On Pairing Constrained Wireless Devices Based on Secrecy of Auxiliary Channels: The Case of Acoustic Eavesdropping

Tzipora Halevi
Electrical and Computer Engineering
Polytechnic Institute of New York University
Six MetroTech Center
Brooklyn, NY 11201
thalev01@students.poly.edu

Nitesh Saxena
Computer Science and Engineering
Polytechnic Institute of New York University
Six MetroTech Center
Brooklyn, NY 11201
nsaxena@poly.edu

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. Lack of good quality interfaces on or physical access to certain constrained devices (e.g., headsets, access points, medical implants) makes pairing a challenging problem in practice. Prior work shows that pairing of constrained devices based on authenticated OOB (A-OOB) channels can be prone to human errors that eventually translate into man-in-the-middle attacks. An alternative and more usable solution is to use OOB channel(s) that are authenticated as well as secret (AS-OOB). AS-OOB pairing can be achieved by simply transmitting the key or a short password over the AS-OOB channel, avoiding potential serious human errors.

A higher level goal of this paper is to analyze the security of AS-OOB pairing. More specifically, we take a closer look at three notable prior AS-OOB pairing proposals 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. The first proposal (IMD Pairing [9]) uses a low frequency audio channel to pair an implanted RFID tag with an external reader. The second proposal (PIN-Vibra [20]) uses an automated vibrational channel to pair a mobile phone with a personal RFID tag. The third proposal (BEDA [22]) uses vibration (or blinking) on one device and manually synchronized button pressing on the other device. In particular, we demonstrate the feasibility of eavesdropping over acoustic emanations associated with these methods. Based on our results, we conclude that these methods provide a weaker level of security compared to what was originally assumed or is desired for the pairing operation.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Authentication; C.2.0 [**Computer-Communication Networks**]: General

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.

General Terms

Security, Human Factors

Keywords

Device Pairing, Authentication, Audio Emanations, Signal Processing

1. 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. 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. The 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 off-line trusted third party, a certification authority, a PKI or any pre-configured secrets. However, such a global infrastructure may not be possible in practice, thereby making pairing an interesting and 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 [11].

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 2.1). 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 [25] that eventually translate into man-in-the-middle attacks.

A natural workaround 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 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, 8] can be executed to achieve pairing. Several prior proposals, including [9, 20, 22, 23] (reviewed below), have taken this approach to pairing.

1.1 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 these methods in detail in the Section 2.2.)

- **IMD Pairing:** This method [9] 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 [20] 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) [22, 23] 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. We refer to the variant that uses vibration as Vibrate-Button and the one that uses blinking as Blink-Button.

1.2 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 prior research. 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 (discussed in Section 2.3) concentrated on recordings from a single 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 [9], 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 further 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 set-up, 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 key-press, 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 set-up, i.e., from a farther distance from the vibrating phone.

For BEDA Vibrate-Button, we again consider acoustic emanations associated with a vibrating phone, and for BEDA Blink-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 set-up, i.e., without manual sensing of vibrations as in Vibrate-Button and without observing the blinking as in Blink-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 is the first to explore 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 [3, 27] (we discuss these in Section 2.3). 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 3, we give an overview of our experimental setup and techniques. In sections 4, 5 and 6, we present our audio eavesdropping attacks on IMD Pairing, PIN-Vibra and BEDA, respectively. Finally, in Section 7, we discuss the implications of our attacks on the security of the three schemes.

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

2. BACKGROUND AND PRIOR WORK

2.1 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., [24, 4, 13]). Such d2d channels require both devices to have transmitters and corresponding receivers (e.g., IR 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 [25]). 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 [18]. 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 [11]. (This is referred to as a *fatal* human error [11] 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 [14, 16]. 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 [11]. Even worse, a *rushing user* [19][2] may simply "accept" the pairing, without having to correctly take part in the decision process.

2.2 AS-OOB Pairing Methods

IMD Pairing: Wireless implantable medical devices, such as pacemakers and cardiovascular defibrillators (ICD), have recently been shown [9] 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 [9], 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. Pairing would allow the IMD to establish a shared secret key with the reader on-the-fly and engage in secure communication thereafter.

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. These constraints may also limit the use of IMD Pairing based on distance bounding techniques [15].

The pairing approach proposed in [9] 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 [9] 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 for a cardiac implant), but was difficult to comprehend 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, 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 road-block, however, 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 vice versa) and a somewhat atypical usage model (users often place RFID tags in their wallets and might not be in direct contact with them).

