VibeRing: Using vibrations from a smart ring as an out-of-band channel for sharing secret keys

Sougata Sen
Northwestern University, USA

David Kotz
Dartmouth College, USA

ABSTRACT

Some Internet of Things (IoT) devices – a.k.a. "smart Things" – collect meaningful information when they are *in use* and in physical contact with their user (e.g., a blood-pressure monitor). A Thing's wireless connectivity allows it to transfer that data to its user's trusted device, such as a smartphone. However, an adversary could also establish a communication channel with the Thing, impersonate the user and obtain access to the user's information collected by that Thing. Thus, it is essential that the Thing connects to the *correct* user's device. Bootstrapping such communication channels usually require an out-of-band channel to share a secret, e.g., by asking the user to input a PIN. However, manual PIN entry is cumbersome, especially when a Thing is used transiently. In this paper, we investigate the use of *vibration*, generated by a custom Ring, as an out-of-band communication channel to unobtrusively share a secret with a Thing. This exchanged secret can be used to bootstrap a secure wireless channel over which the Ring (or another trusted device) and the Thing can communicate. We present the design, implementation, and evaluation of this system, which we call *VibeRing*. Through a user study we demonstrate that it is possible to share a secret with various objects accurately and securely.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**; *Ubiquitous and mobile computing systems and tools*; *Mobile devices*; • **Security and privacy**;

KEYWORDS

Smart Ring; Smart Thing; vibration; security; wearables; IoT

ACM Reference Format:
Sougata Sen and David Kotz. 2020. *VibeRing*: Using vibrations from a smart ring as an out-of-band channel for sharing secret keys. In *10th International Conference on the Internet of Things (IoT '20), October 6–9, 2020, Malmö, Sweden.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3410992.3410995

1 INTRODUCTION

Internet of Things (IoT) devices are increasingly found in smart homes, connected cars, or smart healthcare. Several personal IoT devices have the capability to sense and record information, to communicate wirelessly, and (sometimes) to actuate another device. These IoT devices, hereafter "*Things*", can either be personal or shared by a group of individuals like members of a household or office space. A Thing's user might pick it up and use it for a short period of time. Some examples of such Things include (a) the remote control for an AC or a TV, (b) wireless key fobs for car or garage doors, (c) handheld exercise equipment, (d) a smart mug or water bottle, or (e) a Continuous Glucose Monitor (CGM). During this transient usage, the Thing might be interested in knowing who is using it. Knowledge of its user's identity allows the Thing to make specific choices (e.g., blocking certain TV channels from children) or to preserve a user's privacy (e.g., not divulging sensitive blood-glucose readings to an adversary) by encrypting and transmitting information only to *that* user's trusted device, like a smartphone, while mitigating the threat of eavesdropping. This goal – for the Thing to identify and communicate with its user's trusted personal device – must begin with some form of secure communication between the user's smartphone and the Thing.

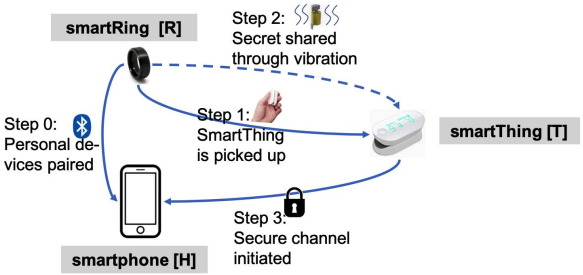


Figure 1: *VibeRing*'s key sharing operation: (1) individual picks up a Thing while wearing a Ring; (2) the Ring shares a secret with the Thing via the vibratory channel; (3) the secret is used to bootstrap a secure channel between the Thing and personal devices.

In this work we answer that foundational question – *how can a shared Thing quickly, securely and unobtrusively receive a secret from an individual who is briefly interacting with it?* This secret can then be used to bootstrap a secure, high-bandwidth, in-band channel (e.g., over Bluetooth or Wi-Fi) between the Thing and the user's trusted device. To answer this question, we explore using an out-of-band channel, vibration, generated by a *Ring* that is worn by the device's user to unobtrusively share a secret between the individual's personal device (smartphone) and the shared Thing that the individual is transiently using. We sketch the operation of the system – *VibeRing*– in Figure 1. Although an individual can interact with Things in numerous ways, we focus on the physical action of *picking up and holding* a Thing with the intention of using the Thing in its expected usage style.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IoT '20, October 6–9, 2020, Malmö, Sweden
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8758-3/20/10…$15.00
https://doi.org/10.1145/3410992.3410995

A trivial solution to this problem is to allow a user to explicitly input a 4-digit PIN code to bootstrap a secure in-band communication channel. However, many current and future Things do not have traditional input capabilities. Moreover, this manual process can be prone to shoulder-surfing attack and can be burdensome, especially when the number of Things an individual interacts with is large and the interaction is short-lived. Consider, for example, a healthcare practitioner who needs to access data from every visiting patient's CGM. Since we are focusing on transient interactions, it is difficult to justify explicit pairing between the devices. It must be noted that *we are not proposing a solution for pairing devices*, but rather are proposing a solution that allows a Thing to quickly establish a temporary secure channel with a trusted device (smartphone), so it can securely communicate with that user's smartphone.

One might envision using existing techniques (such as Near Field Communication (NFC) or ultrasound communication) for such situations. Although these techniques have low latency and high data rates, the number of Things with NFC capabilities is limited. Moreover, for NFC-enabled devices, the user has to deliberately bring the Thing in proximity to her smartphone, making such solutions obtrusive. Since many modern Things are already equipped with an accelerometer, we believe that using the vibration channel as an out-of-band channel will incur no additional cost and obviate the need for these secondary RF channels.

