This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

Noisy Vibrational Pairing of IoT Devices

S Abhishek Anand and Nitesh Saxena

Abstract—Internet of Things (IoT) is embodied by smart network-enabled devices that utilize computing power, networking, and miniaturization for richer and improved user experience. Due to their interconnectedness, ubiquitous nature and low computational power, trustworthy and secure communication between IoT devices is a security concern. For device authentication, "pairing" may be secured by using an auxiliary channel such as audio, visual and vibrations for sharing the key or keying material between the IoT devices. In this paper, we evaluate the security of vibration channel, susceptible to an acoustic eavesdropper that can capture audio leakage from the vibrations of the transmitting IoT device. We propose a noisy vibration scheme for cloaking vibration sounds during pairing against such attacks. The scheme only requires a speaker for emitting the masking sound during key transmission. We evaluate the scheme in proximity, co-located and remote settings with an eavesdropping attacker. We also study motion sensor exploits against this scheme and compliment it with additional measures to mask vibration effects on motion sensors. Our scheme is user transparent and requires only a speaker (may already be present on the device), so it can be readily implemented in the IoT setting, smart wearables, and other commodity gadgets.

Index Terms—pairing, side channel, IoT, signal masking, VoIP.

1 Introduction

Internet of Things as defined by Global Standards Initiative on Internet of Things is "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [3]. Examples of household IoT devices include the Nest smoke detector and thermostats, Smart thermometers from Kinsa, smart bluetooth or wifi bulbs for indoor lightening, smart security sensors including motion sensors etc. [1] A forecast by Gartner, Inc. states that 8.4 billion connected things will be in use worldwide in 2017 and this number will rise up towards 20.4 billion by 2020 [4]. In addition, consumer based IoT devices constitute 63% of total IoT applications in 2017. Such dynamic growth points towards the significance of these class of devices in the future end-user computing infrastructure and services.

In an IoT network, the devices usually communicate with a central hub, usually a smartphone, or among each other through WiFi, Bluetooth, RFID or long range cloud-based interactions. These communication channels are inherently insecure i.e. they can be easily manipulated or eavesdropped upon and therefore present a fundamental challenge of securing such transmission pathways. "Pairing" is commonly referred to the operation of bootstrapping secure communication between two such devices such that the communication between them is resistant to eavesdropping and manipulation (man-in-the-middle) attacks. Pairing of IoT devices is a hard challenge due to lack of a global infrastructure that can enable devices to share an on- or off-line trusted third party, a certification authority, a public key infrastructure (PKI) or any pre-configured secrets.

An existing approach to pairing is to leverage an auxiliary channel, also referred to as an out-of-bound (OOB) channel, that can be controlled by the users operating the devices. OOB channels, unlike wifi or bluetooth, are *human-perceptible* that indicates perception by one or more of the human senses. Some examples of such OOB channels are audio, vibration and visual medium. In these channels, the user is able to confirm the origin of the transmitted message and can detect any manipulation attempts by an adversary. It may not be able to prevent the adversary from eavesdropping on the OOB channel. Pairing using OOB channels has been referred to as authenticated-OOB (A-OOB) [5]. It has been the basis of a number of protocols proposed in literature as surveyed in [6].

Pairing becomes a challenging problem when one of the devices involved in the process falls under the category of *constrained devices*. A device is considered to be a *constrained device* when it lacks a good quality output interface (such as a full screen display) or an input interface (e.g. a keyboard) or receivers (e.g. cameras, microphone). Many IoT devices (like smart home appliances) fall into this category. A-OOB pairing for such devices is difficult due to the fact that establishing bidirectional and automated A-OOB channels on such devices is challenging, in general. In addition, manual mechanisms for pairing constrained IoT devices can be prone to human errors [6] that could potentially lead to man-in-the-middle attack.

An alternative approach for pairing geared especially towards constrained devices involves using a secret as well as authenticated OOB channel (AS-OOB in [5]). Since the channel is supposed to be authenticated as well as secret, it is assumed that an adversary would be unable to detect or manipulate the transmission over this channel between the two devices. In an AS-OOB channel, pairing can be achieved by simply transmitting the key or the keying material over the channel and it is also devoid of any human errors. In case of a low bandwidth channel, a short PIN or password can be transmitted and password-based authenticated key agreement [7] protocol can be used for pairing.

Some examples of pairing protocols that use AS-OOB channel are [8] and [9]. The IMD pairing scheme in [8] uses

This submission is an extension of work done in [1] and [2].

1. A list of IoT devices currently on market is available at *http://iotlist.co*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

a low frequency audio channel to pair an RFID tag (attached to an implanted medical device) with an authorized RFID tag reader. The PIN-Vibra method [9] uses vibration channel to transmit the key/keying material from the transmitter (e.g., the IoT hub smartphone device) to the receiving device (e.g., an IoT appliance). In this scheme, the phone generates a short PIN, encodes it as an ON-OFF vibration scheme and transmits it as vibrations (generated by a miniature vibration motor) to the receiving device. The receiving device can read the vibrations using an MEMS accelerometer followed by a decoding process based on the ON-OFF scheme.

While both the pairing schemes detailed above offer an alternative approach to pairing, they have been shown to be susceptible to eavesdropping attack. In the case of using a low frequency audio channel for transmitting PIN or short password, it was discovered that the audio channel can be eavesdropped upon by an adversary that can learn the transmitted communication [5]. In this work, we show that similar eavesdropping attack can also be detrimental to a vibration based AS-OOB channel pairing protocol that can lead to leakage of the transmitted PIN or short password. Furthermore, we also demonstrate that it is still possible to eavesdrop on the vibrations during the pairing process even when the vibration sounds are cloaked within a band limited white noise based masking signal. We provide a viable countermeasure leading to a secure "noisy" vibrational pairing scheme well-suited for securely connecting the IoT hub smartphone device with other IoT devices.

1.1 Our Contributions

In our work, we perform an investigation of pairing protocols for IoT scenarios based on vibration channel.

- We recreate Pin-Vibra [9] protocol to implement pairing between an IoT hub (a smartphone) and an IoT device. We show that this protocol is vulnerable against an eavesdropping adversary that exploits the acoustic leakage resulting from vibrations ("standard attack model–*proximity attacker*"). To counteract it, we introduce noisy vibration based pairing where the acoustic leakage from vibrations are hidden by a masking signal (noise signal).
- We further show that while noisy vibration based pairing may be able to withstand "standard attack model", an "advanced attack model–*colocated adversary*" can relatively easily defeat noisy vibration pairing. To prevent such an attack, we suggest the addition of low frequency tones to the masking signal that would hide the acoustic leakage at low frequencies preventing the co-located adversary from learning any information about the vibrations.
- We enhance the proposed defense against a co-located motion sensor exploiting adversary by injecting fake readings into the actual sensor readings. We extend the notion used in [10], [11] in our proposed defense by using it to mask the vibration effect on accelerometer on the transmitting device thereby mitigating an adversary that may exploit the accelerometer on the transmitting device.
- We propose a novel eavesdropping scenario where a standard voice call or voice over internet protocol (VoIP) applications are exploited to spy upon the acoustic leakage from vibrations remotely. We show that it is possible for such an attack to decode the transferred keying information (during pairing) by exploiting the vibrations sounds, recorded over the call, in a similar manner to previous acoustic eavesdropping exploits against vibration pairing.

1.2 Outline of the paper

In Section 2, we detail the existing work on pairing constrained devices using an A-OOB channel. Section 3 demonstrates the protocol used in vibration based pairing mechanism in the context of constrained devices. Section 4 showcases the vulnerability of vibration based pairing protocol against a proximity attacker and Section 5 proposes noisy vibration pairing as a possible solution. Section 6 analyzes white noise defense against a proximity attack and Section 7 analyzes it against a co-located adversary. The remote eavesdropping attack is introduced in Section 8 and contains evalaution of the attack against noisy defense mechanism. Section 9 summarizes and discusses the results and Section 10 has the take-home message.

2 Related Work: Pairing for Constrained Devices using OOB Channel

The usage of OOB channel for pairing and bootstrapping security has been proposed in existing literature. Balfanz et al. [12] proposed the use of a location limited channel (based on physical contact between the devices) in a pre-authentication step where the pairing devices can exchange pairing information. This data can later be used for subsequent authentication of the devices on wireless or Bluetooth channels. The location limited channel could be audio, infrared, visual or contact channel. Goodrich et al. [13] developed an audio based OOB channel for secure device association called "Loud and Clear". It used a text-to-speech (TTS) engine for converting an English sentence (derived from device's public key) to speech and display/render the same sentence on the receiving device.

Another audio based pairing approach was taken by Halperin et al. [8] for wireless implantable medical devices (IMD), such as pacemakers and implantable cardiac defibrillator (ICD). They showed that prior communication protocols used by IMDs to communicate wirelessly with an external programming entity are susceptible to various radio-based attacks. They proposed zero-power defenses involving an RFID tag attached to the IMD. A secure communication channel is established between the IMD and an external reader by having a small piezo element attached to the RFID tag on IMD. The piezo element transmits a random key over a low frequency audio channel than can be recorded and decoded by an external reader.

Seeing-is-Believing is a pairing protocol proposed by McCune et al. [14] that is based on a visual channel. In this protocol, a camera is used to take a snapshot of a barcode that encodes the cryptographic material. The barcodes can be pre-configured and attached as labels on devices or they can be produced on demand and displayed on the device's display unit. Saxena et al. [15] extended this work to the context of constrained devices especially devices with limited form of display, such as a single light emitting diode (LED). Since most of the constrained devices may already have a single light sensor and the visual channel for the light sensor can be verified by a human user, this scheme is both cost effective and secured by user perception.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

Kim et al. [16] proposed a vibration based side channel for securing communication between an external device (a reader or a smartphone) and a medical device. The proposed technique is based on an OFF-ON keying demodulation scheme for exchanging a shared cryptographic key for implantable and wearable medical device (IWMD). They also performed a security analysis of the proposed scheme against an acoustic eavesdropping adversary at a distance of 30 meters. They concluded that such an adversary was unable to demodulate the recorded waveform into the shared key in the presence of a strong masking signal.

Several other pairing protocols based on bidirectional device-to-device A-OOB (e.g. [6]) require both devices to have transmitters and corresponding receivers (e.g., IR transceivers), which may not exist on constrained devices. In settings, where dtd 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 (dth) and human-to-device (htd) channel(s) instead (e.g., based on transfer of numbers [17]). However, establishing such channels on constrained devices may also not be feasible.

