can always use a higher-end microphone to record the audio emanations (even if the piezo is relatively quiet) and still use the same techniques. Similarly, for PIN-Vibra and BEDA Vibrate-Button eavesdropping attacks, we use spectrum analysis tools that do not depend on a specific frequency (specifically, the vibration in our tests extended over a large frequency interval). Therefore, this attack can be used on any phone model. Since most mobile phones would emanate some sound—which is even audible to the human ear—when vibrating, we expect our attacks can work on any model phone. In case of the BEDA Blink-Button and Button-Button attacks, since the audio emanations result from both the finger hitting the key and the key hitting the underlying plate beneath the keypad, typically both events will cause acoustic emanations regardless of the specific model of phone used (similarly, all computer keyboards tested in [2] emitted distinct acoustic emanations). Since we only try to detect the existence of each button click (and not which button), we do not need a detailed signal spectrum and can eavesdrop on even a low-volume signal.

Based on our results and discussion above, we can conclude that all three approaches analyzed in this paper provide a weaker level of security compared to what was originally assumed or is desired for the pairing operation. Designing an AS-OOB pairing method—resistant to eavesdropping—thus appears to be a challenging research problem and an avenue for further work. We feel that the broader impact of our work lies in raising awareness that some pairing mechanisms which produce audio emanations are vulnerable to eavesdropping attacks, and in motivating the need for observation-resilient pairing mechanisms for constrained ubiquitous devices.

Open Problem: Given the success of our acoustic eavesdropping attacks, our conclusion is that relying upon the secrecy of audio channels is dangerous (since this channel is inherently not secret given that audio travels in all directions) and should be avoided while developing pairing mechanisms. User-friendly and eavesdropping-resilient pairing is therefore a challenging problem that will need future investigation.

ACKNOWLEDGMENT

The authors would like to thank Md. B. Uddin for his work related to classifiers, E. Uzun for providing them with the BEDA implementation, and S. Halevi, J. Voris, and the TIFS journal anonymous reviewers for their feedback.

REFERENCES

[1] B. Acohido, New security flaws detected in mobile devices [Online]. Available: http://www.usatoday.com/tech/news/story/2012-04-08/smartphone-security-flaw/54122468/1

[2] D. Asonov and R. Agrawal, "Keyboard acoustic emanations," in *Proc. IEEE Symp. Security and Privacy*, 2004, pp. 3–11.

[3] M. Backes, M. Durmuth, S. Gerling, M. Pinkal, and C. Sporleder, "Acoustic side-channel attacks on printers," in *Proc. Usenix Security Symp.*, 2010, p. 20.

[4] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *Proc. Network & Distributed System Security Symp.*, San Diego, CA, USA, 2002.

[5] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using Diffie-Hellman," in *Advances in Cryptology-Eurocrypt*. New York, NY, USA: Springer, 2000, pp. 156–171.

[6] R. Briol, "Emanation: How to keep your data confidential," in *Proc. Symp. Electromagnetic Security for Information Protection (SEPI'91)*, Rome, Italy, 1991.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Unsupervised Learning and Clustering*. Hoboken, NJ, USA: Wiley-Interscience, 2001.

[8] R. Evans, "Clustering for Classification, Computer Science," Master's Thesis, University of Waikato, Hamilton, New Zealand, 2007.

[9] C. Gehrmann, C. J. Mitchell, and K. Nyberg, "Manual authentication for wireless devices," *RSA CryptoBytes*, vol. 7, no. 1, pp. 29–37, Spring 2004.

[10] T. Halevi and N. Saxena, "On pairing constrained wireless devices based on secrecy of auxiliary channels: The case of acoustic eavesdropping," in *Proc. 17th ACM Conf. Computer and Communications Security (CCS'10)*, 2010, pp. 97–108, ACM.

[11] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proc. IEEE Symp. Security and Privacy*, 2008, pp. 129–142.

[12] R. Hamming, "Hamming window: Raised cosine with a platform," in *Digital Filter*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1977.

[13] S. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica J.*, pp. 249–268, 2007.