It is important that the key exchange does not disturb a person's natural interactions, i.e., the person does not have to perform explicit additional actions such as bringing a smartphone in contact with the Thing. It is intuitive that a finger-based vibration source, which is usually in close proximity to handheld objects, could be used for exchanging the secret without disturbing user actions. With the increasing interest in *smart jewelry* [13, 15, for example], rings become an obvious candidate for the secret-sharing source. We can assume that a ring serves as an identity proxy for an authorized individual.

This task of using vibration for sharing a secret from a Ring has several practical challenges: (a) Things can be of various shapes, and sizes, and a person can hold a Thing in many ways. So any solution must be agnostic to the Thing's composition or person's holding style. (b) The time of interaction between an individual and the Thing can be small; the system must transfer the secret quickly. (c) An adversary might be interested in the exchanged secret; hence, the system should ensure that information leakage is minimized.

Overall, in this paper, we make the following **contributions**:

- We describe *VibeRing*, a novel Ring-based system for bootstrapping a secure communication channel between a Thing and a user's smartphone, with minimal user involvement. We identify techniques through which *VibeRing* can mitigate various adversarial attacks.
- We demonstrate the possibility of using vibration from the Ring to share secrets. Through a study we show that it is possible to attain bit rates of 12.5 bits/sec with bit-error rate less than 2.5%, faster and more accurately than several existing techniques.
- Through a user study, we show that if the Ring and Thing are in contact, messages can be exchanged at 12.5 bps with a bit-error rate of less than 5%. For various materials, it is possible to share the secret with a message success rate of 85.9%.

2 RELATED WORK

We categorize works closely related to *VibeRing* as: (i) information exchange via vibration, and (ii) alternate exchange techniques.

2.1 Information exchange using vibration

In the past, several researchers have explored using a mobile device's vibration motor to transmit information, while an accelerometer on a nearby device could receive the message. Hwang et al. demonstrated the possibility of transferring data using vibration from a smartphone at 1 bps [8], while Yonezawa et al. improved the transfer rate to 10 bps [25]. Roy et al. explored approaches to improve the transfer rate further and achieved a data rate of 80 bps in a controlled setting where the smartphone was attached to a cantilever [18]. However, the transfer rate dropped substantially when the phone was held in hand. Kim et al. also explored using the smartphone's vibration motor to share secrets at 20 bps in a controlled environment [9]. More recently, Lee et al. used a smartphone's vibration motor to transfer a secret to a smartwatch, with a success rate of 92% [11]. In comparison to previous work, we test *VibeRing* in a substantially less controlled setting, and using a method that is less obtrusive for everyday usage.

Researchers have also explored using a vibration motor or accelerometer on devices other than smartphones. Wang et al. used a wrist-worn device to transmit information to several accelerometer-rich Things [22], while Yonezawa et al. developed a custom Thing that could transmit information through vibration [24]. In both these approaches, the authors attained a similar transfer rate as smartphones. *VibeRing* requires substantially less user intervention than these approaches. As an alternate to using an accelerometer for receiving vibration signals, Roy et al. demonstrated the possibility of receiving the vibration using a microphone [17]. However, the number of Things equipped with an accelerometer is substantially higher than those fitted with a microphone or other vibration receivers, thus making our approach more practical.

A limitation of the vibration channel is the possibility of audio leaks [7]. Researchers have proposed techniques to mask audio leakage by generating white noise [3] or other sounds [9]. In Section 6.3, we explore various strategies to mask audio leaks.

2.2 Some alternate techniques

NFC is a popular short-range communication technique. It can be used to share a secret, quickly, but several researchers have reported underlying security vulnerabilities in NFC [2, 5]. Other researchers have investigated non-accelerometer short-range information transfer techniques such as using the wearer's electromyography (EMG) signal to produce a secret [23], or a capacitive coupling system to share a secret [16]. Yet others have developed sound-based techniques for information exchange: Lee et al. proposed a system that used *chirp signals* that used ultrasound to transfer information at 16 bps [10], and Nandakumar et al. proposed a system for secure transfer using acoustic signals with nearby devices [14]. Unlike smartphones, however, neither microphones nor NFC are commonly available in Things. With the increasing availability of accelerometers in these devices, we believe that communicating through vibration will be feasible for Things.

3 MOTIVATION AND PROBLEM STATEMENT

The goal of *VibeRing* is to bootstrap a secure wireless communication channel between an individual’s trusted device (like a smartphone) and a transiently used Thing by sharing a secret between the devices via a Ring. This bootstrapping process should involve **minimum user interaction.** The secure channel can allow an individual’s device to know which Thing s/he is using and for the Thing to identify who is using it, **even during transient interactions**. Since the Thing might be used transiently and may be shared with others, it **does not warrant permanent pairing.** Consider the following scenarios where a Thing is used transiently and yet requires a secure communication channel. In a gym, smart dumbbells (weights) can securely transfer exercise details to their user’s smartphone, if they can securely connect to the correct smartphone. In a clinic, a patient’s CGM can wirelessly share its information with a healthcare practitioner’s terminal, if it can securely connect to the correct terminal. In a home where household members share everyday Things, such as smart mugs, a thermometer, or remote controls for televisions and air conditioners, such Things can communicate with the respective user’s smartphone, during each use. To enable personalization (e.g., segregating every house member’s body-temperature reading or customizing the settings of the home-entertainment system), the Thing must exchange information with its current user’s smartphone. Since the exchanged information might be sensitive (e.g., body temperature, mug usage statistics, or the identity of remote control’s user), the transfer must be encrypted using a secret known to the Thing and smartphone and not obtainable by any adversary. With *VibeRing*, a household member like Jack can wear a Ring while picking up any Thing in the house; Jack’s Ring automatically transmits a secret to the Thing and Jack’s smartphone, while ensuring that the secret is not captured by an adversary, Addy. Then the Thing and smartphone can use this secret to bootstrap a secure communication channel between themselves.