3 Vibration based Pairing for IoT Devices

Pairing between devices often involves a shared secret that allows the communicating devices to authenticate themselves and establish a trusted and secure communication channel. For example, pairing between an IoT device and an IoT hub could be achieved by the IoT hub generating a short key or keying material and sending it to the receiveing IoT device over an OOB channel. The receiving IoT device can decode the key/keying material depending upon the nature of the OOB channel and utilize the received data for further securing the main communication channel. The security of producing a shared secret as such, rests upon the premise that the associated OOB channel is inherently secure against eavesdropping and manipulating attacks.

***Vibration based Pairing Protocol*:** In a vibration based pairing scheme, the transmitting IoT device encodes the keying material (a short passphrase or PIN) into vibrations through a vibration motor. The transmitting device is placed in contact with the receiving device prior to the pairing process. The receiving device (also a smart device) contains MEMS accelerometer that is capable of recording the vibrations conducted through the transmitting device. The accelerometer readings after recording the vibrations can be decoded to obtain the transmitted data.

A common encoding technique for encoding bit string into vibrations that can be transmitted to another device is a time based ON-OFF encoding mechanism. Under this scheme, each bit is encoded as a vibration for a fixed time interval (t) if it is "1" and every "0" bit is encoded as same time period (t) of stillness or no vibration. In Vibrate-to-Unlock scheme [9], the time interval t is fixed as $200ms$. In order to detect a valid transmission, a header is attached to the transmitted data which is fixed as "110". As per the scheme, a 4-bit PIN is generated by the transmitting device and is converted to its 14 bit binary equivalent string. Prior to transmission, the preamble (as described before) is attached to the string increasing its length to 17 bits. Since time duration for every bit is $200ms$, the entire transmission takes $17 \times 200ms = 3.4$ seconds.

4 Proximity Attack on Vibration Pairing

The pairing scheme described in Section 3 has been shown susceptible to an acoustic eavesdropping attack in [18]. In this attack, the attacker exploits acoustic emanations that are generated during the process of pairing as a result of the vibrations of the transmitting device. In this attack model, we assume that the pairing protocol being used is known to the attacker. This knowledge includes the bit length of the transmitted string, the scheme used for encoding vibrations and the preamble attached to the bit string that denotes the beginning of the transmission.

The proximity attack model assumes an attacker eavesdropping on vibration based pairing from a nearby location. The distance between the attacker and the pairing devices depends on the loudness of vibrations generated by the transmitting device and on the ability of the eavesdropping attacker to record those acoustic emanations. The attacker does not have access to the microphone (if present) on any of the devices involved in the pairing effort. The attacker or the listening device, being at a distance, can be assumed to unnoticeable by the victim/s.

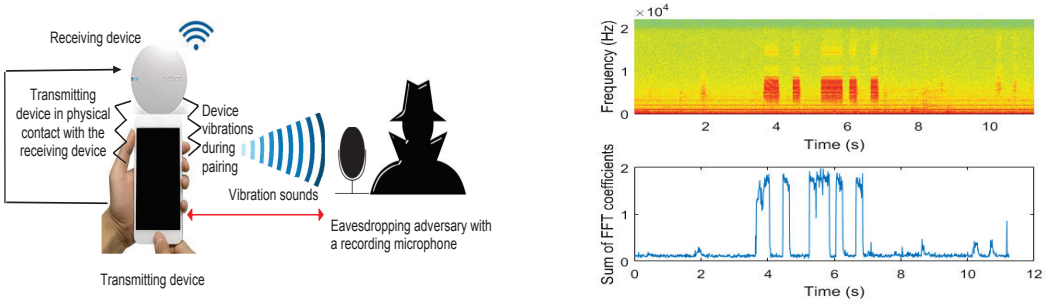
In order to highlight the threat of the attack, the attacker uses off the shelf recording devices such as PC microphones or microphone/s in his own smartphone. These devices are low cost and ubiquitous in nature and hence make the attack simple to launch. A depiction of proximity attack model can be found in Fig.1a. We also assume that the environment is devoid of any intentional background noise that may interfere with the recording capability of the attacker.

4.1 Attack Experiment under Proximity Attack Model

Equipment: We used Motorola Droid X2 phones as pairing devices where one of the phones acted as a transmitter while the other phone acted as a receiver. Both phones were equipped with vibration motor and accelerometer sensor. To record the acoustic leakage from vibrations, we used Dynex USBMIC13 PC microphone [19] with a frequency response of 150Hz – 10kHz and Audacity application. To process the captured audio, we used Matlab signal processing toolbox.
Experiment: On examining the spectrum in Fig.1b, the acoustic leakage from the vibrations shows a dominant response in the frequency band 3.5 kHz to 8.3 kHz. The intensity of the signal in the spectrum (an indication of energy in the signal) seems to concentrated from 6.8 kHz to 7.8kHz. To decode the transmitted data from the recorded signal, we convert the recorded signal to frequency domain and detect the beginning of the transmission (using the preamble "110") using a suitable threshold for the sum of FFT coefficients of the signal. A window size of 441 samples with 50% overlap and the frequency band 6.8 kHz – 7.8 kHz is chosen for best results. The plot of sum of FFT coefficients against time can be observed in Fig.1b.

We tested our eavesdropping attack on ten random PINs using the above setup with the recordings done at a distance of 15cm. The attack was successful with 100% accuracy, demonstrating that communication using vibrations is susceptible to an acoustic eavesdropping attacker with a high

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Acoustic eavesdropping: proximity attack model

(b) Spectrogram of the acoustic leakage (Frequency vs Time)

Fig. 1. Proximity attack model and acoustic characteristics of the vibration leakage for PIN "4562" under the model. Color intensity in the spectrum is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

degree of accuracy. This result is in-line with the results of the acoustic eavesdropping attack scheme proposed by Halevi et al. [5].

5 Our Defense: Noisy Vibration Pairing

A proximity eavesdropping attacker exploits acoustic leakage emanating from the vibrations during pairing for decoding the transmitted PIN or key. To prevent such an attack, acoustic leakage should be minimized or hidden in such manner that it becomes very hard for the proximity attacker to extract any meaningful information about the transmitted PIN or the key. To achieve this objective, either signal cancellation or signal masking can be deployed.

5.1 Audio Leakage Cancellation

Roy et al. [20] proposed canceling of the sounds of vibration (termed SoV) by generating an "anti-noise" signal on the transmitting device (source of vibration). To estimate the effect of surface on which the device has been placed, a short preamble is transmitted and the resulting SoV is recorded. The FFT of SoV is examined for the strongest overtones that are then combined to create the "anti-noise" signal. For phase alignment, the transmitter increases the sampling frequency of "anti-noise" signal keeping track of the phase difference of the "anti-noise" and SoV, switching it back to its original value when the phase difference is minimum.

This approach may not be suitable for IoT devices that are computationally restricted in their ability to perform heavy signal processing tasks like FFT calculation and real time phase synchronization. It may also take more time to generate "anti-noise" signal than the entirety of the pairing process (our setup in Section 3 takes 3.4s). Another possible flaw in signal cancellation defense lies in the fact that it is not possible to cancel out SoV completely and promptly. It takes some time before we can determine the phase difference and then perform the matching, during which SoV would be constantly leaking confidential information. It may also be possible that a more sophisticated attacker (e.g. using machine learning) may be able to recover some remnant information about SoV from the resultant signal.

If we do not consider the duration of the communication as a limiting factor by artificially padding it via an addition of a preamble to the actual PIN, cancellation of audio signal may yet prove to be capable of mitigating the acoustic side channel attack. However, in this work, we restrict ourselves to the examination of easy to generate and computationally light signal masking technique.

5.2 Audio Leakage Masking

Signal masking mechanism is commonly featured in radio communications where the presence of noise in the environment corrupts the signal. If the signal to noise ratio (SNR) is low, it becomes hard to differentiate the signal from the background noise. This phenomenon can be utilized against an acoustic eavesdropping attacker by intentionally introducing noise (referred as masking signal) during the vibrational pairing so that it cloaks the audio leakage from the vibration making it indistinguishable from the masking signal. As the effectiveness of the defense mechanism depends on the difficulty of the adversary's task in filtering out masking signal from eavesdropped signal, we test out variants of masking signal that can be deployed to mask the acoustic leakage from vibrations effectively.

To defend the pairing process against a proximity attacker (Section 4), we propose Vibreaker , a defense mechanism that generates a masking signal obscuring the audio leakage, making it hard for the adversary to extract any information about the data transmitted during the pairing of devices. In order to measure the effectiveness of our mechanism, we experiment with different types of sounds that could potentially be the masking signal and evaluate their security against an attacker as defined in Section 4.

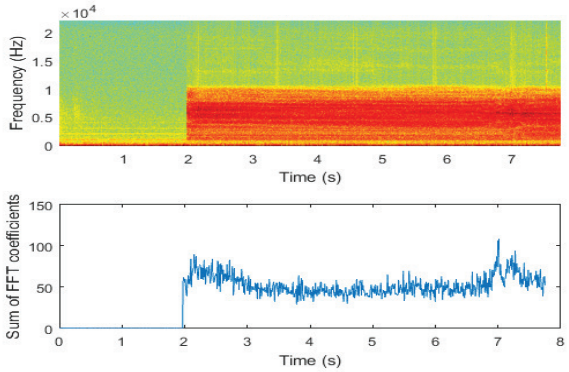
6 Vibreaker against Proximity Attack

We will now evaluate the efficiency of various masking signals against an eavesdropping attacker as described in Section 4. We will also investigate the possibility of filtering out the masking signals by the attacker and extracting any relevant information from the resultant signal.

6.1 White Noise as Masking Signal

White noise is defined as a random signal having a constant power spectral density. White noise is constantly present in our environment for example, the humming sound emanating from air conditioning units. It has also been used for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Frequency spectrum and sum of FFT coefficients plot against time for the eavesdropped signal.

(b) Bit error rate vs threshold values for noisy vibrational pairing. Since we begin from maximum threshold value, the x axis is in descending order.

Fig. 2. Frequency features of the eavesdropped signal and accuracy of a proximity attacker against Vibreaker(white noise) for PIN "4562". Color intensity in the spectrum is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

sound masking in offices by suppressing other distracting sounds. Here, we use white noise as the simplest candidate masking signal that can be generated easily. Filtering the white noise signal is not a hard task, but the process of filtering also diminishes the quality of the recovered signal. Since white noise is evenly distributed over all of the frequency spectrum, trying to filter it out also removes some of the signal in the frequency bands where the white noise overlaps with the spectrum of the original signal (audio leakage from the vibrations).