In [20], 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 simply touch her vibrating phone with the tag or object carrying the tag (e.g., a wallet); 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 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. In this paper, we investigate the feasibility of eavesdropping the PIN-Vibra vibrational channel. In particular, we demonstrate how the acoustic emanations associated with a vibrating mobile phone can be eavesdropped upon from a short distance.

BEDA: BEDA [22] 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 [8], and has two variants we study in this work: "Vibrate-Button" and "Blink-Button".[3] 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 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 security of BEDA is clearly based on the secrecy of the pass-

[2]A rushing user is a user who – in a rush to connect her devices – would skip through the pairing process, if possible [19].

[3]The third variant of BEDA belongs to a different class of pairing approaches than the one considered in this paper (i.e, the one where randomness is derived via user inputs), and is out of scope of our current work.

word which is being transmitted via vibration (or blinking) on one device and synchronized button-pressing on the other device. We show, in this paper, that both 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 is susceptible to acoustic eavesdropping of button-pressing.

2.3 Acoustic Emanations

Prior work has considered the problem of eavesdropping over acoustic emanations as a side channel. Asonov and Agrawal [3] 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 (including PINs or passwords, and other secret information, such as credit card numbers) typed by a user. The authors developed signal processing techniques and applied machine learning classifiers to accomplish the task of eavesdropping using an off-the-shelf PC microphone from a distance of up to 1 meter.

Zhuang et al. [27] examined the same problem and improved upon the work of [3]. In particular, they showed that using Mel Frequency Cepstrum Coefficients (MFCC) features [12] yield better classification accuracies compared to the Fast Fourier Transform (FFT) features used in [3]. Moreover, their techniques are based on unsupervised learning classifiers and do not require training data. They further improve the accuracies by incorporating error correction using Hidden Markov Models (HMM) [2].

In a proof-of-concept work published on the web [21], Shamir and Tromer explore inferring of CPU activities (e.g., patterns of CPU operations and memory access) via acoustic emanations. In particular, they investigate how acoustic emanations associated with RSA decryption and signing operations produce unique signatures per RSA private key, and how they can be used to learn the keys.

3. OVERVIEW OF OUR ATTACKS

In the following sections, we demonstrate the feasibility of acoustic eavesdropping on IMD Pairing, PIN-Vibra and BEDA Vibrate-Button and Blink-Button schemes. We implemented (or used existing prototypes) for each of the these methods and recorded the resulting audio signals. We then used signal processing algorithms and 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 facilitates the receiver (honest decoder) to detect the beginning of the key. Adding a beginning sequence is a well-known approach in coding theory 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 and Blink-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

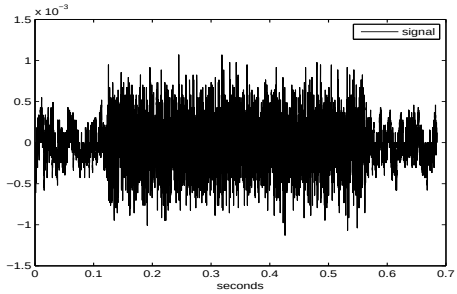


Figure 1: Audio signal for the full key

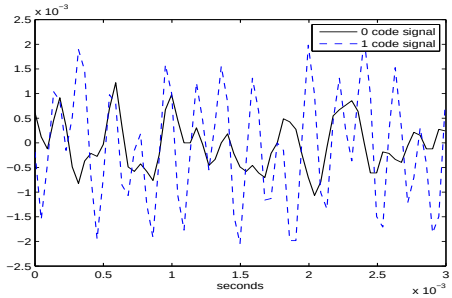


Figure 2: Acoustic signal (in meat)

key press) using signal processing techniques and calculate the key from the time differences between the events.

Our experimental set-up, for the three schemes, consisted of the following common components:

– *PC Microphone*: We used a $20 generic PC microphone (Logitech model number 981-000246) for recordings taken up to 6 ft.

– *Software and System:* We used the Windows sound recorder for recording the sound and the Matlab software for all signal processing and decoding functionalities. The software was run on an IBM Thinkpad X60 running with an Windows XP Professional.

4. EAVESDROPPING IMD PAIRING

4.1 Eavesdropping Challenges and Goals

There are two prior research projects that relate to our work on IMD eavesdropping. The first project [3, 27], (Section 2.3) 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 [26] 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 [9], the main problem in eavesdropping this set-up 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, we do not use an extra 'hail' signal (unlike the codec of [26]) but limit 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. This choice was made deliberately because we found that the lack of a distinct frequency to mark the beginning of the signal makes it harder to detect the exact beginning of the bit, which in turn would make eavesdropping harder.