3.1 System model and assumptions

The *VibeRing* system consists of a *smartphone* (or another trusted device); a personal *Ring* that is worn by its possessor; and a *Thing* that may or may not belong to the individual, but is of interest to the individual. It is not necessary that the Ring (proxy for its wearer) has had any previous interactions with the Thing. All these devices have Bluetooth Low Energy (BLE) (or other Radio Frequency (RF) capability) for in-band data communication. All these devices are assumed to be capable of encrypting and decrypting the data sent over the RF channel. The Ring has a vibration motor, and the Thing has an accelerometer, used together for unidirectional communication to share a secret key from the Ring to the Thing. Additionally, the Ring shares the same key with the smartphone over an existing secure RF channel. This shared key is used to bootstrap a secure session between the smartphone and the Thing over the RF channel. Once the secure session is established, the Thing and smartphone can exchange any information so that the smartphone can learn *what* the Thing is, and the Thing can learn *who* is using it.

Our design rests on several assumptions. We assume the owners of the Thing and the Ring trust their respective devices. In case the devices are not owned by the same individual, then the individuals that own each device know and trust the other. We assume that the Thing is not in physical contact with another object (except the authentic Ring) while receiving the key from the Ring. We assume there exists a secure communication channel between the Ring’s RF interface and the smartphone. This communication channel is used by the Ring to inform the smartphone about the secret. By sharing a fresh secret with the smartphone and with the Thing, the Ring enables the Thing to contact the smartphone and establish a secure RF session, and to exchange information.

3.2 Adversary and threat model

In the smart-home scenario involving Jack, adversary Addy is aware that Jack’s Ring and an in-home Thing will share a secret so that the Thing can communicate with Jack’s smartphone. Addy may try to obtain this secret by impersonating either the Thing, the Ring, or the smartphone, or by eavesdropping. (We assume Addy can observe, modify, and inject transmissions on the RF channel.) We assume that Addy cannot break the underlying cryptographic methods used in the protocol. We also assume that Addy is not physically in contact with the Thing or the Ring; Addy can be in close proximity. (If Addy *were* physically touching the Thing, we expect Jack would notice; thus, it is reasonable to assume that Addy cannot physically touch the Thing while it is being used.) We assume that other types of attacks, e.g., gaining unauthorized access to the Ring, or DDoS attacks, are beyond the scope of this work; in Section 7 we justify these assumptions. We assume that there exists a permanent pairing between the Ring and the smartphone, both of which are personal devices belonging to the same user. The Ring will transmit the secret message over the RF channel it shares with only the smartphone. The potential for Addy to forcibly gain unauthorized access of the Ring and the smartphone, and pair itself with the Ring, is beyond the scope of this work.

If Addy can impersonate the Thing, then Addy can obtain the secret transmitted by the Ring and can decrypt all communication between the devices. If Addy can inject a secret so the Thing believes it comes from the Ring, then Addy can connect with the Thing and the Thing will send all its information to Addy. Thus *VibeRing* has three **security goals**: (i) data from the Ring can be decoded only by the Thing, (ii) data from the Thing can only be decoded only by the smartphone, and (iii) the Thing can verify that the message it receives via vibration is from the Ring, and not from Addy.

4 *VIBERING*

Figure 2 pictorially describes the working of the *VibeRing* system.

4.1 System overview

The Thing’s *pick detection* module keeps its components in a low-power mode until it is *picked up*. When picked up, the Thing’s accelerometer is sampled at a higher frequency and the BLE module starts advertising its presence. Simultaneously, the Ring’s pick detection module identifies a pick-up gesture. (In this paper, we assume that the aforementioned steps already exist. We work towards implementing the subsequent steps.) At this point, the Ring performs a BLE scan to listen for presence of Things. On receiving an advertisement from a Thing, the Ring generates a short random n-bit key K and transmits it as a message in the form of vibration;

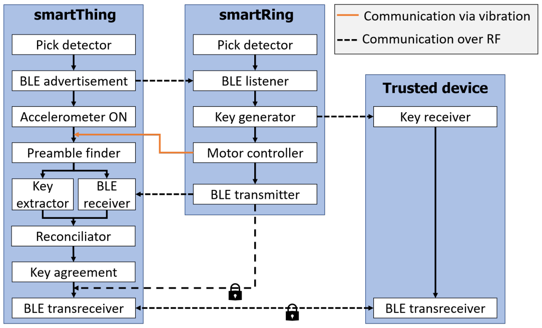


Figure 2: Working of *VibeRing's* secret transceiver

Section 4.2 provides details. The Ring also shares the key K with the smartphone using a secure RF communication channel. When the Thing's accelerometer detects the expected preamble, it processes the subsequent n bits to extract the secret message; Section 4.3 provides details. Next, the Thing updates its BLE advertisement to inform the Ring that it has received the message. The Ring and Thing then perform a key reconciliation step, as explained in Section 4.4. This step ensures that the Thing has the correct key and can use the message to encrypt the communication with the smartphone. Finally, the Ring encrypts details about the smartphone and transmits it to the Thing over the RF channel, allowing the Thing to directly communicate with the smartphone.

4.2 Key generation and transmission

Initialization: The key-exchange protocol begins once the Ring's pick detection module detects a pick-up gesture and the Ring's BLE scan discovers that a Thing is advertising nearby.[1] Once the Ring detects the pick-up gesture, and an advertisement from the Thing, we assume that the Ring is in physical contact with the Thing.

Key Generation: Next, the Ring generates a random key, K of length n. The Ring prefixes an 8-bit preamble to K and transmits the preamble and key in the form of vibrations. Although several vibration-based key-exchange techniques transfer a 4 decimal-digit PIN (e.g., [22]), *VibeRing* currently uses $n = 64$ bits.[2] The Ring also shares the same key with the smartphone using a secure RF channel.