Experimental Setup: We use the *wgn* function of Matlab to generate a 10 second sample of white Gaussian noise at a sampling frequency of 44.1 kHz. White gaussian noise is a good approximation of real world white noise and hence sufficient for our intentions. A frequency filter to the generated noise sample can be applied to limit the white noise spectrum to the same frequency band as the audio leakage from vibrations. Once we have generated the white noise sample, we play it in the background during the pairing of the IoT devices with the transmitting device vibrating to deliver the PIN or key to the receiving device. To make sure that the white noise suppresses all the audio leakage, we introduce a delay in the pairing process such that it begins only after the white noise has started playing in the background.

Observations: To study the effectiveness of white noise as a masking signal against a proximity attacker, we use the pairing protocol described in Section 3. Our observations for the recording done at a distance of 15cm (Fig.2a) show that white noise is able to mask the audio leakage from the vibrations. In addition, plotting sum of FFT coefficients over time (an indication of energy in the signal) does not reveal any vibration sounds in the intended frequency domain. The bit error rate of the attacker (Fig.2b) never reaches 0%, the best effort being at approximately 35%. Apart from covering the spectrum in which the audio leakage from the vibrations lie, the sound level of the white noise can be kept more than that of vibration sounds thereby easily suppressing the leakage. Since vibration sounds are not loud, it is easy to generate louder white noise for a short duration that only lasts till pairing is successfully completed.

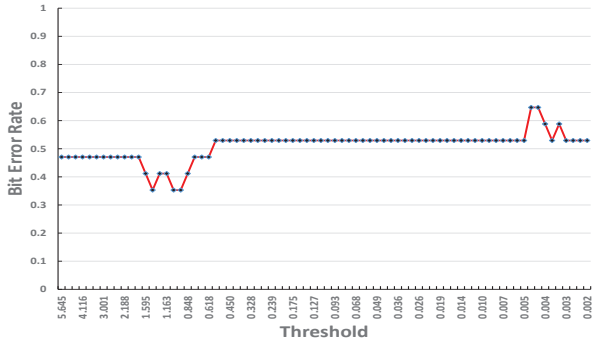


Fig. 3. Bit error rate against threshold values for PIN "4562" after filtering the white noise. Since we begin from maximum threshold value, the x axis is in descending order.

Filtering the Masking Signal: While our results indicate that white noise is capable of hiding acoustic leakage, there exist filtering method that use spectrum subtraction to remove a static signal like white noise from the recorded audio. For filtering the white noise, we use the *noise reduction* effect from Audacity tool that selects a small sample of noise as the profile of the static signal to be removed and subtracts it from the spectrum of the recorded audio. This technique is known as spectral noise gating and it works by suppressing all pure tones that fall below a pre-determined threshold (determined from the created profile of static noise) in each frequency band. We used a noise reduction level of 15dB and a sensitivity value of 6 to get the best results. Fig.3 shows the bit error rate for the attacker after filtering the white noise. From the plot, it is clear that the attacker never achieves a bit error rate of 0% (fully decoded PIN). The best bit error rate remains similar to the one achieved prior to filtering. This effect is due to complete cloaking of the audio leakage by the white noise and the subsequent filtering that destroys any information about the acoustic leakage itself (due to frequency overlap).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

6.2 Pre-recorded Vibration Sounds as Masking Signal

Our next choice of masking signal is a close representation of the audio leakage itself i.e. the sound generated during the vibration of the PIN transmitting IoT device. We pre-record the sound emanated during the vibration and try to confuse the attacker by masking the audio leakage from the vibrations with pre-recorded vibration sounds (henceforth referred as fake vibrations).

Experimental Setup: We generated a random sequence of numbers and encoded them as vibrations using the same protocol as PIN-Vibra [9]. However, in order to make sure that the fake vibrations completely overlap with the actual vibrations, we reduced the duration of silence between the vibrations from 200ms to 100ms. The resulting vibration sequence is recorded offline and stored for use as the masking signal. When the user initiates the protocol for sending the PIN via vibrations, the device in addition to vibrating also plays the stored masking signal in the background. We adjusted the timings of the masking signal such that it always begins playing at approximately the same time as the vibrations. The proximity attacker is again presumed to be eavesdropping at a distance of 15cm.

Observations: Our results (Fig.4) show that fake vibrations are able to mask the audio leakage resulting from the device's vibration. It is very hard to distinguish between fake vibration signals and the audio leakage based only on frequencies as demonstrated in Fig.4a. The sum of FFT coefficients also shows identical response from the fake vibrations and the acoustic leakage from vibrations indicating that audio leakage has completely been masked. The bit error rate for the proximity attacker for each threshold value is shown in Fig.4b. The best bit error rate achieved by the attacker is 30% which is similar to the accuracy achieved in the proximity attack model in noisy vibration pairing. Thus fake vibrations provide similar defense capabilities as white noise in our experiments.

Filtering the Masking Signal: We applied the same filtering process that was used for filtering out white noise. Since sounds of fake vibration differ from actual vibration sound due to imperfect reproduction by the speakers, we (as an attacker) listened to the eavesdropped audio signal and selected the part that we believed to be the fake vibrations. The selected part of the audio was used as the noise profile and applied to the full length of the eavesdropped audio signal for filtering the fake vibrations. The results (Fig.5) after the filtering process show that the bit error rate is around 20% indicating that the attacker fails to completely decode the recorded sound even though it is a slightly better error rate when compared against white noise. Thus fake vibrations could also serve as a masking signal for obfuscating the vibration sounds.

7 VIBREAKER AGAINST CO-LOCATED ATTACK

In the attack model described in Section 4 (*proximity attack*), the eavesdropping attacker is at a distance from the pairing devices. This threat model can be further strengthened by decrementing the distance between the pairing devices and the eavesdropping attacker to almost zero. This extension of the attack model places the adversary in the same physical location as either of the devices involved in pairing, henceforth called *co-located attack*.

The advantage of a co-located eavesdropper from an attack point of view is a more accurate recording of the acoustic leakage from vibrations as it places the recording device close to the source of acoustic emanation from the vibration motor of the device transmitting the pairing data. The assumption that the eavesdropper is residing on one of the devices (*co-resident*) can be realized in real life if the vibration transmitting device is equipped with an on-board microphone (for example, a smartphone) that can be manipulated into recording vibrations through a malicious application installed by an attacker. Another way for an attacker to implement *co-located* attack scenario would be by attaching a tiny listening bug [21] to either of the pairing devices. This effort would, however require one-time access to the compromised device constituting a *lunch-time* attack.

A *co-located* attacker can exploit the microphone on the device to record the acoustic leakage resulting from vibrations. One point to note in the context of smartphones is that access to microphone requires explicit permission from the user. However, studies done on android smartphones suggest that the comprehension and attention level of users while granting these app permissions are very dismal [22]. This may help an eavesdropping attacker slip past such restrictions. In addition, audio channel opens up other different attack possibilities that include eavesdropping via a voice based call constituting a case of remote attacker–coresident eavesdropping scenario. Towing the same line as *proximity attack* model, we assume that attacker is capable of recording the acoustic leakage from vibrations and process it offline using signal processing tools. The environment is considered to be noise-free except for the participating devices in the scenario.

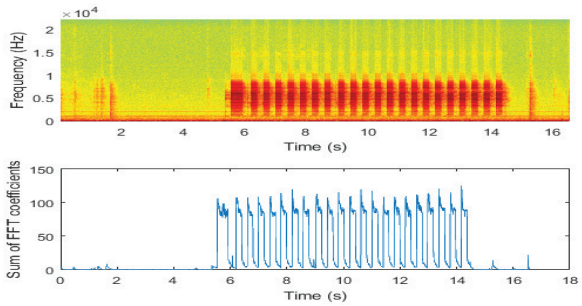
In addition to a co-located attacker that can exploit the microphone of the transmitting device (IoT hub), another type of co-located attack exists that can exploit on-board motion sensors of the IoT hub (smartphone) and learn the transmitted PIN/passphrase from the effect of IOT hub's vibrations on its own motion sensors. We study this type of attack and propose a defense that seeks to transparently hide the vibration's effect present in the motion sensor's readings from the co-located attacker.

7.1 Attack Experiment under Co-located Attack Model

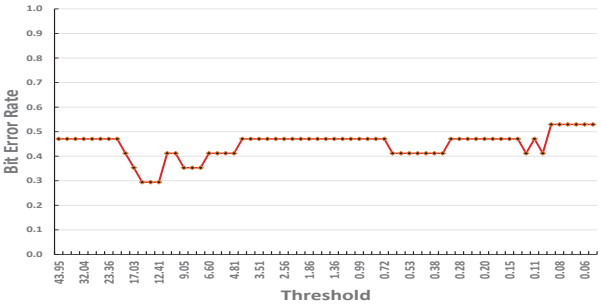
The attack principles under this advanced eavesdropping attack model are similar to *proximity attack* model. We implemented the same ON-OFF encoding scheme for transforming 4-bit PIN into vibration sequence. We used the same set of Motorola Droid X2 smartphones as the communicating devices, with one acting as the transmitter and other as the receiver. To record the audio generated from the vibrations, we used Dynex PC microphone and Matlab's signal processing toolbox for processing the recorded audio. The microphone is placed at a distance ($\leq 1cm$) from the vibrating device in order to record the vibration sounds at the closest possible distance for emulating a co-located adversary. The on-board microphone can also be used for this purpose as per detailed in the threat model.

To complete the noisy vibrational pairing setup, we also implemented the defense measures as proposed in [23].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Signal Features in presence of fake vibrations.



(b) Bit error rate against threshold values in the presence of fake vibrations. Since we begin from maximum threshold value, the x axis is in descending order.

Fig. 4. Frequency features of the eavesdropped signal and the effectiveness of a proximity attacker in presence of fake vibrations for PIN "4562". Color intensity in the spectrum is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

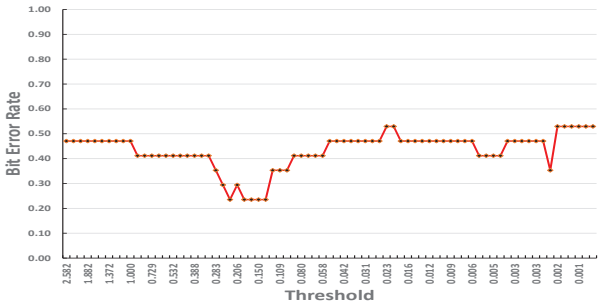


Fig. 5. Bit error rate against threshold values for PIN "4562" after filtering fake vibrations. Since we begin from maximum threshold value, the x axis is in descending order.