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 [3, 27]) 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 rate of successful bit decoding.

What also complicates our problem is the fact that the piezo sound amplitude further subdues when inserted in a human body (meat). We Provide sound level measurements in the section 3.

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 off-the shelf recording equipment and signal-processing based algorithms. We further attempt to show that even when using a simple PC microphone and recording a few feet away (outside of a typical PC microphone's optimal recording range), an attacker may still be able to decode the secret key sent.

4.2 Set-Up

In addition to the components described in Section 3, we used PUI Audio piezo model AT-2310-T-LW100-R, resonant frequency 2000 (+- 500) Hz, Voltage 3V, current 3mA. The piezo was connected to a WISP tag [17] (similar to [9]). For distant recording (12 ft), we also used the Educational Insights Sonic Sleuth, model 5200, sold for around $25 at educational toy stores.

We took the following steps for our eavesdropping experiments:

- We encoded a random 144-bit (128-bit key + 8-bit preamble start sequence + 8-bit postamble stop sequence "01111110") binary key with 2-FSK modulation with a baud rate of 341 bps as indicated in [9].
- We inserted the piezo within a combination of beef and bacon to emulate a system inside a human chest exactly as described in [9]. The meat-bacon combination included 1 cm of bacon on top of 4 cm of 85% lean ground beef (overall combination was $19 \times 12 \times 5$ cm). The piezo was attached to the WISP which in turn was attached to the computer.

Sound Level Measurements: We measured the level of the piezo sound from different distances and compared it to the readings reported in [9]. We found that our measurements were comparable. Without meat, the piezo measured 102 dB SPL from close by (5 cm from meat) but degraded to 67 dB SPL when inserted inside the meat (measured just outside the surface of the meat). When measuring from a distance of 1 meter, the SPL measurements were 62 dB for the piezo both inside and outside the meat. This was quieter than the piezo described in [9] which measured 84 dB the piezo buzzing volume just outside the meat surface and 67 dB SPL from 1 meter away. Therefore, although our system is using a standard quiet piezo (quieter than the original one used in [9]), we attempt to demonstrate that we can still eavesdrop upon it.

4.3 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 Figure 1.

We then examined encoded keys recorded from different distances and used those frequency characteristics to detect the beginning sequence.

4.4 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 and produced spectrum-based features.

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 Figure 2) 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.

Since the bits are consecutive, we start by detecting the beginning sequence in the key. We then extract the features from each following 3 ms signal window corresponding to each bit.

Piezo 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 (Figure 10(a) in the Appendix).

We observed that the '0' bit spectrum has two peaks in the 1.67 - 2.68 kHz frequency interval while the '1' bit has a peak at the 2.68-3.35 kHz frequency 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 (Figure 10(b) in the Appendix) 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.

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 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.

To further classify each beep 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 and classify the signal as '1' if it is above that ratio and '0' otherwise. We use this classification when decoding the beginning sequence.

Key Detection and Decoding.

We perform the full key detection in two steps. We first find the key beginning using a specialized procedure that utilizes frequency analysis. Then we decode the key with the help of a machine learn-

ing classifier that uses frequency-based features extracted from the key bits.

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 (associated with the lower frequency interval), we chose the region with the highest frequencies related to this bit (in the 1.67-2.68 kHz interval).

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.

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 mark the first sample in this region as the beginning of the bit.

At this point we continue processing each consecutive constant bit-length signal and classify its value until we locate the preamble sequence ("01111110").

It is expected that the beginning sequence would be the same for all piezo elements (unlike the key, which is random). Therefore, 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.

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 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' bit.

4.5 Classifiers

As discussed in Section 4.4, each recording of 144-bit long keys had 144 rows and 34 columns of FFT features and 13 columns of MFCC features (i.e., for each bit there are 34 FFT features and 13 MFCC features). The resulting feature vectors are then used with two different types of classifiers – supervised [10] and unsupervised [6]. We performed experimental comparison using different classifiers as a tool in order to find which classifiers are able to decode the keys in a robust manner. We also examined the classifier accuracies for different features and distances.

4.5.1 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 used the same key (as mentioned in Section 4.2) for both training and testing, since the classification is only based on the features of each bit tested and is independent of the features of any of the other bits (previous or consecutive bits in the key).

