Pervasive and Mobile Computing 78 (2021) 101505

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

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

Sougata Sen [a], David Kotz [b]

[a] *BITS Pilani Goa campus, 17B Bypass road, Zuarinagar, Goa 403726, India*
[b] *Dartmouth College, 9 Maynard Street, Hanover, NH 03755, USA*

ARTICLE INFO

Article history:
Received 19 November 2020
Received in revised form 27 October 2021
Accepted 31 October 2021
Available online 14 November 2021

Keywords:
Smart Ring
Vibration
Security
Wearables
IoT

ABSTRACT

Many Internet of Things (IoT) devices are capable of sensing their environment, communicating with other devices, and actuating on their environment. Some of these IoT devices, herein known as "smartThings", collect meaningful information from raw data when they are *in use* and in physical contact with their user (e.g., a blood-glucose monitor); the smartThing's wireless connectivity allows it to transfer that data to its user's trusted device, such as a smartphone. However, an adversary could impersonate the user and bootstrap a communication channel with the smartThing while the smartThing is being used by an oblivious legitimate user.

To address this problem, in this paper, we investigate the use of *vibration*, generated by a smartRing, as an out-of-band communication channel to unobtrusively share a secret with a smartThing. This exchanged secret can be used to bootstrap a secure wireless channel over which the smartphone (or another trusted device) and the smartThing can communicate. We present the design, implementation, and evaluation of this system, which we call *VibeRing*. We describe the hardware and software details of the smartThing and smartRing. Through a user study we demonstrate that it is possible to share a secret with various objects quickly, accurately and securely as compared to several existing techniques. Overall, we successfully exchange a secret between a smartRing and various smartThings, at least 85.9% of the time. We show that *VibeRing* can perform this exchange at 12.5 bits/second at a bit error rate of less than 2.5%. We also show that *VibeRing* is robust to the smartThing's constituent material as well as the holding style. Finally, we demonstrate that a nearby adversary cannot decode or modify the message exchanged between the trusted devices.

1. Introduction

Internet of Things (IoT) devices are increasingly found in smart homes, connected cars, or smart healthcare. Several personal Internet of Things (IoT) devices have the capability to sense and record information, to communicate wirelessly, and (sometimes) to actuate another device – e.g., a smart remote control. These IoT devices, hereafter *smartThings*, can either be personal or shared by a group of individuals like members of a household or office space. A smartThing's user might pick it up and use it for a short period of time. Some examples of such smartThings 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 smartThing might be

E-mail addresses: sougatas@goa.bits-pilani.ac.in (S. Sen), david.f.kotz@dartmouth.edu (D. Kotz).

https://doi.org/10.1016/j.pmcj.2021.101505
1574-1192/

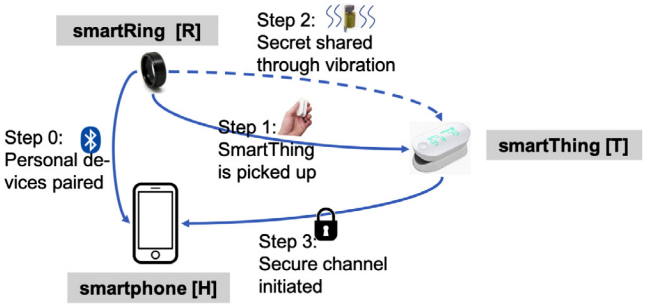


Fig. 1. Operation of key sharing using *VibeRing*: (1) individual picks up a smartThing while wearing a smartRing; (2) the smartRing discovers the smartThing and shares a secret through the vibratory channel; (3) the secret is used to bootstrap a secure wireless channel between the smartThing and a personal device.

interested in knowing who is using it. Knowledge of its user's identity allows the smartThing 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 smartThing 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 smartThing.