Key Encoding: The Ring encodes the key using Manchester encoding, which allows the signal to be self synchronized. Because a vibration motor is made up of mechanical components, which have significant inertia, we observed that the motors have prolonged *ramping* and *damping* phases while transitioning from completely OFF or continuously ON states respectively. Manchester encoding prevents the motor from reaching a completely OFF or continuously ON state for more than b milliseconds during transmission, where b is the time taken to transmit a single bit. This is possible because in Manchester encoding the motor is ON only for $\frac{b}{2}$ milliseconds (first $\frac{b}{2}$ milliseconds to transmit 1 and second $\frac{b}{2}$ milliseconds to transmit 0) while transmitting a bit. *VibeRing* uses two motors: M_1, connected directly to the ring's shank, and M_2, connected to the ring's head, but has a padding between itself and the ring. M_1 is used to transmit the message, while M_2 masks audio leakage.

Table 1: Features extracted from each frame

id	Feature	Feature Description
F_1	average	average value of the samples in the frame
F_2	slope	slope of the linear regression line of the frame
F_3	change_s	change in the average of current frame from average of the previous frame
F_4	change_l	change in the average of current frame from average of the previous four frames
F_5	kurtosis	shape of the distribution of samples in frame

4.3 Key reception and extraction by the Thing

On detecting movement, the Thing (a) turns on its BLE radio to start advertising, and (b) increases its accelerometer sampling rate to receive the secret. It then applies the following processing steps:

Pre-processing and envelope detection: This step removes the gravity component and low-frequency hand movement. The Thing uses a band-pass filter to remove frequency components below 100 Hz and above 200 Hz, a range we selected empirically based on the motor's rotation speed (RPM). Next, the Thing rectifies the filter's output and extracts the envelope of the signal using a t-term moving average. The Thing uses the magnitude of the accelerometer data in subsequent processing. Our use of the magnitude captures acceleration observed on any accelerometer axis and makes the system agnostic to device orientation.

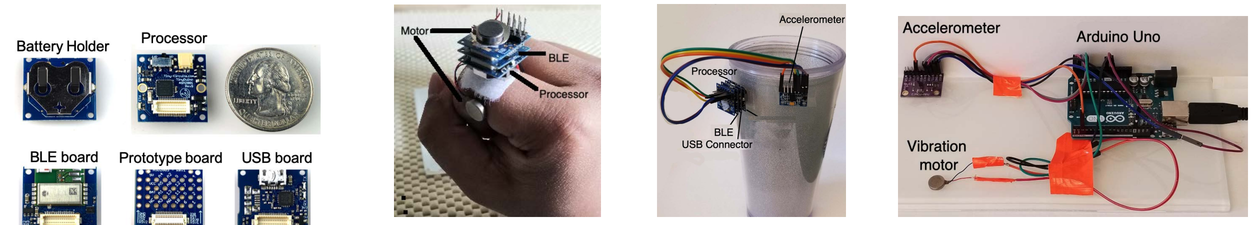
Preamble detection: The Thing next determines the start of the message by identifying the 8-bit preamble. To identify the preamble, the Thing uses a sliding window with 75% overlap on the accelerometer data. It applies the subsequently described *Framing, Feature Extraction,* and *Bit Extraction* steps to detect the preamble. On detecting a preamble, the Thing updates its advertised BLE name temporarily to indicate that it has started receiving vibration data.

Framing: Since the data is Manchester encoded, the number of frames that the Thing must extract is equal to twice the number of bits ($2n$) in the secret message. The Thing knows the value of n and the time taken to transmit a single bit (b), it thus uses ($n \cdot b$) milliseconds of accelerometer data to extract the frames.

Feature Extraction: The Thing first extracts a global feature, F_g, the average of the top-k amplitudes recorded in all frames. Since individuals hold objects with varied intensity, F_g is used to normalize the data. The Thing next extracts five frame-level features. F_1 is the average of the accelerometer data amplitudes recorded in the frame, normalized using F_g. When F_1 is greater than an upper threshold, the frame is considered a 'high' frame (i.e., frame value is 1), while F_1 less than a lower threshold is an indicator of a 'low' frame (frame value is 0). Otherwise, subsequent features are used to determine the bit. F_2 is the slope of the linear regression line of the frame; a steep positive slope is an indicator of a 'high' frame (since it indicates that the motor changed state from OFF to ON at the start of this frame), while a steep negative slope indicates a 'low' frame. For non-steep slopes, thresholds for F_3 to F_5 are used to determine bit value. F_3, change_short, is the change in the average of the data in the current frame from the previous frame. F_4, change_long, is the change in the average of the data in the current frame from

[1] We assume that there exists a pick-up detection module (e.g., as proposed in [19]). However, as an alternative, though more obtrusive, a small button could be embedded into the Ring and the wearer could press the button to initiate the key-exchange.

[2] 64 bits encodes more than 19 decimal digits, far stronger than the typical 4 decimal-digit PIN used by common pairing or device-unlock protocols [20].



(a) Components of the prototypes. **(b) The prototype Ring.** **(c) The prototype Thing.** **(d) The prototype adversary connected to a laptop.**

Figure 3: Components and prototype of the Ring, Thing and the adversary made using off-the-shelf boards for evaluation purposes; they would be smaller and suitably sized if engineered as a product.

the average of previous four frames. F_3 and F_4 capture information about whether the frame is undergoing a ramping phase or a damping phase. F_5, the frame’s kurtosis, provides the shape of the distribution in the frame. We tabulate these features in Table 1.

Bit Extraction: The next step is to combine information from two adjacent frames to generate a bit. Consider two frames (f_i, f_{i+1}) that together constitute a bit. In Manchester encoding we expect the two values to be different; thus:

$$\text{bit}= \begin{cases} 0 & \text{when } (f_i = 0 \ \wedge \ f_{i+1} = 1) \\ 1 & \text{when } (f_i = 1 \ \wedge \ f_{i+1} = 0) \end{cases}$$

Noise in channel may cause $f_i = f_{i+1}$, however. If $f_{i-1} \neq f_{i+2}$, then

$$\text{bit}= \begin{cases} 0 & \text{when } (f_i = f_{i+1} \wedge f_{i-1} = 0 \wedge f_{i+2} = 1) \\ 1 & \text{when } (f_i = f_{i+1} \wedge f_{i-1} = 1 \wedge f_{i+2} = 0) \end{cases}$$

In case the Thing still cannot infer the value of the bit, it sets the bit to zero and adds the bit position to its list of “disputed bit positions”. Now, the Thing has a key (K') that it received from the vibration channel.