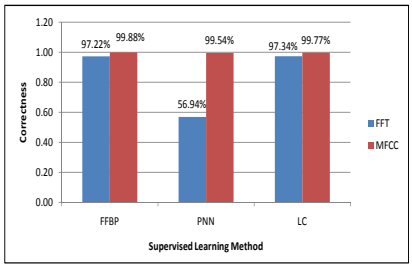


Figure 3: Average correctness of key retrieval for Supervised methods with FFT and MFCC features (3 feet distance)

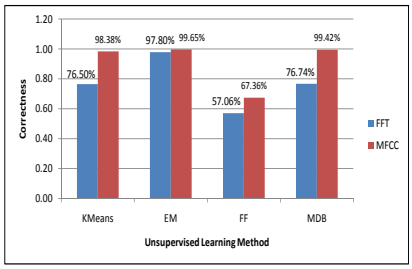


Figure 4: Average correctness of key retrieval for Unsupervised methods with FFT and MFCC features (3 feet distance)

We used well known supervised classifiers: Feed Forward Back-Propagation (FFBP) Neural Network with 20 layers, Probabilistic Neural Network (PNN) and Linear Classifier (LC) [10] implemented in Matlab. The classifier output of each test session was compared with the transmitted key and bitwise comparison was performed to calculate the correctness. We averaged the data from five recordings to calculate the average correctness (%) as follows:

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

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

Figure 3 shows that MFCC features always performed better than FFT features as input to our classifiers, which is also inline with the findings of [27]. Most methods yielded an accuracy of 99-100% for MFCC features. FFBP for MFCC has the highest average accuracy 99.88% (out of five test data set, only one data set returned 1 bit error, others were fully correct), while correctness of LC and PNN are 99.77% and 99.54%, respectively. PNN implementation in Matlab does not work with 34 features (columns) of FFT features, it can not handle that many features for classification but it works with good accuracy (99.54%) for MFCC features having 13 columns. LC has the highest accuracy 97.34% for the FFT features. Overall, we see that LC and FFBP are robust classifiers for IMD eavesdropping.

4.5.2 Unsupervised Classifiers

Unsupervised classifiers can be used in situations where training data is not available or possible to generate. The classifiers divide the test data into different clusters and each cluster is assigned to a label. Since the bits are binary, we only needed two clusters ('0' or '1'). Then, the final key is derived by assigning '0' or '1' to each of the clusters. Therefore, in the IMD eavesdropping setting, the adversary can decode the key using the unsupervised clustering methods without having to rely on previously labeled training data.

We used KMeans, Expectation-Maximization (EM), Farthest First (FF), and Make Density Based (MDB) clustering algorithms implemented in Weka [7]. We used each recording individually to feed

into clustering algorithms. A total of 5 recordings were used to calculate the accuracy of each of the clustering algorithms.

Result of unsupervised learning algorithms are depicted in Figure 4. The graph shows 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).

4.5.3 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 feet), 1-3 feet, 1 meter, and 4-6 feet. Half of the recordings were used for building training data set when using supervised learning algorithms and rest of the recordings were used for testing both supervised and unsupervised algorithms. Methods described in sections 4.5.1 and 4.5.2 are followed for each of the above distances. Graphs in Figure 11 of the Appendix depict the accuracy of IMD eavesdropping from 8 different distances.

From the Figures 11(a) and 11(b), it is clear that for almost all distances, MFCC has higher correctness than FFT. FF has the worst correctness with both FFT and MFCC features. PNN has the lowest correctness as this algorithm can not handle so many features (34 column) in a feature vector. For FFT features, LC and FFBP are quite robust ($>$ 90% correctness) among supervised classifiers up to a distance of 4 feet and EM seems a winner ($\geq$ 80% correctness) among unsupervised classifiers. Overall, supervised classifiers seem to have better performance up to 4 feet distances. Beyond 4 feet, the accuracy for all classifiers degrade significantly.

From Figure 11(b), we find that supervised methods have better correctness than unsupervised methods. However all methods, Except FF, are quite robust up to 4 feet and provide good results ($>$ 90% correctness). The FF classifier provides the worst results since we have only two clusters (the first of the cluster centers is chosen randomly in FF).

However, there is high degradation of correctness between 4-5 feet and sharp degradation between 5-6 feet. So, beyond 5 feet, all methods have poor correctness, which prompts us to consider a parabolic microphone.

4.6 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 and a simple parabolic microphone (without utilizing classifiers). To this end, we took recordings using a parabolic microphone with the same setup (piezo beeping inside meat). We took recordings 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 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.

5. EAVESDROPPING PIN-Vibra

5.1 PIN-Vibra Encoding

In our eavesdropping experiments, we used the original prototype implementation of the PIN-Vibra method [20]. 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 ms = 3.4 seconds.