In this work we answer that foundational question – *how can a shared smartThing quickly, securely and unobtrusively receive a secret from an individual who is briefly interacting with it?* To answer this question, we explore using an out-of-band signal generated by a *smartRing* that is worn by the device's user. The smartRing *vibrates* to unobtrusively share a secret to the smartThing held in the same hand, and uses a pre-established secure wireless connection (Bluetooth) to share that same secret with the smartRing wearer's smartphone. This secret, now known by the smartThing and the relevant user's smartphone, can then be used to bootstrap a secure, high-bandwidth, in-band channel (e.g., over Bluetooth, Wi-Fi, or LoRaWAN) [1]. This secure channel can subsequently be used by the smartThing to learn who is using it, and for the smartphone to learn which smartThing its owner is using. We sketch the operation of the system – *VibeRing* – in Fig. 1. Although an individual can interact with smartThings in numerous ways, we focus on the physical action of *picking up and holding* a smartThing with the intention of using the smartThing in its expected usage style.

A straightforward solution to the problem is to allow an individual (or device manufacturer) to either embed a static *secret key* into these IoT devices' memory (as demonstrated by Kumar et al. [2]), or to allow a user to explicitly input a 4-digit PIN code to bootstrap a secure in-band communication channel between the smartThing and the user's smartphone or other trusted device. However, many current and future smartThings do not have traditional input capabilities, even a simple keypad or a display screen. Moreover, this manual process can be prone to shoulder-surfing attack and can be burdensome, especially when the number of smartThings an individual interacts with is large and the interaction with these smartThings is short-lived. Consider, for example, a healthcare practitioner who needs to access data from every visiting patient's CGM, or a member of a household who intends to change the TV channel. Since we are focusing on transient interactions, it is difficult to justify explicit pairing between the devices permanently. It must be noted that *we are not proposing a solution for pairing devices*, but rather are proposing a solution that allows a smartThing 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 smartThings with NFC capabilities is limited. Adding components to smartThings not only increases the device's cost, but also affects the size and form factor of the device, which in turn affects usability [3]. Moreover, for NFC-enabled devices, the user has to deliberately bring the smartThing in proximity to her smartphone, making such solutions obtrusive. Finally, recent research has revealed security vulnerabilities in these channels [4,5].

One might also consider using techniques like Simple Secure Pairing (SSP). SSP used in Bluetooth Low Energy v4.2 and above have four functional modes for pairing — Just Works, Passkey Entry, Out-Of-Band mode, and Numeric Comparison mode. Numeric Comparison was introduced in BLE v4.2, and it enables prevention of passive eavesdropping, which none of the other three modes allow. However Numeric Comparison requires the addition of hardware (e.g., display) in the IoT devices, which might not be possible to include in devices such as smart key fobs. The simplest pairing approach in SSP is *Just Works*, used when two devices exchanging pairing request/response have the OOB field and the MITM field in the LLData packet set to low. Although Just Works in LE Secure Connections (an enhanced security feature introduced in Bluetooth v4.2) performs an Elliptic Curve Diffie Hellman (ECDH) key exchange, it does not provide any form of authentication, making it susceptible to Man-in-the-middle (MITM) attacks. Alternately, one might consider the usage of biometrics or smart cards [6]. Such approaches require multimodal verification from a user and performing actions that

deviate from natural actions. Since many modern smartThings 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 Radio Frequency (RF) channels.

An important usability requirement for such a system is to ensure 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 smartThings. 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* [7,8, for example], and their constantly improving battery life, rings become an obvious candidate for the secret-sharing source.

In this work we assume that a ring is an identity proxy for the authorized individual and that the intended wearer of the ring is actually wearing the ring. Identifying the individual wearing the ring is an orthogonal problem beyond the scope of this work and can be answered using an individual's biometric information (e.g., Cornelius et al. [9]) or from the individual's physical actions (e.g., Li et al. [10]). While neither of these works focus on smartRings, we believe that techniques described in these works can be adapted to a smartRing.