4.4 Reconciliation

Ideally, K' is identical to the key K transmitted by the Ring. However, due to noise in transmission (or bit-manipulation attack by an adversary), certain bits in the transmission might get corrupted. To ensure that the Thing knows whether it has obtained the correct key, the Ring transmits an encrypted version of the key to the Thing over the BLE channel; specifically, it transmits $E = f(K, K)$, where $f()$ is a common symmetric encryption function, here encrypting payload K with key K. We currently use Arduino’s AESLib library for encryption as it adds little additional computational overhead.

Upon receiving K' via vibration channel, and E via BLE, the Thing decrypts the message using the decrypting function $D = g(E, K')$. If $D = K'$, then the Thing has received the key correctly, i.e., $K' = K$. However, if $D \neq K'$, the key was corrupted during transmission. If the number of disputed bits detected is small (typically ≤ 3 in our experiments), the Thing tries all possible bit values for these “disputed bits” and tries decrypting E using $g(E, K'_i)$, where K'_i is K' constructed with the i^{th} combination of disputed bit values. However, if the Thing still fails to decrypt E correctly, or if the number of disputed bits is large, then the Thing uses the RF channel to notify the Ring to send an Error Correction Code (ECC) via vibration, and it uses that ECC to correct more bits.

Error Correction Code: We analyzed various ECC techniques and decided to use the Hamming(m,n) code, where n is the length of the original message and m is the length of the message with parity bits added. The number of parity bits (r) needed is determined by the equation $2^r \geq m + r + 1$. For the 64-bit message with 8-bit preamble, $r = 7$ is necessary to detect 6 error bits and correct 1 error bit.

5 METHODS

Figure 3 presents the prototype devices. The current prototypes are only for experimental and evaluation purposes, built using development boards (shown in Figure 3a) from TinyCircuits [21].

Figure 3b presents the prototype Ring, which consists of two coin-type Eccentric Rotating Mass (ERM) vibration motors, a modified Arduino Uno board with Atmega328P MCU (TinyDuino ASM2001-R-L), and a board with STMicroelectronics’ BLE chipset (ST BLE TinyShield ASD2116-R). We use the I^{2}C bus to connect the two motors to the processor, one of which is attached to the shank of the ring, and the other is placed on top of the prototype board with padding inserted between itself and the board.

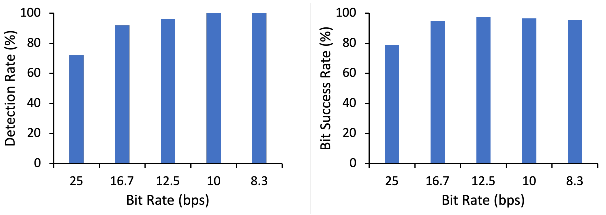
Figure 3c presents the prototype Thing, which uses the same processor and BLE chip as the Ring. Additionally, it has a USB connector (USB TinyShield ASD2101-R) for transferring the accelerometer data. A 3-axis accelerometer (MPU-6050), sampling at 500 Hz, is connected to the Thing. The Thing can lower the accelerometer’s sampling rate when it is not picked up and used.

The *adversary* (shown in Figure 3d) is an Arduino Uno board connected to an accelerometer (MPU-6500 sampling at 1000Hz) and an ERM motor. To log the accelerometer data, the Arduino Uno board is connected to a computer via USB. Additionally, the adversary uses the computer to record sound.

5.1 Dataset

We attached the Thing module to six everyday objects – a hard plastic tumbler (PT), a hard plastic box (PB), a glass tumbler (GT), a steel tumbler (ST), a metal block (MB), and a wooden box (WB). These objects represent hypothetical Things made of such materials. We collected data from a controlled study where we taped the Ring to a Thing and collected data, and a user study where we recruited participants who wore the Ring prototype and picked up the Things.

Controlled study (CS): We performed this study to determine the feasibility of the *VibeRing* system. We attached the Ring directly to the PT, 4 cm below the Thing’s accelerometer. The Ring transmitted 10 randomly chosen 64-bit messages that followed an 8-bit preamble. The ring transmitted each message at bitrates of {8.3, 10, 12.5, 16.7, 25} bps. For each bitrate, the ring transmitted a message 5 times. Thus, we collected 250 messages of length 72 bits.



(a) *PDR* **(b)** *BSR*

Figure 4: The variation of *PDR* **and** *BSR* **at various** *BR*

User study (US): We recruited 12 participants (5 males, 7 females; aged between 18 and 30) after obtaining approval from our university's Institutional Review Board (IRB). The participants' finger length and finger circumference varied between 7 cm to 9.5 cm, and 5 cm to 7 cm respectively. Each participant performed 27 distinct pick-up gestures. During every pick-up gesture, the Ring transmitted a message from a pool of messages at bitrates of {8.3, 10, 12.5, 16.7, 25} bps. We collected 135 messages from each participant.

5.2 Evaluation parameters and metrics

We next define the metrics used to evaluate *VibeRing*.

- *Bit Rate (BR):* the number of bits transmitted by the ring per unit time, measured in bits per second (bps). Higher *BR* allows quicker message transmission.
- *Bit Error Rate (BER):* the ratio of the number of incorrectly interpreted bits, even after reconciliation, to the total number of transferred bits. *BER* is affected by the *BR*. *VibeRing's* Bit Success Rate (*BSR*) is represented as $BSR = 1 - BER$.
- *Message Error Rate (MER):* the ratio of the number of messages with at least one bit error (after reconciliation) to the total number of transmitted messages. A lower *MER* indicates that the Thing could successfully transfer more secrets. *VibeRing's* Message Success Rate (*MSR*) is represented as $MSR = 1 - MER$.
- *Preamble Detection Rate (PDR):* The ratio of the number of preambles that were correctly detected to the total number of messages (preambles) transmitted.