The defense measure in [23] utilized band-limited Gaussian white noise that lies in the same frequency range as the audio leakage from the vibration, to hide the acoustic leakage from the vibrations. The masking sound was generated as the transmitting device starts the communication with the receiving device for pairing. On the side of the attacker that has the capability to process the eavesdropped audio signal offline, we also used the "noise reduction" feature of the audio processing tool Audacity to filter out the noise and reveal part or whole of the audio leakage. This feature allows the selection of a small portion of the audio signal consisting of the noise only to build a noise profile that is then filtered from the whole audio signal.

7.1.1 Effectiveness of the Attack

As per the threat model detailed in Section 7.1, we recorded vibration sounds superimposed by the masking sound at a distance of 0cm. We also recorded vibration sounds at a distance of 10cm for comparing it with the co-located adversary scenario. Figure 6a and Figure 6b represent the frequency spectrum of the eavesdropped signal at distances 0cm (as per our threat model) and 10cm (similar to [23]). The frequency spectrum revealed that masking sound may be able to hide the audio leakage due to the vibrations from an adversary eavesdropping at a distance. However, for a co-located adversary, the masking sound was unable to hide the audio leakage resulting from the vibrations at the lower frequency range of 50Hz-250Hz. Since white noise has same intensity for all frequency, while the vibration emanations have higher intensity at low frequencies (as evident in frequency spectrum of Figure 6a, we believe white noise alone would be unable to hide the emanations due to uneven spectral power density of the emanations.

In our proximity attack model (Section 4), we showed that white noise was enough to hide acoustic leakage occurring from vibrations of the device for an attacker at a distance of 10-15cm. Yet, as we demonstrated in our experiments, white noise (same as *pseudorandom noise*) alone was inadequate to mask the vibration sounds for a *co-located* attacker. Hence, we needed to explore further options to bolster the white noise masking signal in order to mitigate an eavesdropping attack from a co-located attacker.

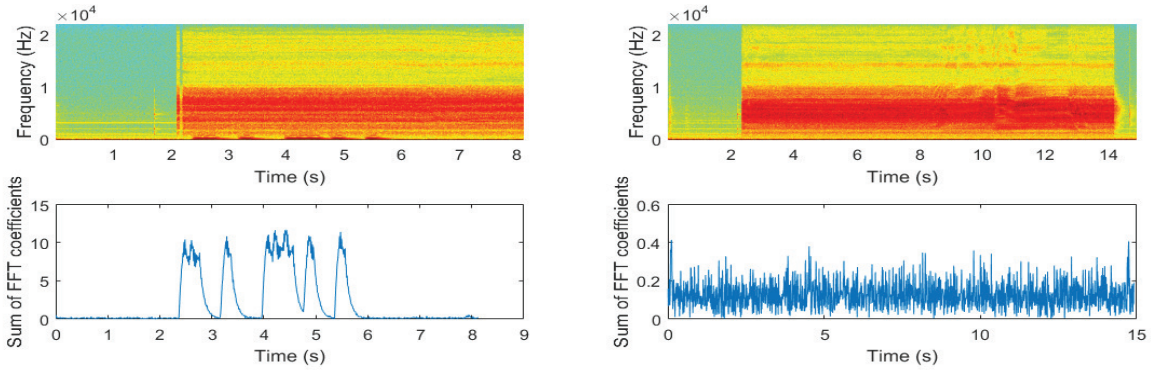
7.2 Novel Defense based on Low-Frequency Noise

Since white noise was unable to mask the vibration sounds at low frequencies, we tried to strengthen the white noise at the lower frequency band of 50Hz-250Hz against a co-located adversary. We also faced some challenges in the implementation of the proposed masking signal with the attack setup described in Section 7 that we will describe here. Lastly, we evaluated the efficiency of the proposed masking signal against sophisticated attacks and its effect on vibrational sensors of the receiving device.

7.2.1 Masking with Vibrations and Low-Frequency Tones

As observed in the previous section, white noise alone proved ineffective at masking low-frequency vibration sounds. To overcome this shortcoming, we considered other signals that could prove effective at masking audio leakage at low frequencies. We tried to add sounds that are acoustically similar to vibration sounds to confuse the adversary between the real vibration sounds and the pre-recorded vibration sounds mixed with the white noise. We recorded vibrations of Droid X2 phone from our setup with the inbuilt microphone, with the phone placed on a glass surface. Ripple [20] indicated a glass plate as producing the strongest side channel leakage when the vibrating device is placed on it. This motivated us to record the vibrations on a glass surface (henceforth referred as fake vibration sounds) as stronger the vibration sounds, stronger would be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Frequency spectrogram for audio recorded at a distance 0cm. (b) Frequency spectrogram for audio recorded at a distance 10cm.

Fig. 6. Difference in audio characterstics of recorded vibration sounds at close (0cm) and far (10cm) distances. Color intensity in the top graph is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

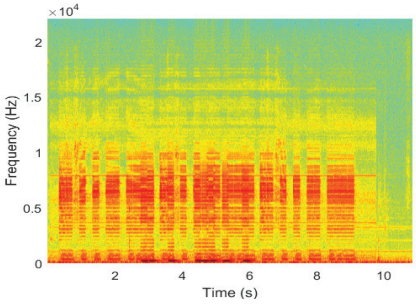


Fig. 7. Frequency spectrogram of real vibration sounds mixed with fake vibrations sounds. Color intensity in the spectrum is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency.

their recording, producing similar effect as actual vibration sounds when played back during the pairing process.

After recording fake vibration sounds, we played it on the device during pairing process to gauze similarity of the fake vibration sounds with audio leakage of real vibration on the frequency spectrum. Figure 7 shows the resulting frequency spectrum. As it is clear from the spectrum that while fake vibration sounds matched the audio leakage from the real vibration to most extent, they lacked the required low-frequency components contained in the audio leakage. Hence, they offered no better protection over the white noise masking signal and neither did the combination of both the white noise and fake vibration sounds (due to their inefficiency at lower frequency range).

The lack of low-frequency components in the fake vibration signal implored us to explore software based solutions for improving the quality of the audio playback to recover the desired low-frequency response. Since, our device ran on Android platform, we utilized audio effects and controls offered by the platform via AudioTrack API (Application Programming Interface) but no improvements were observed and boosting the signal only resulted in clipping of the audio signal, a phenomenon explained below:

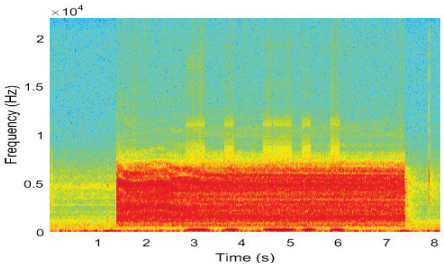
Non Linearity: This phenomenon is widely encountered in electrical circuits e.g. an amplifier, where the generated output signal strength is not directly proportional to the input signal strength.

Clipping: This phenomenon occurs due to distortion of the waveform when an amplifier is over-driven by trying to produce an output signal, the strength of which is beyond the specified limits of the amplifier. This causes the signal to be clipped at the limits resulting in a distorted wave. A side effect of clipping is the introduction of harmonics of the signal at higher frequencies.

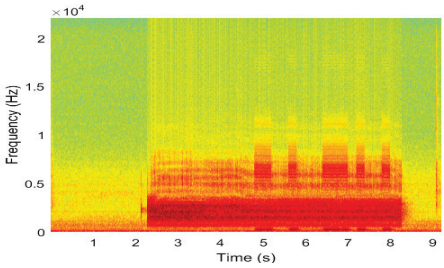
The next choice in our experiments was to generate tones in the desired frequency range and add them to the white noise to obfuscate the audio leakage from the vibration sounds. For this purpose, we used the Tone Generator function in Audacity along with the Noise Generator, and used "mix and render" functionality to produce the combined signal that is a mixture of white noise and a sinusoidal tone of 150Hz. The resulting observations are shown in Figure 8a. As Figure 8a shows, there was no masking at lower frequency band despite the introduction of a low-frequency (150Hz) tone. In particular, there was no presence of the tone at the intended frequency level. This behavior was similar to that of fake vibration sounds which also lacked the low-frequency components present in the audio leakage. We further investigated the issue by trying to reproduce various low-frequency sounds on two devices: Motorola Droid X2 and LG G4 smartphones. Droid X2 is an old smartphone, first released in 2011 whereas G4 is one of the latest devices announced in 2015.

During our attempts to reproduce low-frequency tones while testing the speakers of both old (Droid X2) and new (LG G4) devices, we re-encountered the non-linear behavior of the speaker response. The output audio signal for low-frequency tones was very low, barely registering on the microphone. Any attempts to increase the gain would inadvertently result in clipping of the signal producing unwanted harmonics at higher frequency levels with no improvement at the intended low frequency. We expected better performance from LG G4 smartphone featuring an improved speaker but the results were only slightly better (Figure 8b). The speaker was barely an improvement over

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Frequency spectrogram for the vibrations recorded by Droid X2.



(b) Frequency spectrogram for the vibrations recorded by LG G4.

Fig. 8. Spectrum for acoustic leakage captured through different devices. Color intensity in the spectrum is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency.

the Droid X2 speaker suffering from the same drawbacks of non linearity and clipping. Since the inbuilt speakers of the smartphones did not fulfill our purpose, we turned to other setups where we could obtain better speaker quality for improved sound reproduction.

7.3 Enhancing Vibreaker with Low Frequency Tone

While smartphone speakers may be limited in their capacity to reproduce low frequency sounds (sub 300Hz), we can try to boost their capabilities by complementing them with better hardware. Such an approach has been used in the industry by introducing a case like system with built-in speakers and/or a separate audio engine to boost the quality of smartphone's speaker [24], [25], [26]. While [24] and [25] are geared towards iPhones, [26] is offered as an accessory for Moto Z family of phones. These accessories can be put on as a case on the phone (Figure 9).

Fig. 9. JBL Soundboost case for Motorola Z series of smartphones.

We simulated this concept by taping a small portable speaker to our device and playing the sound through it. This setup also emulated the scenario where the receiving device could have an inbuilt powerful speaker like a payment terminal or high end media devices e.g. a smart television. For our experiments, we used three different portable speakers Altec [27], Sony SRS-XB2 [28] and JBL [29]. The frequency specifications for the tested speakers are presented in Table 1. In order to test the effectiveness of speakers in producing low frequency sounds, we played a 150Hz sinusoidal tone through each speaker and observed the recorded signal in frequency domain.