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

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

- We describe *VibeRing*, a novel smartRings-based system for bootstrapping a secure communication channel between a smartThing 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 smartRing to share secrets. Through a study we show that it is possible to attain bit rates of 12.5 bits/s with bit-error rate less than 2.5%, faster and more accurately than several existing techniques. Additionally, we show that it is possible to attain a message success rate of 88% using a forward error correction technique.
- Through a user study, we show that if the smartRing and smartThing are in contact, messages can be exchanged at 12.5 bps with a bit-error rate of less than 5% for various holding styles. For various materials, it is possible to share the secret with a message success rate of 85.9%. Additionally, it is difficult for an adversary to capture the message that the smartRing and smartThing are exchanging.

The rest of the paper is organized as follows: In Section 2 we review related work that are most relevant to *VibeRing*. We describe the motivation and necessary background in Section 3. Section 4 is about the system design and operation of *VibeRing*. We describe our data collection approach and evaluation metric in Section 5, and the performance of *VibeRing* in Section 6. In Section 7, we discuss possible future steps. Finally we summarize and conclude the work in Section 8.

2. Related work

With the deployment of large number of heterogeneous IoT devices, security and privacy is a major IoT-related challenge. According to Bagchi et al. IoT applications fall under three categories: one, applications for enhancing the space in which humans live; two, empowering our devices, and three, improving the efficiency of systems that improve human lives, such as power grids [11]. While we focus on the first two categories, for completeness, we will like to mention that several researchers have explored IoT security and privacy concerns for IoT devices in areas such as industrial IoT devices [12], smart meters [13], and smart grids [14,15]. For the first two types of applications, an initial step in managing this security and privacy concern is exploring key-exchange schemes to set up a secure communication channel. In this paper, we, however, specifically focus on key exchange for personal IoT devices. This key exchange can either occur through an in-band channel (e.g., as described by Ghose et al. [16,17]) or through an out-of-band channel (e.g., as described by Kim et al. [18]). In this work, we focus on using *vibration* as an out-of-band mechanism for exchanging secret. We thus categorize works closely related to *VibeRing* as: (i) approaches that perform information exchange via vibration, and (ii) other secret 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; an accelerometer on a nearby device could receive the message [19–21]. Hwang et al. demonstrated the possibility of transferring data using vibration from a smartphone at 1 bps [20], while Yonezawa et al. improved the transfer rate to 10 bps [21]. The authors performed their experiments in a controlled setting; the phones were places on a horizontal surface while they transmitted the information. 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 [19]. However, the transfer rate dropped substantially when the phone was held in hand. Kim et al. explored use of the smartphone's

vibration motor as an out-of-band communication channel to share secrets [18]. The authors developed an implantable and wearable medical device with RF communication capability. They attained a reasonably high bitrate of 20 bps when they tested the same in an emulated environment by placing the devices between layers of meat. More recently, Lee et al. used a smartphone's vibration motor to transfer a secret to a target device, with a success rate of 92% at 22.2 bps [22]. The authors, however, performed their experiments in a controlled setting in which they clamped the smartphone to an Arduino device. The authors did not explore the physical world challenges by performing a user study. In comparison to all this prior research work, we test *VibeRing*with rigorous user studies in less-controlled settings involving a wearable device; the result is an unobtrusive method that could be suitable for everyday usage.

Researchers have also explored sending messages using a vibration motor or accelerometer on devices other than smartphones. Wang et al. used a vibrating wrist-worn device to transmit information to several accelerometer-enabled IoT devices [23], while Yonezawa et al. developed a custom IoT device that could transmit information through vibration to objects to which it was attached [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 [25]. However, the number of smartThings equipped with an accelerometer is substantially higher than those fitted with a microphone or other vibration receivers, thus making our approach more practical.

Vibration is also generated by human movement and gestures. Shen et al. showed the possibility of using an individual's natural action (two individuals shaking hands) to generate information [26]. This natural gesture's pattern provided the secret key. Several other researchers have also used movement patterns (either natural or forced) to generate secrets for smartphones [27,28].