6 EXPERIMENTS

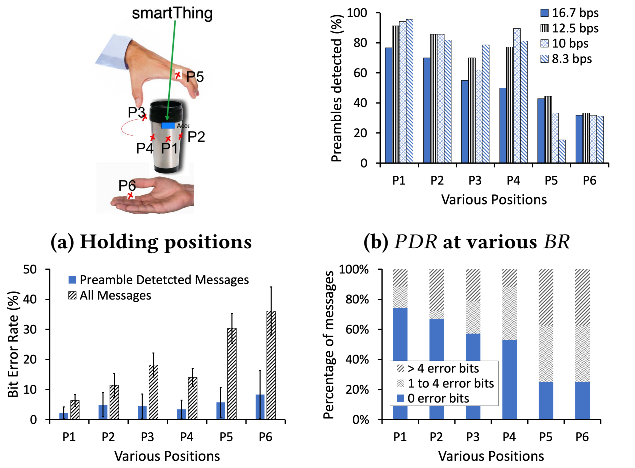
We next evaluate the performance of *VibeRing* by answering the following research questions:

A. How well can our method decode messages transmitted via the vibration channel?
B. What parameters affect the system's performance?
C. Can an adversary eavesdrop or spoof the message?

6.1 Inferring the transmitted messages

The first step in inferring the message is preamble detection. We analyze the 'CS' dataset to determine *PDR*. From Figure 4a we see that at 16.7 bps, the Thing could successfully detect the preamble in 92% of messages and it achieved 100% for 10 and 8.3 bps. At 12.5 bps, the *PDR* was 96% and the time required to transmit a 72-bit message (including the preamble) was 5.76 seconds.

Figure 4b shows the *BSR* for all messages where the Thing could detect the preamble. We observed that at 12.5 bps, the *BSR* was 97.5%. This indicates that in every 72-bit message, an average of 1.8 bits were interpreted incorrectly by the Thing. Overall, we observed that 76% of messages had 3 or fewer disputed bits, which the Thing could correct using the dispute-resolution approach. The effective *MER* at 12.5 bps (after accounting for modification of disputed bits and applying error correction) was 12%, indicating that the Ring could transmit a 64-bit secret to the Thing in less than 6 seconds in 88% of instances. The time taken to transfer the 64-bit secret is similar to the time taken to gesturally input a 7 digit PIN (20 bit), as shown by Ahmed et al. [1], but requires no effort from the user.



(a) Holding positions **(b)** *PDR* **at various** *BR*

(c) *BER* **at 12.5 bps** **(d)** *MER* **at 12.5 bps**

Figure 5: Effect of Holding Position on the *BER* **and** *MER*

6.2 Parameters affecting message detection

We next evaluate *VibeRing* when a user held (i) a Thing at various positions, and (ii) Things that are made of various materials. We use the 'US' dataset for this evaluation.

6.2.1 Effect of holding position. For the effect of holding position on message delivery, we use the data when the participant picked up and held the PT at positions illustrated in Figure 5a. For each position, three messages were transferred at all five bitrates. We empirically observed that for positions *P*1 to *P*4, usually the user's palm and fingers wrapped around the tumbler. For *P*5, only the finger was in contact with the tumbler, while for *P*6, only the palm was in contact with the tumbler's base. Figure 5b portrays *PDR* at various *BR*. Overall, *PDR* deteriorates as distance from the accelerometer increases. On average, for position *P*1, the Thing could detect over 90% of messages transmitted at 12.5 bps. This indicates that for high *MSR*, the accelerometer and motor should be in close proximity, a consideration for the design of Things in the future.

Figure 5c portrays *BER* at *BR* = 12.5 bps separately for messages whose preamble was identified, as well as all transmitted messages. We compute the *BER* as follows: (i) for messages whose preamble was missed, the Thing assumes that the bit inference is equivalent to random guessing (i.e., *BER* = 50%), and (ii) for messages whose preamble was detected correctly, *BER* is the ratio of bits that were incorrectly inferred (post reconciliation) to the total number of bits. From Figure 5c, we observe that for positions *P*1 to *P*4, the *BER* was less than 5% for messages whose preamble was detected.

Figure 5d shows the percentage of messages (for BR = 12.5 bps) with no error bits, 4 or fewer error bits (3 or fewer disputed bits and 0 or 1 error bit), or more than 4 error bits. Messages with more than 4 error bits includes messages whose preamble was not detected. For position $P1$, more than 88% of messages had 4 or fewer error bits, while for positions with little contact between the Ring and the Thing ($P5$ and $P6$), 36.1% of messages had 4 or more error bits. For $P1$, the system achieved a MSR = 88% after error correction. This indicates that to achieve high MSR, the Ring and the Thing should be in contact. Although a user may need to adjust her grip to attain good contact, this effort is still lower than manually inputting a secret. Also, it is unlikely that an adversary with no contact with the Ring can receive the message.

6.2.2 Effect of different materials. In 'US', participants picked up each of the six Things described in Section 5.1. During every pick-up, the Ring transmitted two messages at all specified bitrates.

The line plot in Figure 6 presents the BER for the different material types when BR = 12.5 bps. The Thing considers all messages while computing the BER. From the figure we see that for the GT and PT, the BER was less than 5%. The BER was higher for the ST because the Thing missed more preambles than it did for other tumblers. However, for the ST, the overall BER for messages whose preamble was correctly detected was 0.7%, lower than the PT. The overall BER was less than 11% for all items except the PB. We observed the manner in which participants picked up the PB and noticed that in several cases the box was not in contact with the Ring during the pick up gesture, thus further advocating the need for proper contact between the Ring and the Thing.