Since a tone below 150Hz distorted the response from Altec speaker, we used 150Hz tone in our next stage of experiment.

We connected the speaker to the smartphone via an audio cable (or bluetooth) while rest of the experimental setup was similar to our previous attack experiment (Section 7) under similar threat model (Section 4). The masking sound that was used to obfuscate audio leakage was a mix of white noise and a low-frequency tone (150Hz). We generated two separate tracks containing white noise (generated using noise generator functionality in Audacity) and 150 Hz tone (generated using tone generator functionality in Audacity) which were then mixed and rendered to form a new track. The low-frequency tone helped in masking the low frequencies of the audio leakage while the white noise spread across rest of the frequency spectrum masked the audio leakage at higher frequencies.

During our experiments, we observed the effectiveness of masking signal against co-located adversary. We also observed the effect of sound level of the masking signal in the event of clipping. This was of particular importance as we were operating around the lowest frequency response for some of the tested speakers.

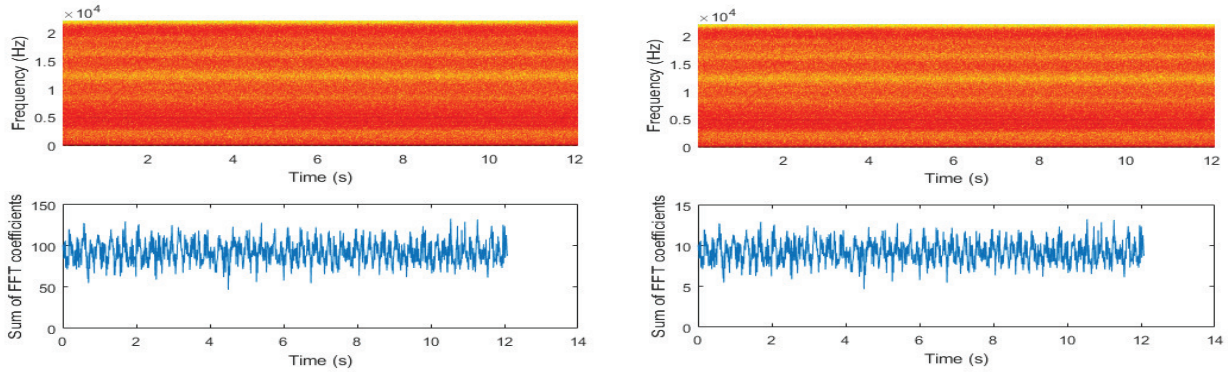
TABLE 1
Frequency response for tested speakers

Speaker	Frequency Response (in Hz)
Altec Lansing Mini H2O Speaker	Not specified
JBL Clip Portable Bluetooth Speaker	160-20,000
Sony SRS-XB2 Speaker	20-20,000

The results for the portable speaker are shown in Fig.10a. The frequency spectrum did not show the presence of audio leakage resulting from vibrations, particularly at low frequencies (50Hz-250Hz). The graph of the sum of FFT coefficients vs time showed that the quality of audio leakage degraded to an extent that it became very hard to choose a suitable threshold to determine a constant period of vibration. While the spikes in the graph may indicate towards presence of vibration, the resulting pattern could not be decoded into a valid PIN making the detection infeasible.

This observation showed that external portable speaker had the required sound reproduction quality that was found lacking on the inbuilt smartphone speakers. We also measured the sound level of the masking signals via a sound level measurement application for Android phone and recorded the sound level at a distance of 10cm. We observed that the optimal sound level for producing low-frequency sound of an amplitude sufficient to mask the audio leakage from vibration sounds was around 58 decibels. This sound level is approximately equal that of conversational speech and thus not considered harmful to the human ear.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Frequency spectrogram of the audio signal in presence of masking sound

(b) Frequency spectrogram of the audio signal in presence of masking sound after noise filtering.

Fig. 10. Characterstics of the audio signal in presence of masking sound from Sony SRS-XB2 speaker before and after noise filtering. Color intensity in the graph is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

7.3.1 Security under Sophisticated Attacks

In this section, we will evaluate the masking effectiveness of the white noise boosted with low-frequency tones against some attack vectors that may be used by the adversary for a more sophisticated analysis of the eavesdropped signal. The attack techniques that we will discuss here would involve noise filtering and source separation techniques.

Noise Filtering: The defense mechanisms that we studied till now, relied on deliberate injection of a masking signal in the environment for obfuscating the audio leakage during vibrational pairing. From the adversary's point of view, the masking signal was the noise accompanying the audio leakage (that was to be acquired and decoded). Hence, the adversary could try to remove or suppress the noise using noise removal algorithms.

Since a co-located adversary had the ability to process the eavesdropped signal offline and recover the information from the audio leakage, we repeated the attack experiment (Section 7) according to our threat model (co-located attacker) with the masking signal comprising of white noise with a low-frequency tone of 150Hz that was capable of masking the audio leakage from the vibrations at the low-frequency bands as detailed previously.

To evaluate the efficiency of our masking signal against noise filtering, we applied the noise reduction technique called "spectral noise gating" to the eavesdropped signal. This technique is used in most of the audio processing software tools like Audacity. We chose a short sample from the eavesdropped signal as the noise profile and applied it to the signal to be removed as noise. This process could be repeated multiple times until satisfactory results were obtained. The results for the Sony SRS-XB2 speaker are shown in Fig.10b. and they show no indication of the audio leakage from the vibrations in relevant part of the frequency spectrum. This affirmed the effectiveness of masking signal at hiding the audio leakage from the vibrations.

7.3.2 Effect on Vibrational Sensing

In a pairing mechanism based on vibrations like PIN-Vibra [9], the receiving device uses its accelerometer to read the vibrations and then decode it based on the protocol. The masking signal, proposed in this work, comprised of a low-frequency tone along with the white noise. The bass effect of the low-frequency tone has a tendency to produce deep rumbling sounds that have the capability of producing faint vibrations in the speaker. This effect may negatively affect the accelerometer readings of the receiving device that could have an negative impact on the accuracy of the vibrational decoding and thereby the success of the pairing process.

In order to test the impact of the masking signal on the ability of the receiving device to decode the vibrations correctly, we collected accelerometer readings in the background on the receiving device during the vibrational pairing in the presence of masking signal (as proposed in Section 7.3). We recreated the experiment setup as in Section 7.3. We play the masking sound (white noise mixed with a 150Hz tone) at different loudness level and gauge the effect of the masking sound on the receiver as measured by the accelerometer. We use Motorola Droid X2 as the sender, Sony SRS-XB2 as the external speaker generating the masking sound, and Samsung Galaxy S6 as the receiver.

We plot the bit error rate for the receiver against multiple thresholds for the tested loudness levels. In Section 7.3, we proposed 58dB as an appropriate sound pressure level for the masking sound. A louder masking sound would always be better at hiding the vibration sounds but may also cause vibrations on the smartphone (especially low frequency tones) that may impact the readings on the receiving device. Figure 11 shows bit error rate at the receiving device in presence of different loudness levels of the masking sound. We observe that at all the loudness levels ranging from 53dB till 82dB, for at least one threshold value, bit error rate falls to 0 i.e. the transmitted PIN is successfully decoded.

7.4 Security against Motion Sensor Exploits

While our threat model consists of a co-located acoustic eavesdropping adversary that exploits vibrations sounds, there exists another exploit that eavesdrops on motion sensors thereby compromising vibration based pairing protocols. This class of co-located adversary can trick the unsuspecting victim in installing a malicious application on their IoT device (smartphone) that eavesdrops on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

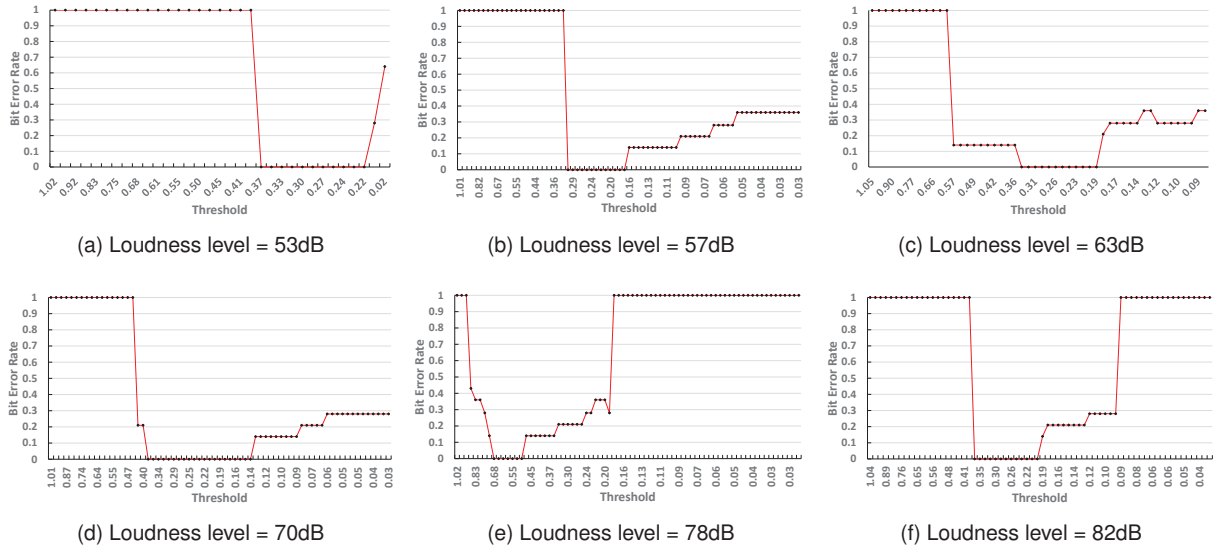


Fig. 11. Bit error rate against threshold values for PIN "4562" at the receiving device with different loudness level of masking sound. Since we begin from maximum threshold value, the x axis is in descending order.

motion sensor readings and forwards them to the attacker. The attacker has now access to the same sensor readings as the receiving device thereby leading to leakage of the transferred PIN/pairing information.

This exploit works due to the underlying assumption that the vibration channel (AS-OOB channel in the protocol) can not be eavesdropped directly, which is not true for this exploit. Android platform also does not impose any usage permissions (zero-permission) on an application for reading the motion sensor readings. This leads to an unrestricted access for a malicious application that can monitor accelerometer readings during the vibrational pairing process and extract the transferred PIN/passphrase.