A limitation of the vibration channel is the possibility of audio leaks [29]; researchers have proposed techniques to mitigate these limitations. Anand et al. proposed using white noise [30], or low-frequency audio tones [31] to mask the audio leak, while Kim et al. used a microphone to generate a masking sound [18]. In Section 6.3, we describe 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 [4,5]. Similarly, researchers have reported that secure-pairing techniques such as Secure Simple Pairing (SSP) in Bluetooth Low Energy (BLE) are vulnerable to MITM attack [32]. MITM attacks can be prevented by using an out-of-band channel (e.g., the visual channel, or numeric comparison) to agree upon a secret [33]. However, bringing the user into the loop in such pairing can be burdensome for the user. Other researchers investigated non-accelerometer short-range information transfer using the wearer's electromyography (EMG) signal to produce a secret [34], or used the human body to share a secret, as proposed by Roeschlin et al. [35]. 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 [36], and Nandakumar et al. proposed a system for secure transfer using acoustic signals with nearby devices [37]. They used acoustic signal because microphones are more common than NFC in mobile devices. Unlike smartphones, however, neither microphones nor NFC are commonly available in smartThings. With the increasing availability of accelerometers in smartThings, we believe that communicating through vibration will be feasible for these smartThings.

Finally, several researchers have explored the use of an in-band RF channel for secret exchange. Gollakota et al. used the Tamper Evident Announcement (TEA) mechanism for in-band secret exchange [38]. More recently, Ghose et al. proposed bootstrapping secure communication by using the in-band channel's RSSI [16]. Their protocol requires the user to hold a *Helper* device and perform a sweeping motion. The requirement of a Helper device adds overhead, and the system can be obtrusive because the user has to perform the motion. Ghose et al. also proposed using multiple IoT devices for integrity protection [17], but the requirement for multiple devices to be present during each secret-sharing session is infeasible for the sort of transient interactions we support with *VibeRing*.

3. Motivation and background

The goal of *VibeRing* is to bootstrap a secure wireless (RF) communication channel between an individual's trusted device (smartphone) and a transiently used hand-held device (smartThing) by using the individual's smartRing to share a fresh secret to each device. This bootstrapping process should involve **minimum user interaction.** This secret is required to establish a secure RF channel [39] between smartphone and smartThing, allowing them to exchange data or identity information, **even during transient interactions**. Since the smartThing might be used transiently and may be shared with others (e.g., smart hand-held weights in a gym or the remote-control unit for a conference-room projector), the situation **does not warrant permanent pairing.** Although techniques such as Just Works allows establishing an unauthenticated channel, however, they use the in-band RF channel with a longer range, allowing an adversary to eavesdrop and perform a MITM attack. With the use of the vibration channel, a short range out-of-band channel in *VibeRing*, the possibility of a MITM attack is minimized.

To understand the possibility of occurrence of (and need for) such secure transient interactions, let us consider the following scenarios where a smartThing is used transiently and yet requires a secure communication channel.

Scenario 1: In a home, household members share everyday smartThings, such as smart mugs, a thermometer, or remote controls for televisions and air conditioners. Such smartThings 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 smartThing 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 smartThing and smartphone and not obtainable by any adversary. With *VibeRing*, a household member like Jack can wear a smartRing while picking up any smartThing in the house; Jack's smartRing automatically transmits a secret to the smartThing and Jack's smartphone using an out-of-band channel, while ensuring that the secret is not captured by an adversary, Addy. Then the smartThing and smartphone can use this secret to bootstrap a secure communication channel between themselves and exchange sensitive information such as Jack's body temperature or mug usage.

Scenario 2: Jack is a nurse at a local clinic; there, he is responsible for obtaining blood-glucose data from every patient's CGM. Every patient's CGM has the capability of communicating data over RF to a hub, such as Jack's computer terminal. Before extracting the data from a patient's CGM, Jack has to ensure that the CGM is securely paired to his trusted terminal, and that the CGM transfers the blood-glucose data in ciphertext (encrypted using a secret that is known to itself and Jack's terminal) so that an adversary cannot obtain the blood-glucose information. Although Jack could manually input a PIN to pair the CGM to his terminal, this manual effort can be labor intensive. Moreover, in addition to the extra overhead of inputting the PIN into the CGM, Jack is perturbed by the possibility of an adversary observing the PIN that Jack enters.