5.2 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 (we measured it to be 64 dB from close by) 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. Therefore, we found that, similar to the IMD set-up, 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 vs. 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 cancelation 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 (whereby the system regards low amplitude sounds as noise and therefore attempts to discard them).

Our experiments showed that the vibration spectrum indeed became very blurred when we took recordings from a few feet away (vs. 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 cancelation mechanism which causes larger signal blurring when captured from a few feet away.

5.3 Set-up

For our experiments, we used the same components discussed in Section 4.2. Additionally, we used a Nokia mobile phone model E61 the same model used in the experiments reported in [20].

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 [20]).
- The phone was held next to a wallet (the two touched each other) and set to send the PIN. This was done to emulate RFID authentication as described in [20]).

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. The volume was reduced to 64 db SPL at 2 ft away and 62 db SPL from 3 ft away. We also tried to measure the volume of the signal when holding the phone by itself as opposed to touching it with the wallet, but found no noticeable difference in the measurements (which seems to indicate that there was no dampening effect due to touching the phone with 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.

5.4 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. 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. We first detect the beginning sequence using signal-processed based techniques. We then create frequency-based features for each bit and use classifiers to decode each bit in the key from its features. Note that the classification is only based on the features of each bit tested and is independent of the features of previous or consecutive bits (as in the case of IMD).

5.5 Audio Signal Decoding Algorithm

We start by characterizing the phone vibrations as recorded from a close distance (a few cm 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 and the overall bit length is 4410 samples. 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.

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 [20] includes a 'start' sequence equal to "110" as a prefix to the PIN. We start 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.

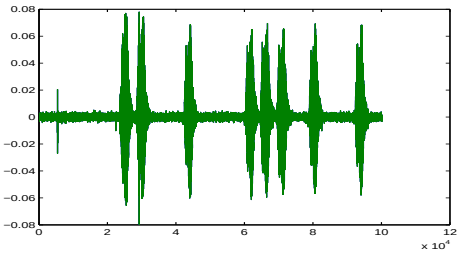
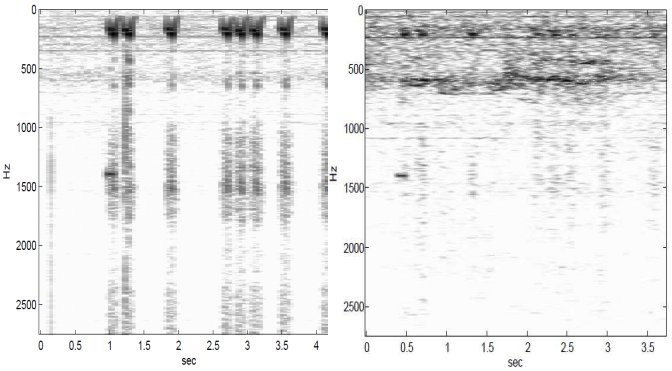


Figure 5: Acoustic signal for full PIN



(a) close by (b) 3 ft distance

Figure 6: Spectrogram

To detect the beginning of the first vibration we used spectrum analysis of the signal. To calculate the spectrum, we used 150 ms window size 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.

After the first vibration was detected, we used energy calculation 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 thi region to mark the beginning of the first bit).

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' or '0'. This is done by calculating the FFT for the 0.2 second window for each consecutive bit. We then create two curves for the sum of the 130-250 hz and 1.1-1.5 frequency coefficients and use set thresholds to determine if each bit is a vibration (marked as a '1' bit) or a sleep period.

Decoding the Signal.

Upon detecting the start sequence, we create both MFCC and FFT feature files for each following bit. We feed these input through machine learning classifiers (we will discuss these in Section 5.6). 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.

Effect of Distance.

When recording from a close distance, we get very clear fre-

quency intervals in the bit spectrum during the phone vibration (Figure 6(a)). Since the PC microphone we used has a noise cancelation mechanism and the vibration amplitude is very low, the signal gets distorted as we get farther from the microphone. When examining the signal from 3 ft away, we see that the frequency response gets blurred and the vibration signal falls over a wide range of frequencies (Figure 6(b)). To counter this effect, we inspected recording signals taken at 3 ft and calculated the sums of the FFT coefficients over the whole 0.4-11 kHz band. We threshold the resulting curve to get the position of the vibration bits.