Our threat model only considers a co-located adversary on the transmitting device (IoT hub) as the device already knows the PIN/passphrase that it is going to transfer while the effect on its accelerometer during the vibrational pairing leaks this confidential information. The receiving device in our model is usually a dedicated IoT device (smart lock, smart bulb, smart thermostats etc.) that usually would not allow access to internal sensors as opposed to the IoT hub i.e. smartphone which is a multipurpose device and allows unrestricted access to motion sensors.

We observed the effect of vibration pattern generated during the pairing protocol on the accelerometer sensor. The readings along z axis are shown in Figure 12a and it can be observed that the pattern is visible in the plot. These vibrations are also visible in the spectrum as dark red bands in Figure 12c. Thus, these readings can be exploited by an adversary to extract information about the PIN/passphrase transferred during the vibrational pairing. Mohamed et al. [10] proposed SMASheD framework for stealthily modifying various sensors on Android platform. SMASheD uses Android Debug Bridge (ADB) for installing a service on Android along with shell privileges that allow reading and writing to sensor files. For launching this attack, two scripts are used that push the service from the computer to the android device via USB and then launch it. Mohamed et al. [10] used the proposed framework as a potential security exploit against Android sensors. Shrestha et al. [11] incorporated SMASheD as a potential defense mechanism "Slogger" against sensor based touchstroke logging attacks. Touchstroke logging attacks work by inferring the start and end of a finger touch on a smartphone's screen by eavesdropping on motion sensor readings. Slogger aims to inject fake sensor readings into the system file that logs the motion sensor readings from the sensor hardware. The fake readings serve as a noise cover to the susceptible real sensor readings. We apply a similar idea as "Slogger" to defend against this class of motion sensor exploiting adversaries that could learn the vibration patterns from the motion sensors of the transmitting device (IoT hub).

We used the SMASheD framework to install a service on the transmitting device (Droid X2 in our experiments) that injects fake accelerometer readings into the accelerometer system file. The accelerometer system file logs the accelerometer readings obtained from the sensor hardware and allows unrestricted "read"operation. Since the injected readings from SMASheD should be similar to accelerometer's response during the transmitting device's vibration, we determined the minimum and maximum accelerometer readings during vibration along x, y and z axis to imitate the actual accelerometer readings during vibrations.

Once we determined the minimum and maximum accelerometer readings during vibrations, we generated random numbers in that range and injected them into the accelerometer system file at random intervals lasting less than time duration between successive accelerometer events (inverse of sampling frequency and measured in milliseconds). The minimum and maximum readings for accelerometer sensor during a vibration event were <-70, 120>, <-80, 80>, and <0, 1600> for x, y and z axis respectively. We plot the accelerometer readings, along z axis, as recorded by a malicious application residing on the transmitting device and eavesdropping on accelerometer readings while our defense injects fake sensor readings similar to vibrations

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

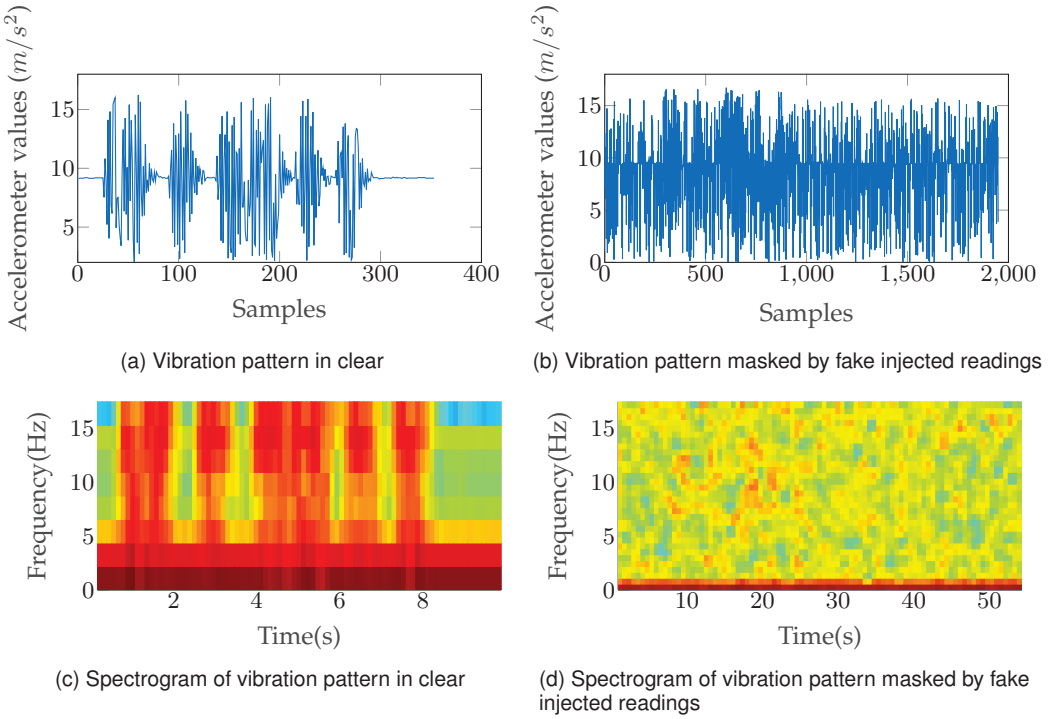


Fig. 12. Vibration patterns as recorded on accelerometer for PIN "4562" in absence and presence of fake injected readings

that mask the real vibration events in Figure 12b. When compared to Figure 12a, we are unable to determine which sensor readings correspond to actual vibrations and which readings correspond to fake injected vibrations.

Observing the spectrum in presence of our enhanced defense in Figure 12d, we notice that the red bands, corresponding to the vibrations in Figure 12c have completely disappeared. Since the injected fake accelerometer readings are similar to the vibration readings, there no longer exists a contrast between the background noise and vibrations in the accelerometer signal. Our defense has enhanced the background noise with fake readings so that it is now comparable to vibration patterns and so we are unable to spot any features that may characterize vibrations in the spectrum. Thus we are able to mask the effect of vibrations affecting the accelerometer during the pairing process.

Thus, we believe that the masking signal may not have any effect on the decoding accuracy of the receiving device while enhancing the security by obfuscating the audio leakage resulting from the vibrations at the same time.

8 Vibreaker against Remote Attack

So far, we have discussed acoustic eavesdropping attacks involving a *proximity* attacker and a *co-located* attacker eavesdropping on pairing of IoT devices. The advantage of a *proximity* attacker lies in the fact that proximity scenario is more plausible and allows the attacker to eavesdrop on the acoustic leakage covertly. On the other hand, when the attacker is co-located on the device itself, the attack itself becomes more powerful and just masking the acoustic leakage emanating from vibrations during pairing using background noise (white noise) is not sufficient. In this section, we will discuss a new attack model that involves a

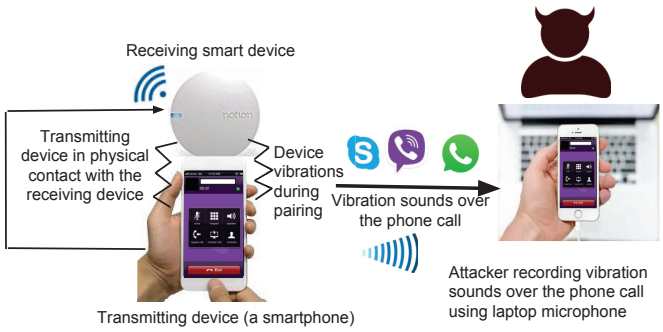
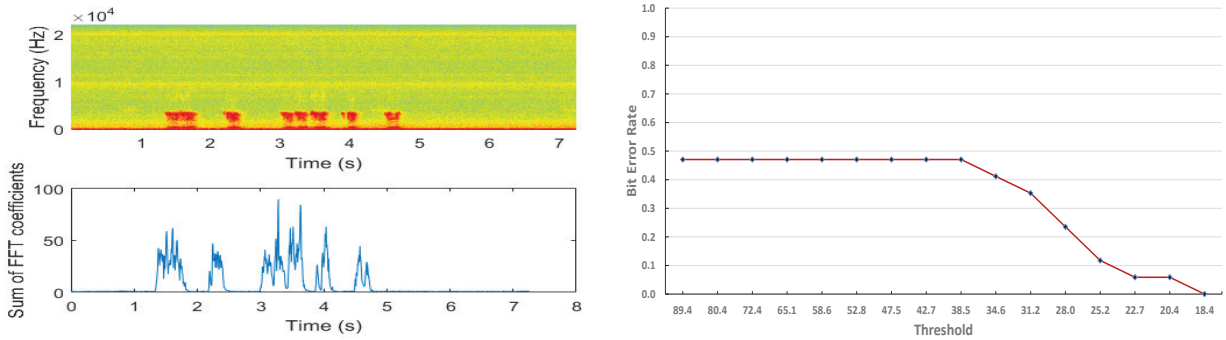


Fig. 13. Remote eavesdropping attack model for vibration based pairing of IoT devices

remote attacker eavesdropping on the vibrations during the pairing process. In this attack model, we assume that the user is trying to pair her device (for example, smartphone) to her smart devices while she is on call using the smartphone. As a real life example, a user may already be on call with another person while she attempts to pair her phone with one of the smart devices say, a smart thermostat. To pair the phone with the thermostat, she momentarily places the phone in contact with the thermostat without hanging up the call or placing the call on hold. This is a possible use case scenario as the pairing process disrupts the call only momentarily taking only a few seconds.

We also assume that the entity on the other end of the user's call plays the role of a malicious adversary or attacker. Since the user has no way of knowing if the attacker is eavesdropping on the conversation, the attacker can choose to record the whole call or wait for the moment when the user begins the pairing process. For best quality

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing



(a) Frequency spectrum of the audio signal as captured by the attacker.

(b) Bit error rate plotted against threshold values. Since we begin from maximum threshold value, the x axis is in descending order.

Fig. 14. Acoustic characteristics of the vibration leakage and acuracy of an attacker for PIN "4562" under remote attack model. Color intensity in the spectrum is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

of eavesdropping, the attacker may choose to put the call on speaker so as to amplify the vibration sounds. The attack model is described in Fig.13. The transmitting device while on call with the attacker is used for pairing with a smart device by bringing it in physical contact with the receiving device. During the pairing, the transmitting device generates vibrations that are picked by the receiving device and decoded for pairing. However, the sounds generated during vibrations are also picked up by the microphone of the transmitting device and thereby are delivered to the attacker by the voice call connection.