With *VibeRing*, Jack can wear the smartRing while picking up and holding the CGM, with the intention of transferring the patient's data from it to his terminal. When Jack starts interacting with the CGM, his smartRing automatically transmits a secret to the CGM (through a vibration channel) and to the terminal (through Wi-Fi using a long-standing secure relationship). The CGM and the terminal use this secret to bootstrap a secure communication over the Wi-Fi channel.[1]

An *out-of-band channel* is a communications channel distinct from any channel used for direct data communication between the CGM and the terminal, or between the smartRing and the CGM. An out-of-band channel can be used to share a secret between two parties – here the CGM and the terminal – so they can use that secret to bootstrap a secure communication channel over a conventional RF link like Bluetooth or Wi-Fi. The use of any RF channel for sharing a secret poses the risk of an adversary eavesdropping on the communication and discovering the secret. Instead, *VibeRing* uses a *vibration channel* to share the secret between the smartThing and smartphone. It is necessary to first understand the characteristics of the vibration channel before developing an approach for using vibration as an out-of-band channel.

3.1. Possibility of communicating through vibration

We performed a preliminary analysis to determine the feasibility of vibration as an out-of-band channel. Our initial idea was to generate vibration using a wrist-worn smartwatch. To test the feasibility, one of the authors wore a smartwatch (Samsung Gear 2) naturally on his dominant hand while his finger touched an accelerometer[2] that was connected to an Arduino Uno board. Two consecutive vibration signals (one of duration 500 ms and another of duration 5 s) were transmitted from the smartwatch, with a gap of 5 s between the two signals. However, we noticed insignificant variation in the accelerometer readings. Thus, in addition to the wrist, we conducted feasibility tests by attaching the smartwatch to several other on-hand positions. Fig. 2 shows the representative on-hand position of the center of the smartwatch's bezel when it was placed on various on-hand positions.

For the finger and back of hand positions (Figs. 2a and 2b), the watch was taped firmly onto the indicated positions. The experimenter held the watch in his hand to measure the vibration generated when the watch was in contact with the palm (Fig. 2c) and the experimenter wore the watch naturally to measure the vibration from the wrist (Fig. 2d). For each position, the accelerometer's x-axis readings are shown in the bottom row of Fig. 2. The variation is similar for other axes. From the data we can see that (due to its closer proximity to the accelerometer), the signal from the vibration motor located on the finger causes the largest displacement. It is interesting to also note the substantial drop in amplitude when the watch moves $\approx$ 10 cm away from the accelerometer (from the finger to back of the hand), while there is little noticeable amplitude variation when the watch is worn on the wrist. This finding prompted us to proceed with a ring-based form-factor, rather than a smartwatch, as *VibeRing*'s vibration source.

[1] Concerns about ensuring the privacy of the data generated by IoT devices is beyond the scope of this work, and has been addressed by several researchers, including Dwivedi et al. [40]. *VibeRing* can borrow similar approaches once it has established a secure communication channel.

[2] An accelerometer sensor is capable of measuring the intensity of vibration of the signal that it receives.