Figure 6 also presents the MER. For all three tumblers, where the Ring was in contact with the Thing, we observed that at least 70% of the messages were received with no error bits. In fact, the GT extracted 78.2% of messages with no bit error. The GT and PT could receive at least 83.3% of messages with 3 or fewer *disputed bits* and, after performing the error correction step, they could both extract at least 85.9% of messages. This result indicates that for objects where the Ring is in contact with the Thing, 70% of the message exchange can occur unobtrusively in less than 6 seconds, while in another 15.9% of messages, the exchange can successfully take place within an additional few seconds (depending on the time necessary to transmit the ECC).

6.3 Possibility of spoofing or eavesdropping

VibeRing's security goals are: (i) data from the Ring can be decoded only by the Thing, (ii) data from the Thing can only be decoded only by the smartphone, and (iii) the Thing can verify that the message it receives via vibration is from the Ring, and not from Addy.

6.3.1 Prevent Eavesdropping. We consider two approaches through which Addy can obtain the secret: (i) Addy captures the vibration directly, and (ii) Addy captures the sound of the vibration.

(i) *Vibration Leakage*: Participants in our study held the Thing (a) while the Thing rested on a glass tile, and Addy also placed an accelerometer on the same glass tile, 20 cm away from the base of the Thing, and (b) 25 cm from Addy's accelerometer, while being in contact with only the participant's hand. Addy's accelerometer was located on the glass tile. For (a), we found that at 12.5 bps, for messages with correctly detected preamble, Addy could extract messages with BER = 19% and MSR = 0% (without reconciliation). However, in 66% of cases, Addy could not detect the preamble, deeming the subsequently collected information worthless. When the Thing was not in contact with the glass tile (case (b)), Addy could not infer any transmitted bit.

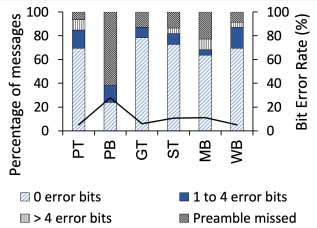


Figure 6: Variation of BER and MER for Plastic Tumbler (PT), Plastic Box (PB), Glass Tumbler (GT), Steel Tumbler (ST), Metal Block (MB), Wooden Box (WB).

(ii) *Audio Leakage and counter measures*: During preliminary studies we observed substantial acoustic leakage when the motor vibrated. To mask the audio leakage, we added a second motor, M_2 that vibrated in the opposite pattern as the original motor, M_1. The padding between M_2 and the ring reduced M_2's interference with M_1's transmission. However, we noticed that unless M_2 was touching the same material as M_1, at a similar contact intensity, the audio pattern from the two motors were distinguishable. We reduced the acoustic leakage using a white-noise approach [4]. Our prototype uses an external source to generate the white noise, but future prototypes will include this capability; we will also evaluate other techniques for using M_2 to mask the audio leak.

6.3.2 Prevent Spoofing. To understand whether Addy could impersonate the Ring (Addy sends a key to the Thing and creates a secure channel between itself and the Thing), we instructed participants to hold the hard plastic tumbler while the tumbler rested on a glass tile. Addy placed a vibration motor 20 cm from the base of the tumbler. During the secret sharing step between the Thing and the Ring, Addy transmitted a random message at the same bitrate and vibration intensity as the Ring's transmission. We found that due to the difference in the intensity of reception, the Thing could easily filter out vibration signals transmitted by Addy.

6.3.3 Summary. In Section 3, we stated the security goal of *VibeRing* as: (i) data from the Ring can be decoded only by the Thing, (ii) data from the Thing can only be decoded only by the smartphone, and (iii) the Thing could verify that the data they receive is not from Addy. Our experiments demonstrated that Addy could not collect the secret shared between the Ring and the Thing through eavesdropping. Without the secret message, Addy cannot decode the encrypted messages exchanges between the Ring, Thing and smartphone, thus addressing security goals (i) and (ii). For security goal (iii), we showed that the Thing ignored messages from Addy.

7 DISCUSSION AND FUTURE WORK

Several aspects for a vibration-based secret-exchange system still require further investigation.

Energy Drain: Energy drain of the Ring has two different aspects: energy drain due to the motor's functioning, and energy drain to keep the other electronic components functional. The motor that we currently use has a typical load power consumption of 165 mW, and its typical operating current draw is 55 mA. This translates to 13.2 mJ energy requirement for generating each bit of the secret, or over 1.16 hour battery life when a 3V, 130 mAh battery is used to drive the two motors continuously with a peak-to-peak amplitude of 1.9g (where g is the gravitational constant). Of course, *VibeRing* only drives the motor during the secret-exchange protocol – perhaps at most a few minutes per day – easily allowing the ring battery to last all day. For other electronic components, our prototype uses inexpensive off-the-shelf boards; these boards have additional components that, though unused by *VibeRing*, nonetheless draw power. Future prototypes, or any commercial product, would be built from custom printed circuit boards (PCB) and consume far less energy. Indeed, a commercially available Ring, the Oura ring [15], performs complex behavior-monitoring tasks, yet its battery lasts for days. Our vision for Rings and Things include a wake-up circuit (e.g., as used in [6]) to ensure they emerge from low-power mode only when they receive certain contextual cues, such as motion.

Eavesdropping: One might wonder whether an adversary could detect and decode the vibration signal, even when not in direct contact with the Ring. We conducted a small study in which we attached a motor (representing Ring) directly to a wooden plank, and an accelerometer (representing the adversary) at other points on the plank, and measured the vibration intensity at various distances from the motor. Overall, we observed that the intensity of the vibration signal dropped by over 40% when the distance from the motor increased from 5 cm to 45 cm. In this experiment, we measured the vibration intensity across a homogeneous material. However, in a free-living setting, the vibration will be further attenuated while traversing multiple mediums. An adversary might also eavesdrop using the visual channel [12]. Prior work has suggested techniques to mitigate such a leakage source – e.g., adding a pre-known pseudo-random vibration message [18]. *VibeRing* can use such existing techniques.