8.1 Attack Experiment under Remote Attack Model

The experiment setup for the attack model constitutes a user calling the attacker using her smartphone. We used Nexus 5X phone at the user's end to place and call and also to act as the IoT device generating the vibrations. We also use Samsung Galaxy S6 as the receiving device that reads the vibrations, decode it and authenticate the transmitting device. On the other end of the call, the attacker uses IPhone 6s to receive the call and Macbook Air (2013) inbuilt microphone to record the vibrations. While we used normal phone call during the experiment, any voice call applications (Skype, Viber, Whatsapp etc.) can be used for this purpose.

The spectrum of the captured vibration sounds on the attacker's end are depicted in Fig.14a. Comparing this frequency spectrum against frequency spectrum observed in Fig.1b and Fig.6a shows that the audio emanations from vibrations are more distorted and diffused unlike the captured audio in *proximity* attack model and low frequency audio captured under *co-located* attack model. To decode the captured audio, we use the same algorithm as in *proximity* and *co-located* attack models described iin previous sections but introduce minor changes to some of the parameters. We restrict the energy estimate in the frequency band of 3kHz to 4kHz and increase the window size for each vibration from 200ms to 210ms. The corresponding bit error rate plotted against threshold is shown in Fig.14b.

The bit error rate starts at 47% because the binary representation of PIN 4562 has 8 bits set and for maximum threshold value, the bit string is all zero bits leading to a bit error rate of $8/17 = 0.47$. For a threshold value of 18.4, the bit error rate drops to 0% indicating successful recovery of the PIN from acoustic leakage of vibrations as captured by the remote attacker.

8.2 Vibreaker against Remote Attack

As we discovered in previous section that vibration based pairing for IoT devices runs the risk of being eavesdropped upon if the IoT hub (i.e. smartphone) is on call with the attack using normal phone service or one of the many VoIP applications. To safeguards in such scenario, we tested our masking based defense, Vibreaker against a remote attacker as described in Section 8.1.

The frequency spectrum for the audio signal recorded at the remote attacker's end during the pairing process is depicted in Fig.15. On the spectrum, the acoustic leakage from the vibration of the IoT hub can be clearly seen in the lower frequency band (50Hz-300Hz). To quantify the acoustic leakage in the spectrum, we summed up the sum of FFT coefficient in a narrower frequency band of 200Hz-300Hz that we can observe in the lower subfigure of Fig.15. Using similar attack principles as described previously in this work, we were able to decode the correct PIN value that was transferred from the IoT hub to the IoT device in our experiment. The graph depicting bit error rate for different threshold values is seen in Fig.16. As we can observe in the graph, the bit error initially starts at 47% but drops down to 0% at a threshold value of 11.7 that indicates that the correct PIN value has been decoded by the attacker. This results is similar to performance of Vibreaker against a co-located attacker (Section 7) where white noise alone is unable to mask vibration sounds at low frequency bands.

8.3 Secure Pairing with White Noise and Low Frequency Tones

Since Vibreaker is unable to mask the acoustic leakage at sub 500Hz frequencies, we use the same defense mechanism that we utilized against a co-resident attacker due to similarities in the capabilities of the attacker. For both co-located and a remote attacker, the recording device is the inbuilt microphone in the IoT hub which is the closest location to the source of vibrations. In both scenarios, white noise alone is unable to mask the acoustic leakage from vibrations at low

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

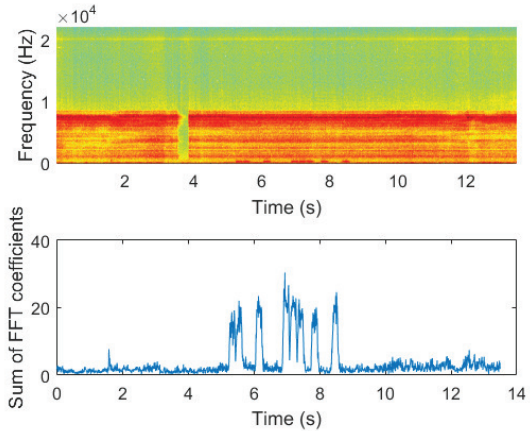


Fig. 15. Frequency spectrum of the audio signal captured by a remote attacker in presence of Vibreaker. Color intensity in the graph is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

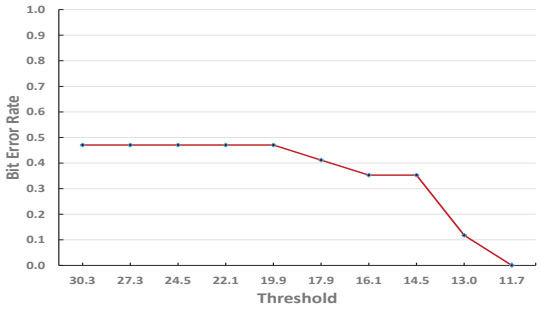


Fig. 16. Bit error rate plotted against threshold values in remote attack scenario under Vibreaker defense. Since we begin from maximum threshold value, the x axis is in descending order.

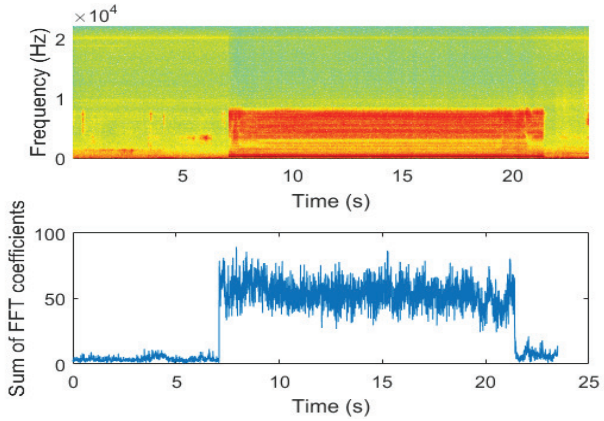


Fig. 17. Frequency spectrum of the audio signal captured by a remote attacker in presence of Vibreaker and low frequency tones. Color intensity in the graph is proportional to energy in the frequency band with blue color indicating lowest energy per frequency and red being the highest energy contained per frequency. Sum of the FFT coefficients indicates the estimated energy at the time instant.

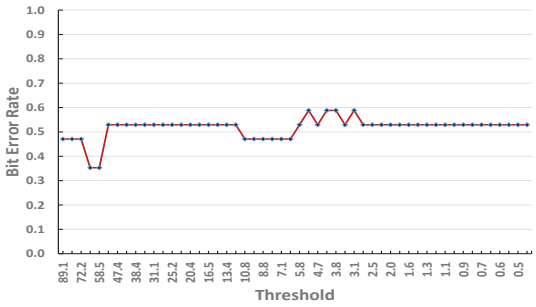


Fig. 18. Bit error rate plotted against threshold values in remote attack scenario under Vibreaker combined with low frequency tones. Since we begin from maximum threshold value, the x axis is in descending order.

frequencies so we boost the defense mechanism by adding low frequency tone to the white noise signal.

We play the enhanced masking signal (white noise added with low frequency tone) similar to the defense setup in Section 7.3. the frequency spectrum of the recorded audio signal from the remote attacker is depicted in Fig.17. The lower frequency band does not show any presence of acoustic leakage from the vibrations as opposed to Fig.15 which had only white noise to mask the vibration sounds. The graph of bit error rate against various threshold values show that the minimum threshold value that could be achieved was 35%. The bit strings obtained at both threshold values however were not valid bit strings as they lacked the necessary header ("110") that is attached to the bit representation of the PIN for indicating the beginning of a valid pairing transmission.

9 Summary and Discussion

The emerging field of IoT devices and their spreading use everyday exposes this nascent technology to various security risks. Since these devices are low energy with limited computational power, traditional security mechanism designed do not work in the context of IoT devices as expected. Additionally, their tendency to collaborate with each other by exchanging information makes a secure communication channel a necessity. Pairing of devices facilitates in establishing a trusted way of data transfer between devices and involves the exchange of a shared short key or passphrase. In this paper, we studied pairing among IoT devices (especially an IoT hub and other IoT devices) by using an auxiliary channel (OOB channel). In particular, we focused on the use of vibration as a means of communicating shared key or PIN among the IoT devices and evaluated its security against an acoustic eavesdropping attack.

We implemented an ON-OFF vibration scheme [9] and used it for pairing two IoT devices by transferring a 4-bit PIN. To create an acoustic eavesdropping attack, we simulated such an attacker as described in [18] that proposes an acoustic eavesdropping attack against pairing of constrained devices such as an IMD and its reader. We study the attack in multiple scenarios by placing the attacker at different locations. We also propose a signal masking based defense mechanism for thwarting the acoustic eavesdropper and evaluate it under the studied scenarios.

In a proximity eavesdropping scenario, we showed that an attacker can decode vibration sounds successfully from a distance of 15cm. We also showed that if white noise were to be injected in the environment surrounding the pairing devices, it would make it very hard for an attacker to extract

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

vibration sounds from the resulting audio that contains both acoustic leakage from vibrations and the masking signal. We also tested fake vibration sounds as possible masking signal for the acoustic leakage from vibrations and found it viable for securing the process. Enhancing the attacker's capability by using noise filtering tools did not increase the accuracy of the eavesdropping attack.

Our next attack scenario placed the eavesdropping attacker on the pairing device (IoT hub) itself in the form of a malicious application residing upon one of the device that had access to microphone of the device. Such a device would commonly be a smartphone but could also be smart speakers such as Amazon Echo or Google Home. For such an attacker, white noise defense is not effective at hiding the acoustic leakage. In particular, we found out that due to co-location of the attacker with the source of vibration, a better quality of acoustic leakage is picked up by the eavesdropper especially at low frequencies. As speakers found in low power constrained devices are designed to produce just acceptable sound, lower frequencies of the masking signal that are sub 500Hz may not be reproduced satisfactorily.

Since white noise reproduced from a low powered constrained device (such as many IoT devices) does not hide vibrations sounds at sub 1kHz frequencies, we looked towards introducing low frequency tones via an external speaker. This could remedy the shortcomings of on-device speakers responsible for masking acoustic emanations due to vibrations. We tested the enhanced defense mechanism with low frequency tone of 150Hz added to the white noise, both being generated via low-cost external speaker co-located with the device. We tested the effect of our enhanced setup on the decoding ability of the receiving IoT device and found it to absent from its sensor readings.