Further examination showed that since our PC microphone includes noise cancelation algorithm, the vibrations were hard to detect in recordings taken from 4 ft away and farther (as the vibrations were regarded as 'noise' by the microphone). However, when investigating the IMD eavesdropping, we were able to utilize a parabolic microphone to distinguish between recordings of the piezo beep taken from a distance of 12 ft away. Since the phone vibrations last longer than the piezo beep (0.2 second vs. 3 ms), an eavesdropper may listen on the system from a distance over 3 ft using such a microphone with a similar or higher degree of success (relative to the IMD eavesdropping attack). We acknowledge the lack of additional tests to prove this hypothesis.

5.6 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 for 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 4.5).

Result of supervised and unsupervised algorithms for compromising the key by audio eavesdropping on PIN-Vibra method are depicted in the Appendix Figure 12. 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.

6. EAVESDROPPING BEDA

6.1 Encoding and Decoding

In the BEDA scheme [22], 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.

6.2 Challenges and General Approach

We attempt to eavesdrop on both Blink-Button and Vibrate-Button BEDA methods. We note that eavesdropping over button presses in the Blink-Button scheme is somewhat similar to keyboard eavesdropping as discussed in [3, 27]. 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.

While examining the Vibrate-Button method, however, we note that this is more complex due to the fact that the sound of vibration on one device interferes with the sound of button pressing on the other device. Since the vibration lasts longer than the button click (about 500 ms vs. 2-3 ms of the button click), we attempt to detect the overall vibration-button period (which lasts for about 500 ms). We therefore inspect the audio spectrum and find the frequency range of the vibration-button combination. We use this range to detect the starting-time of the audio vibrations and extract the overall secret key.

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. Then, rounding off the resulting time to seconds provides each 3-bit digit within the overall secret. 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.

6.3 Set-up

For our experiments, we used the same components we discussed in Section 4.2. Additionally, we used one Nokia mobile phone model E61 (as also used in the PIN-Vibra setup), and one Nokia N90 Both of these models were used in the experiments reported in [22]. E61 served as the server (the one that vibrates or blinks) and N90 as the client (used for button pressing).

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.

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 and no difference in the SPL measurements were visible on our sound meter (Radio Shack Sound Level Meter model 33-2055). Therefore, we see that the audio sound of the button click is very feeble from a distance of a few feet away and adds to the challenge of eavesdropping on the cell phone button clicks. E61 vibration readings are specified in Section 5.3.

6.4 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 6.2, 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 PIN-Vibra scheme (Section 5.2). 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).

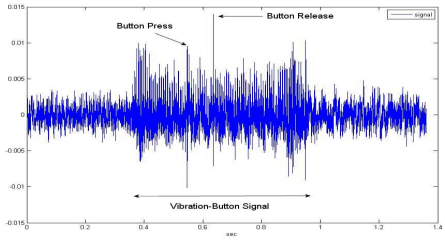


Figure 7: Audio signal (Vibrate-Button)

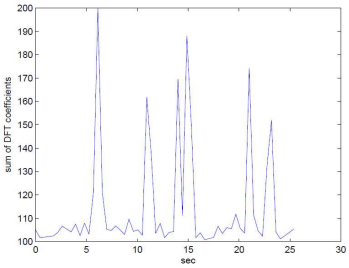


Figure 8: FFT sum (Blink-Button 5-digit key)

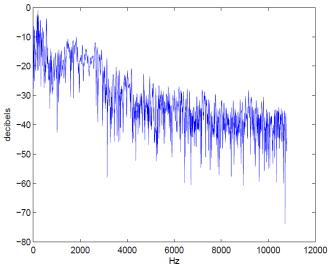


Figure 9: spectrum (Vibrate-Button)

When analyzing the Vibrate-Button audio signal (shown in Figure 7), we note that the vibration lasts around 0.5 seconds while the button click has one main observable peak which lasts only 2-3 seconds 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).

Further inspection of the recording spectrum (Figure 9) revealed that the spectrum is not even throughout the duration of the vibration. This is due to the overlap of the short button pressings with part of the vibration as well as variations in the phone vibrations themselves. Therefore, our experiments showed that dividing the vibration interval into smaller parts can help us more accurately detect regions which are part of the vibration intervals. 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 the 125 ms window size and a step size of 62.5ms. When examining the spectrum, we notice that the vibration spectrum is higher over the range 1 - 7 kHz. To mark potential vibration parts, we calculated the sum of the frequencies between 1-7 kHz. We plot the resulting curve in Figure 8. We use a threshold to determine potential vibration regions. Since each window only lasts 125 ms and our vibrations last around 500 ms, we found that we get more than one point over the threshold within the each vibration period. Therefore, to distinguish between actual phone vibrations and random sounds, we marked a window as a confirmed 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 units of 500 ms.