[14] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun, "Caveat emptor: A comparative study of secure device pairing methods," in *Proc. Int. Conf. Pervasive Computing and Communications (PerCom)*, 2009, pp. 1–10.

[15] C. V. Lopes and P. Aguiar, "Acoustic modems for ubiquitous computing," *IEEE Pervasive Comput., Mobile Ubiquitous Syst.*, vol. 2, no. 3, pp. 62–71, Jul./Sep. 2003.

[16] J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," in *Proc. IEEE Symp. Security and Privacy*, 2005, pp. 43–56.

[17] R. Prasad and N. Saxena, "Efficient device pairing using "human-comparable" synchronized audiovisual patterns," in *Proc. Applied Cryptography and Network Security*, 2008, pp. 328–345.

[18] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[19] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proc. ACM Conf. Computer and Communications Security*, 2009, pp. 410–419.

[20] V. Roth, W. Polak, E. Rieffel, and T. Turner, "Simple and effective defenses against evil twin access points," in *Proc. ACM Conf. Wireless Network Security (WiSec)*, 2008, pp. 220–235.

[21] A. Sample, D. Yeager, P. Powledge, and J. Smith, "Design of a passively-powered, programmable sensing platform for UHF RFID systems," in *Proc. IEEE Int. Conf. RFID*, 2007, pp. 149–156.

[22] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel," in *Proc. IEEE Symp. Security & Privacy*, 2006, pp. 306–313.

[23] N. Saxena and M. B. Uddin, "Secure pairing of "interface-constrained" devices resistant against rushing user behavior," in *Proc. Applied Cryptography and Network Security*, 2009, pp. 34–52.

[24] N. Saxena, M. B. Uddin, and J. Voris, "Treat 'em like other devices: User authentication of multiple personal RFID tags," in *Proc. Symp. Usable Privacy and Security (Poster Session)*, Mountain View, CA, USA, 2009.

[25] N. Saxena, M. B. Uddin, J. Voris, and N. Asokan, "Vibrate-to-unlock: Mobile phone assisted user authentication to multiple personal rfid tags," in *Proc. 2011 IEEE Int. Conf. Pervasive Computing and Communications (PERCOM'11)*, 2011, pp. 181–188.

[26] A. Shamir and E. Tromer, Acoustic cryptanalysis on nosy people and noisy machines [Online]. Available: http://people.csail.mit.edu/tromer/acoustic/

[27] C. Soriente, G. Tsudik, and E. Uzun, "BEDA: Button-enabled device association," in *Proc. Int. Workshop on Security for Spontaneous Interaction (IWSSI)*, Innsbruck, Austria, 2007.

[28] C. Soriente, G. Tsudik, and E. Uzun, "Secure pairing of interface constrained devices," *Int. J. Security and Networks (IJSN)*, vol. 4, no. 1, pp. 17–26, 2009.

[29] F. Stajano and R. J. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," in *Proc. Security Protocols Workshop*, 1999, pp. 172–194.

[30] S. S. Stevens, J. Volkman, and E. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, pp. 185–190, 1937.

[31] E. Uzun, K. Karvonen, and N. Asokan, "Usability analysis of secure pairing methods," in *Proc. Usable Security (USEC)*, 2007.

[32] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," in *Proc. ACM Conf. Computer and Communications Security*, 2005, pp. 373–382.

Tzipora Halevi is a postdoctoral Researcher in the Department of Computer Science and Engineering, Polytechnic Institute of NYU. She received the Ph.D. degree from Polytechnic Institute of NYU in electrical engineering. Her research area is security. Her work focuses on constrained devices areas, biometric authentication, and cyber-security.

Nitesh Saxena is an Assistant Professor in the Department of Computer and Information Sciences, University of Alabama at Birmingham (UAB), and the founding director of the Security and Privacy in Emerging Systems (SPIES) lab (http://spies.uab.edu/). He works in the areas of computer and network security, and applied cryptography, and has published over 60 papers at top-tier venues in computer science. His current research is externally supported via multiple grants by NSF, Google, Intel, Nokia, and Research in Motion. He has significant experience architecting and leading security programs, and is currently serving as a Codirector of a major security program at UAB.