To evaluate our defense against an advanced attacker, we applied noise filtering algorithm and found out that noise filtering did not help the attacker in removing the masking signal from the eavesdropped audio signal and recovering the acoustic leakage of the vibrations. Such an attempt by an attacker resulted in the whole signal being reduced with barely any audio present in the filtered signal. While this defense setup required an external speaker, we believe that for smartphones, better quality external speakers already exists as phones cases or as external modules that can be attached to the phone without affecting their usability.

To complete the security of our defense, we proposed an additional measure that prevents a co-located adversary on the transmitting device to exploit the accelerometer readings to determine the information transferred during noisy pairing protocol. The proposed method works on a similar notion as noisy vibrations by injecting fake readings in accelerometer readings thereby masking the effect of vibrations of the transmitting device on its accelerometer. Since the injected readings are done programatically, the enhanced defense is transparent to the user. We also observed the effect of noisy vibrational pairing on the receiving device's capability of decoding the transmitted PIN/passphrase in presence of masking sound at different loudness levels. We show that masking sound at different loudness levels does not affect the ability of the receiving device to correctly decode the PIN/passphrase.

The rise of voice over IP (VoIP) applications has led to multiple communication options in addition to normal voice calling facility. We investigated the prospect of a malicious entity eavesdropping over VoIP calls including normal phone calls and potentially recording vibrations sounds during pairing. We recorded vibrations sounds over a normal call and discovered that these sounds were audible after being recorded on the other end of the call. Moreover, they could also be decoded using similar technique as previous eavesdropping attacks leading to full disclosure of the pairing secret being shared between the two devices.

We tested the remote eavesdropping attacker against Vibreaker and showed that such an attack model is successful in decoding the transmission into the exchanged information in a similar manner to a *co-located* attack model. The acoustic leakage from the vibrations was visible at low frequencies that demonstrated the inability of white noise in masking alone. We then tested the remote eavesdropping attack model against enhanced defense that combined Vibreaker with low frequency tone. The enhanced defense was found out to be sufficient at thwarting the remote attacker as the remotely captured audio signal could not be decoded correctly into the transferred PIN for all values of the threshold in the relevant frequency bands.

10 Conclusion

In this work, we showed that vibration based pairing protocols for constrained devices like those comprising IoT network, can be defended against an acoustic eavesdropping attack by adding artificial noise to the environment. We showed that traditional acoustic eavesdropping mechanism that places the attacker in proximity of the pairing devices, can be impeded by using white noise as a masking signal to cloak audio leakage resulting from vibrations. We also examined a novel scenario where the eavesdropping may be done through the device's microphone itself or a spying bug attached on the device. This setup, termed as "*co-resident* or *co-located*" attack, proved to be more potent than proximity attack as "*co-located*" attack is able to capture more details of the audio leakage than the proximity attack. To defend against our new attack model, we demonstrated that white noise alone was not sufficient and needed to be enhanced with the addition of low frequency tones through an external speaker. This step was also necessitated by the fact that smartphone speakers were not able to reproduce low frequency tones at sufficient intensity to cloak the audio leakage from vibrations in our experiments.

The "co-located" adversarial model was also expanded to include motion sensor exploits that can compromise the on-board motion sensors like accelerometer on the IoT hub to learn the transferred secret via vibrations. We extend our defense by complimenting it by injecting fake accelerometer readings that are able to mask the actual vibration effect on the accelerometer. We also examined the possibility of a remote attack model that records the acoustic leakage resulting from the vibration over a remote connection such as phone call or a VoIP connection. Such a situation is plausible if one of the devices involved in the pairing process is a smartphone on call with a dishonest entity. This attack model was successful against Vibreaker but enhancing Vibreaker with low frequency tones cloaked the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2873372, IEEE Transactions on Dependable and Secure Computing

acoustic leakage to a sufficient degree so as to impede the attack. Our analysis shows that while vibration pairing may seem to be an attractive mechanism for ensuring the security and trust in an IoT network, it needs to be protected against acoustic side channel attacks by defensive measures such as masking signals that are low cost and easy to implement.

REFERENCES

[1] S. A. Anand and N. Saxena, "Vibreaker: Securing vibrational pairing with deliberate acoustic noise," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '16, 2016, pp. 103–108.

[2] ——, "Coresident evil: Noisy vibrational pairing in the face of co-located acoustic eavesdropping," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '17, 2017, pp. 173–183.

[3] Global Standards Initiative on Internet of Things. (2012) Overview of the Internet of Things (ITU-T Y.4000/Y.2060). [Online]. Available: http://handle.itu.int/11.1002/1000/11559

[4] Gartner, Inc. (2017) Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[5] T. Halevi and N. Saxena, "On Pairing Constrained Wireless Devices based on Secrecy of Auxiliary Channels: The Case of Acoustic Eavesdropping," in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, Oct. 2010.

[6] R. Kainda, I. Flechais, and A. W. Roscoe, "Usability and Security of Out-Of-Band Channels in Secure Device Pairing Protocols," in *Proc. of Symposium on Usable Privacy and Security (SOUPS)*, 2009.

[7] V. Boyko, P. Mackenzie, and S. Patel, "Provably Secure password-Authenticated Key Exchange using Diffie-Hellman," in *Advances in Cryptology — Eurocrypt 2000: International Conference on the Theory and Application of Cryptographic Techniques Bruges, Belgium, May 14 – 18, 2000 Proceedings*, B. Preneel, Ed. Springer Berlin Heidelberg, 2000, pp. 156–171.

[8] 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. of of the 2008 IEEE Symposium on Security and Privacy*, 2008.

[9] N. Saxena, M. B. Uddin, J. Voris, and N. Asokan, "Vibrate-to-unlock: Mobile phone assisted user authentication to multiple personal RFID tags," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2011, pp. 181–188.

[10] M. Mohamed, B. Shrestha, and N. Saxena, "Smashed: Sniffing and manipulating android sensor data for offensive purposes," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 901–913, April 2017.

[11] P. Shrestha, M. Mohamed, and N. Saxena, "Slogger: Smashing motion-based touchstroke logging with transparent system noise," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '16, 2016, pp. 67–77.

[12] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong, "Talking to Strangers: Authentication in Ad-Hoc Wireless Networks," in *Symposium on Network and Distributed Systems Security (NDSS 2002)*, feb 2002.

[13] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and Clear: Human-Verifiable Authentication Based on Audio," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, 2006, pp. 10–10.

[14] 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, pp. 110–124.

[15] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure Device Pairing based on a Visual Channel," in *IEEE Symposium on Security and Privacy*, 2006.

[16] Y. Kim, W. S. Lee, V. Raghunathan, N. K. Jha, and A. Raghunathan, "Vibration-based secure side channel for medical devices," in *Proceedings of the 52Nd Annual Design Automation Conference*, ser. DAC '15. New York, NY, USA: ACM, 2015, pp. 32:1–32:6.

[17] E. Uzun, K. Karvonen, and N. Asokan, *Usability Analysis of Secure Pairing Methods*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 307–324.

[18] T. Halevi and N. Saxena, "Acoustic Eavesdropping Attacks on Constrained Wireless Device Pairing," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 563–577, March 2013.

[19] (2015) Dynex–USB Microphone–Multi. [Online]. Available: https://www.dynexproducts.com/pdp/DX-USBMIC13/6084114

[20] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: Communicating through physical vibration," in *NSDI*, 2015.

[21] C. A. Inc. (2017) B6 Omnidirectional Lavalier. [Online]. Available: http://www.countryman.com/b6-omnidirectional-lavalier-microphone

[22] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:14.

[23] Y. Kim, W. S. Lee, V. Raghunathan, N. K. Jha, and A. Raghunathan, "Vibration-based secure side channel for medical devices," in *Proceedings of the 52Nd Annual Design Automation Conference*, ser. DAC '15, 2015.

[24] AmpAudio. (2016, 4) AmpAudio. [Online]. Available: {https://www.ampaudio.com/}

[25] Zagg. (2016, 4) Zagg Speaker Case. [Online]. Available: {http://www.zagg.com/us/en_us/cases/iphone-6-case/speaker-case}

[26] Motorola. (2017, 3) JBL Soundboost Speaker. [Online]. Available: {https://www.motorola.com/us/products/moto-mods/jbl-soundboost-speaker}

[27] A. Lansing. (2016, 4) Altec Lansing - Mini H2O Bluetooth Speaker . [Online]. Available: {http://www.alteclansing.com/en/al-products/mini-h20-speaker/}

[28] Sony. (2017, 3) Sony SRS-XB2. [Online]. Available: {http://www.sony.com/electronics/wireless-speakers/srs-xb2}

[29] JBL. (2016, 4) JBL - Clip Portable Bluetooth Speaker. [Online]. Available: {http://www.bestbuy.com/site/jbl-clip-portable-bluetooth-speaker-purple/6050039.p?id=1219696711027&skuId=6050039}

S Abhishek Anand is a PhD candidate in Department of Computer Science at University of Alabama at Birmingham. He serves as a graduate research assistant with the Department of Computer Sciences, University of Alabama at Birmingham. His research interests include security of privacy-sensitive channels and side channel attacks. He was the recipient of the Outstanding Masters student in the Department of Computer Science, University of Alabama at Birmingham in 2015.

Nitesh Saxena is an Associate Professor of Computer and Information Sciences at the University of Alabama at Birmingham (UAB), and the founding director of the Security and Privacy in Emerging Systems (SPIES) group/lab. He works in the broad areas of computer and network security, and applied cryptography, with a keen interest in wireless and mobile device security, and the emerging field of usable security. Saxena's current research has been externally supported by multiple grants from NSF and NIJ, and by gifts/awards/donations from the industry, including Google (2 Google Faculty Research awards), Cisco, Comcast, Intel, Nokia and Research in Motion. He has published over 110 journal, conference and workshop papers, many at top tier venues in Computer Science, including: IEEE Transactions, ISOC NDSS, ACM CCS, ACM WWW, ACM WiSec, ACM ACSAC, ACM CHI, ACM Ubicomp, IEEE Percom, IEEE ICME and IEEE S&P. On the educational/service front, Saxena currently serves as the director and principal investigator for the UAB's Scholarship for Service (SFS) program and a co-director for UAB's MS program in Computer Forensics and Security Management. He serves as an Associate Editor for flagship security journals, IEEE Transactions on Information Forensics and Security (TIFS), and Springer's International Journal of Information Security (IJIS).