6.5 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 [1] and performed FFT on the resulting signal. This method is similar to the one described in [27] 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 Figure 8). To verify the button click and eliminate background sounds, we confirm the existence of an actual vibration. The code was then calculated by computing the difference between the verified button clicks (in units of 500 ms).

6.6 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 success rate for detecting the code was 98%.

7. DISCUSSION AND CONCLUSION

Implications of Our Attacks: The attacks we demonstrated on IMD Pairing, PIN-Vibra and the two BEDA variants can be accomplished with a high accuracy by using inexpensive off-the-shelf equipments, 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 and 98% for BEDA variants (for eavesdropping up to 3 ft). We also found that beyond 5 ft, PC microphone may not be very effective for eavesdropping. These distances are inline with prior work on keyboard acoustic eavesdropping [3, 27]. Execution of attacks from these 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)[4]. Moreover, for the IMD set-up, we also explored eavesdropping 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 set-up, the adversary can always verify the correctness of the key that was eavesdropped once equipped with a known plaintext-ciphertext pair, for example. For PIN-Vibra and BEDA,

[4]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).

the adversary can try to use the PIN/password that was eavesdropped to unlock the phone, and launch the man-in-the-middle attack, respectively. Since the PIN/password are correct with a very high probability (as shown by our high accuracy rates), the adversary can compromise the security of these approaches with a high probability, 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 theft); the method still provides strong protection against unauthorized reading and some relay attacks.

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 protocol 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 attack, we also use standard signal processing techniques that can be used to process any button click recordings. 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 some acoustic emanations regardless of the specific model of phone used (similarly, all computer keyboards tested in [3] emitted distinct acoustic emanations). Since we only try to detect the existence of each button click (regardless of what button was pressed), 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.

Acknowledgements

We thank Md. Borhan Uddin for his work related to classifiers, Ersin Uzun for providing us with the BEDA implementation, and Shai Halevi, Jon Voris and CCS'10 anonymous reviewers for their feedback on a previous version of this paper. This work is funded in part by an NSF Grant: CNS-0831397.

8. REFERENCES

[1] Hamming Window Function. Wikipedia, available at `http://en.wikipedia.org/wiki/Window_function`.

[2] A. Moore, School of Computer Science, Carnegie Mellon University. Hidden Markov Model,. http://www.autonlab.org/tutorials/hmm14.pdf.

[3] D. Asonov and R. Agrawal. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy*, 2004.

[4] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network & Distributed System Security Symposium*, 2002.

[5] V. Boyko, P. MacKenzie, and S. Patel. Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In *Advances in Cryptology-Eurocrypt*, pages 156–171. Springer, 2000.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. Unsupervised Learning and Clustering. In *Pattern classification (2nd edition), Ch. 10 , p. 571, Wiley, New York*, 2001.

[7] R. Evans. Clustering for Clasification. In *Master's thesis, Computer Science, University of Waikato*, 2007. `http://adt.waikato.ac.nz/uploads/approved/adt-uow20070730.091151/public/02whole.pdf`.

[8] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA CryptoBytes*, 7(1):29 – 37, Spring 2004.

[9] 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 *IEEE Symposium on Security and Privacy*, 2008.

[10] S. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. In *Informatica Journal*, 2007.

[11] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. Caveat emptor: A comparative study of secure device pairing methods. In *International Conference on Pervasive Computing and Communications (PerCom)*, 2009.

[12] L. Rabiner and B.H. juang. Mel-Frequency Cepstrum Coefficients. Prentice-Hall Signal Processing Series, 1993, ISBN:0-13-015157-2.

[13] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, 2005.

[14] R. Prasad and N. Saxena. Efficient device pairing using "human-comparable" synchronized audiovisual patterns. In *Applied Cryptography and Network Security*, 2008.

[15] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun. Proximity-based access control for implantable medical devices. In *ACM Conference on Computer and Communications Security*, pages 410–419, 2009.

[16] V. Roth, W. Polak, E. Rieffel, and T. Turner. Simple and effective defenses against evil twin access points. In *ACM Conference on Wireless Network Security (WiSec)*, 2008.

[17] A. Sample, D. Yeager, P. Powledge, and J. Smith. Design of a passively-powered, programmable sensing platform for uhf rfid systems. In *IEEE International Conference on RFID*, 2007.

[18] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *IEEE Symposium on Security & Privacy*, 2006.