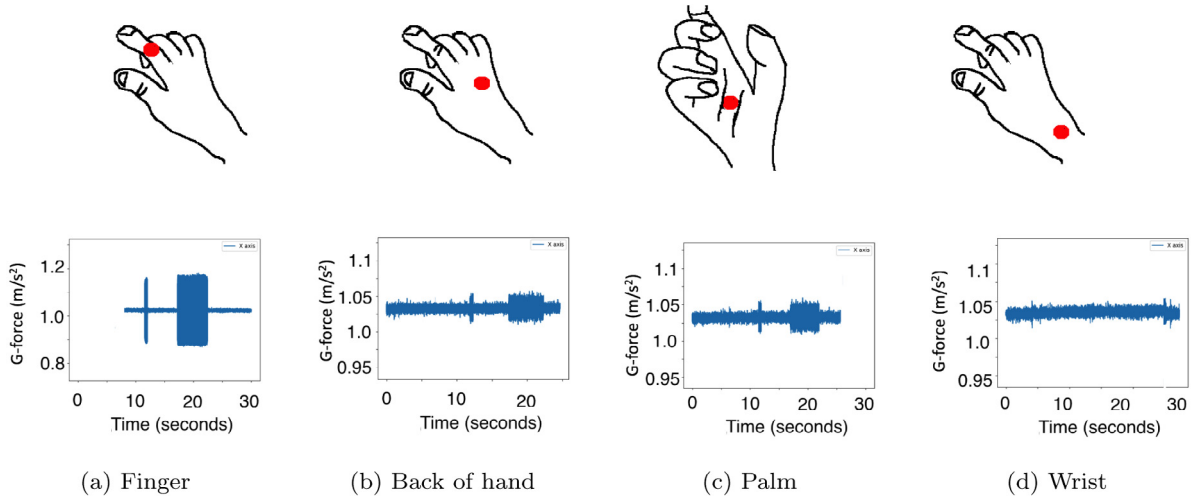


Fig. 2. Placement of the smartwatch's vibration motor (indicated by the red dot) at various on-hand positions to determine the possibility of sending a message using vibration to an object that is in contact with the tip of the index finger. Also shown are plots of accelerometer reading for various on-hand positions.

3.2. System model and assumptions

The *VibeRing* system consists of a *smartphone* (or another trusted device); a personal *smartRing* that is worn by its possessor; and a *smartThing* that may or may not belong to the individual, but is of interest to the individual. It is not necessary that the smartRing (proxy for its wearer) has had any previous interactions with the smartThing. However, prior interactions will not affect the system. All these devices have BLE (or other RF capability) for in-band data communication. All these devices are assumed to be capable of encryption standards sufficient to provide confidentiality, integrity, and authenticity of messages sent over the RF channel. The smartRing has a vibration motor, and the smartThing has an accelerometer, used together for unidirectional communication to share a secret key from the smartRing to the smartThing. Additionally, the smartRing 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 smartThing over the RF channel. Once the secure session is established, the smartThing and smartphone can exchange any information so that the smartphone can learn *what* the smartThing is, and the smartThing can learn *who* is using it.

Our design rests on several assumptions. We assume the owners of the smartThing and the smartRing 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 smartThing is not in physical contact with another object (except the authentic smartRing) while receiving the key from the smartRing. We assume that there exists a permanent pairing between the smartRing and the smartphone, both of which are personal devices belonging to the same user; this relationship provide them a secure communication channel. This communication channel is used by the smartRing to inform the smartphone about the secret. By sharing a fresh secret with the smartphone and with the smartThing, the smartRing enables the smartThing to contact the smartphone and establish a secure RF session, and to exchange information.

3.3. Adversary and threat model

In the scenarios involving Jack, an adversary Addy is aware that Jack's smartRing and a smartThing will share a secret so that the smartThing can communicate with Jack's smartphone. Addy may try to obtain this secret by impersonating either the smartThing, the smartRing, 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 smartThing or the smartRing; Addy can be in close proximity. (If Addy *were* physically touching the smartThing, we expect Jack would notice; thus, it is reasonable to assume that Addy cannot physically touch the smartThing while it is being used.) We assume that other types of attacks, e.g., gaining unauthorized access to the smartRing, or DDoS attacks, are beyond the scope of this work. Some such attacks can be mitigated by introducing hardware like the Secure Element, as proposed by Durand et al. [41]. The potential for Addy to forcibly gain physical access to the smartRing and pair with the smartRing, is beyond the scope of this work.