Usability: At the end of our user study, we asked participants to comment about any discomfort in using the system. Most participants reported that they felt the vibration was similar to a smartwatch's vibration and they felt that it was not disturbing.

8 CONCLUSION

In this paper we present the use of a Ring with vibration capabilities to bootstrap a secure communication channel with a Thing, even during transient interactions. This channel can allow Things to exchange information with its user's personal device, such as a smartphone. Such a system can share the secret reliably – with message success rate of 85.9% – while being robust to the way a person holds the Thing, and the Thing's constituent material. *VibeRing* will enable secure communication in Rings of the future without disturbing natural human actions.

ACKNOWLEDGEMENTS

We thank Varun Mishra from Dartmouth College for feedback on early drafts of this paper. This research results from a research program at the Institute for Security, Technology, and Society at Dartmouth College, supported by the National Science Foundation under award number CNS-1329686. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

REFERENCES

[1] Imtiaj Ahmed, Yina Ye, Sourav Bhattacharya, N. Asokan, Giulio Jacucci, Petteri Nurmi, and Sasu Tarkoma. 2015. Checksum gestures: Continuous gestures as an out-of-band channel for secure pairing. In *International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. ACM, 391–401.

[2] Nicholas Akinyokun and Vanessa Teague. 2017. Security and privacy implications of NFC-enabled contactless payment systems. In *International Conference Proceeding Series*, Vol. Part F1305. ACM.

[3] S. Abhishek Anand and Nitesh Saxena. 2016. Vibreaker: Securing Vibrational Pairing with Deliberate Acoustic Noise. In *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM.

[4] S Abhishek Anand and Nitesh Saxena. 2017. Coresident evil: noisy vibrational pairing in the face of co-located acoustic eavesdropping. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. ACM.

[5] Waldemar Berchtold, Patrick Lieb, and Martin Steinebach. 2017. Secure communication protocol for a low-bandwidth audio channel. In *European Signal Processing Conference (EUSIPCO)*. IEEE.

[6] Shengjie Bi, Tao Wang, Nicole Tobias, Josephine Nordrum, Shang Wang, George Halvorsen, Sougata Sen, Ronald Peterson, Kofi Odame, Kelly Caine, Ryan Halter, Jacob Sorber, and David Kotz. 2018. Auracle: Detecting Eating Episodes with an Ear-mounted Sensor. *In Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (9 2018), 1–27. https://doi.org/10.1145/3264902

[7] Rainhard Dieter Findling and René Mayrhofer. 2015. Towards Device-to-user Authentication: Protecting Against Phishing Hardware by Ensuring Mobile Device Authenticity Using Vibration Patterns. In *International Conference on Mobile and Ubiquitous Multimedia (MUM)*. ACM.

[8] Inhwan Hwang, Jungchan Cho, and Songhwai Oh. 2012. Privacy-aware communication for smartphones using vibration. In *Embedded and Real-Time Computing Systems and Applications (RTCSA)*.

[9] Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K. Jha, and Anand Raghunathan. 2015. Vibration-based secure side channel for medical devices. In *Design Automation Conference (DAC)*.

[10] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *Conference on Computer Communications (INFOCOM)*. IEEE.

[11] Kyuin Lee, Vijay Raghunathan, Anand Raghunathan, and Younghyun Kim. 2018. SYNCVIBE: Fast and Secure Device Pairing through Physical Vibration on Commodity Smartphones. In *International Conference on Computer Design (ICCD)*.

[12] Marci Meingast, Christopher Geyer, and Shankar Sastry. 2005. Geometric Models of Rolling-Shutter Cameras. *ACM Transactions on Graphics (TOG)* 33, 4 (2005).

[13] Motiv Ring. 2020. -. http://mymotiv.com/. Last Accessed: 04-04-2020..

[14] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. 2013. Dhwani: secure peer-to-peer acoustic NFC. In *Special Interest Group on Communication (SIGCOMM)*. ACM.

[15] Oura. 2020. Oura Ring. https://ouraring.com/ Last Accessed: 04-04-2020..

[16] Marc Roeschlin, Ivan Martinovic, and Kasper B Rasmussen. 2018. Device Pairing at the Touch of an Electrode. In *Network and Distributed System Security Symposium (NDSS)*.

[17] Nirupam Roy and Romit Roy Choudhury. 2016. Ripple II: faster communication through physical vibration. In *Conference on Networked Systems Design and Implementation (NSDI)*. USENIX.

[18] Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. 2015. Ripple: Communicating through Physical Vibration. In *Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX.

[19] Sougata Sen, Archan Misra, Vigneshwaran Subbaraju, Karan Grover, Meera Radhakrishnan, Rajesh K. Balan, and Youngki Lee. 2018. I4S: Capturing shopper's in-store interactions. In *International Symposium on Wearable Computers*.

[20] Yaniv Shaked and Avishai Wool. 2005. Cracking the Bluetooth PIN. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM.

[21] TinyCircuits. 2020. -. https://tinycircuits.com/. Last Accessed: 5-5-2020..

[22] Wei Wang, Lin Yang, and Qian Zhang. 2016. Touch-and-guard: secure pairing through hand resonance. In *International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. ACM.

[23] Lin Yang, Wei Wang, and Qian Zhang. 2016. Secret from Muscle: Enabling Secure Pairing with Electromyography. In *Conference on Embedded Network Sensor Systems (SenSys)*. ACM.

[24] Takuro Yonezawa, Hiroshi Nakahara, and Hideyuki Tokuda. 2011. Vib-Connect: A Device Collaboration Interface Using Vibration. In *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE.

[25] Takuro Yonezawa, Jin Nakazawa, and Hideyuki Tokuda. 2015. Vinteraction: Vibration-based information transfer for smart devices. In *International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. IEEE.