[19] N. Saxena and M. B. Uddin. Secure pairing of "interface-constrained" devices resistant against rushing user behavior. In *Applied Cryptography and Network Security*, 2009.

[20] N. Saxena, M. B. Uddin, and J. Voris. Treat 'em like other devices: user authentication of multiple personal rfid tags. In

Symposium on Usable Privacy and Security (Poster Session), 2009.

[21] A. Shamir and E. Tromer. Acoustic cryptanalysis on nosy people and noisy machines. `http://people.csail.mit.edu/tromer/acoustic/`.

[22] C. Soriente, G. Tsudik, and E. Uzun. BEDA: Button-Enabled Device Association. In *International Workshop on Security for Spontaneous Interaction (IWSSI)*, 2007.

[23] C. Soriente, G. Tsudik, and E. Uzun. Secure pairing of interface constrained device. *International Journal on Security and Networks (IJSN)*, 4(1), 2009.

[24] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols Workshop*, 1999.

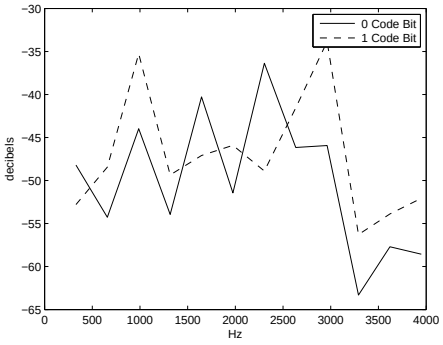
[25] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Usable Security (USEC)*, 2007.

[26] C. V.Lopes and P. Aguiar. Acoustic modems for ubiquitous computing. *IEEE Pervasive Computing, Mobile and Ubiquitous Systems*, 2(3):62–71, July-September 2003.

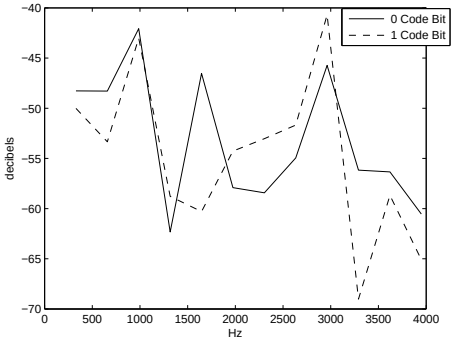
[27] L. Zhuang, F. Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. In *ACM conference on Computer and communications security*, 2005.

APPENDIX

A. ADDITIONAL FIGURES

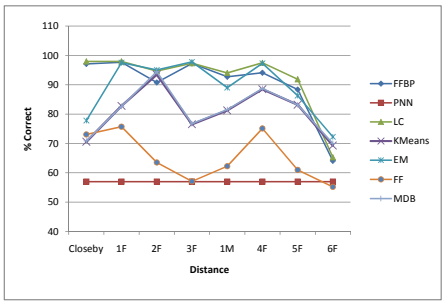


(a) open environment

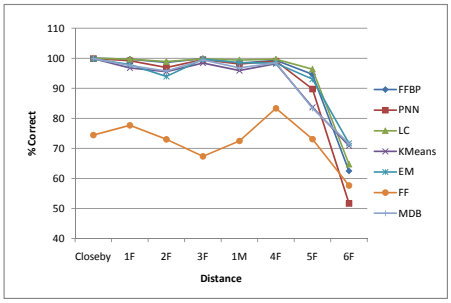


(b) piezo inside meat

Figure 10: Spectrum



(a) Using FFT Features



(b) Using MFCC Features

Figure 11: Result/comparison of Supervised and Unsupervised methods for different distances with different learning algorithms

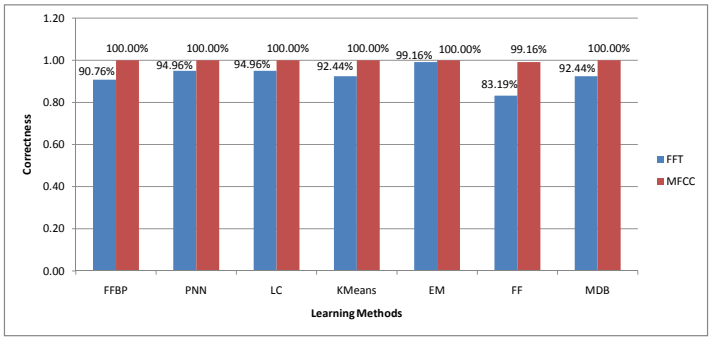


Figure 12: Result of PIN decoding using different learning methods