If Addy can impersonate the smartThing, then Addy can obtain the secret transmitted by the smartRing and can decrypt all communication between the devices. If Addy can inject a secret so the smartThing believes it comes from the smartRing,

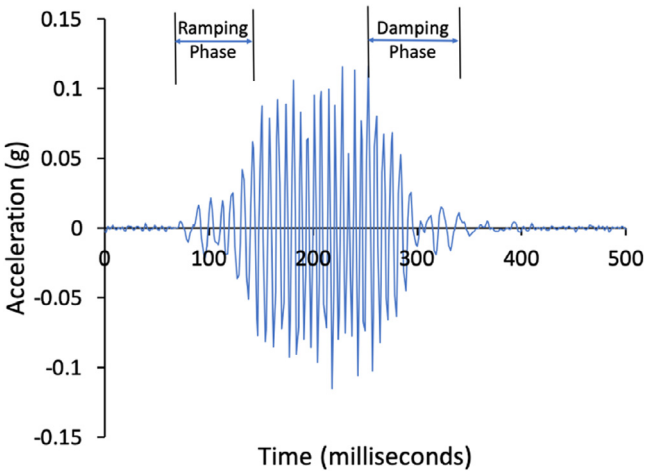


Fig. 3. Characteristics of the captured vibration.

then Addy can connect with the smartThing and the smartThing will send all its information to Addy. Thus *VibeRing* has three **security goals**: (i) data from the smartRing can be decoded only by the smartThing, (ii) data from the smartThing can only be decoded only by the smartphone, and (iii) the smartThing can verify that the message it receives via vibration is from the smartRing, and not from Addy.

3.4. Characteristics of the vibration motor

We explored the ON–OFF Keying (OOK) approach for transmitting data through vibration. An ON state indicated a bit value '1' and an OFF state indicated a bit value '0'. Once we identified that the finger was the best position to transfer information to a smartThing through touch, we next investigated motors or devices that could transmit this vibration. One of the more popular vibration motors for haptic feedback is the Eccentric Rotating Mass (ERM) motor.[3] ERM motors are available in various shapes and sizes and support numerous use cases, like in mobile phones for providing haptic feedback. A coin type ERM is usually smaller in size as compared to the other form-factors and would be more apt for the finger position. In general, every ERM motor has an eccentric mass (i.e., a mass on the shaft that is non-symmetrically distributed) that is connected to a motor. When the motor is connected to a DC source, it starts rotating and the nonuniform mass causes the haptic feedback. Ideally, the motor should start rotating as soon as it is connected to a power source. However, since the motor consists of mechanical parts, the motor starts rotating slowly and gradually reaches the normal operating velocity. This phase between the start of supply and the motor attaining a constant velocity is known as the *ramping phase*. Similarly, once the power source has been off, the motor does not stop rotating immediately, but gradually slows down until it reaches a standstill. This phase is known as the *damping phase*. Fig. 3 pictorially depicts the two phases. We anecdotally observed that the ramping phase and the damping phase were prolonged when the motor started from a state of complete rest, or when it was continuously ON for a few seconds. This prolonged ramping and damping phase could cause a bit decoding discrepancy during the two phases. Thus, we needed a scheme that was not affected by the prolonged ramping and damping phases. We provide details of our chosen scheme in Section 4.1.

4. *VibeRing*: Design and working

Now that we have established both the system and the threat model, and have explored the characteristics of the vibration motor, we next describe the working of *VibeRing*. Fig. 4 pictorially describes the working of the *VibeRing* system. The smartThing's *pick detection* module keeps its components in a low-power mode until it is *picked up*. When picked up, the smartThing's accelerometer is sampled at a higher frequency and the BLE module starts advertising its presence. Simultaneously, the smartRing'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 smartRing performs a BLE scan to listen for presence of smartThings. On receiving an advertisement from a smartThing, the smartRing generates a short random n-bit key K and transmits it as a message in the form of vibration; Section 4.1 provides details. The smartRing also shares the key K with the smartphone using a secure RF communication channel. When the smartThing's accelerometer detects the expected preamble, it processes the subsequent n bits to extract the

[3] Details about the make and working of the ERM motor can be found at https://www.precisionmicrodrives.com/vibration-motors/eccentric-rotating-mass-vibration-motors-erms/ (Last accessed 2020-11-18).

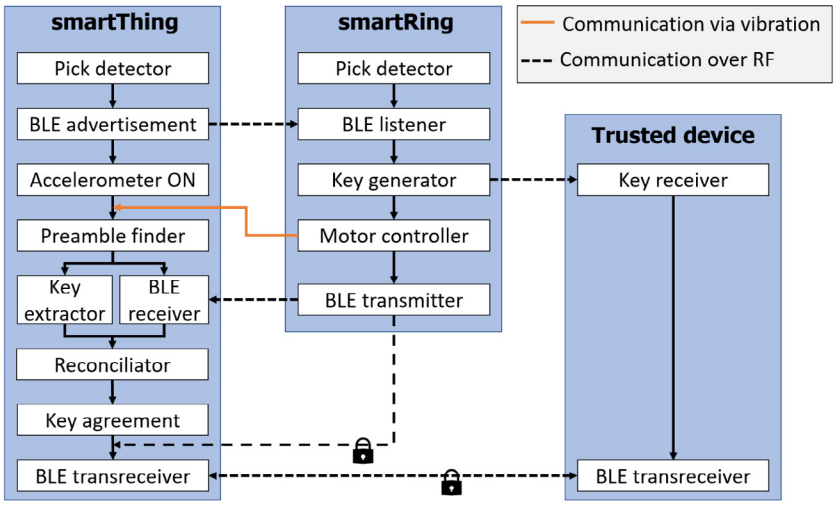


Fig. 4. Working of *VibeRing's* secret transceiver.

secret message; Section 4.2 provides details. Next, the smartThing updates its BLE advertisement to inform the smartRing that it has received the message. The smartRing and smartThing then perform a key reconciliation step, as explained in Section 4.3. This step ensures that the smartThing has the correct key and can use the message to encrypt the communication with the smartphone. Finally, the smartRing encrypts a message with details about the smartphone (e.g., IP address, MAC address) and transmits it to the smartThing over the RF channel, allowing the smartThing to directly communicate with the smartphone.

4.1. Key generation and transmission

Initialization: The key-exchange protocol begins once the smartRing's pick detection module detects a pick-up gesture and the smartRing's BLE scan discovers that a smartThing is advertising nearby.[4] Once the smartRing detects the pick-up gesture, and an advertisement from the smartThing, we assume that the smartRing is in physical contact with the smartThing.

Key Generation: Next, the smartRing generates a random key K of length n. The smartRing prefixes an 8-bit preamble to K and transmits the preamble and key in the form of vibrations. Although several key-exchange techniques transfer a PIN of only 4 decimal digits (e.g., [23]), *VibeRing* currently uses $n = 64$ bits.[5] The smartRing also shares the same key with the smartphone using a secure RF channel.

Key Encoding: The smartRing 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 (ON for the first $\frac{b}{2}$ milliseconds to transmit 1 and OFF for the 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 as explained in Section 6.3.

4.2. Key reception and extraction by the smartThing

On detecting movement, the smartThing (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 smartThing uses a band-pass filter to remove frequency components below 100 Hz and above 200 Hz, a range we

[4] We assume that there exists a pick-up detection module (e.g., as proposed in [42]). However, as an alternative, though more obtrusive, a small button could be embedded into the smartRing and the wearer could press the button to initiate the key-exchange.

[5] 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 [43].

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

selected empirically based on the motor's rotation speed (RPM). Next, the smartThing rectifies the filter's output and extracts the envelope of the signal using a t-term moving average. The smartThing 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 smartThing next determines the start of the message by identifying the 8-bit preamble. To identify the preamble, the smartThing 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 smartThing 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 smartThing must extract is equal to twice the number of bits ($2n$) in the secret message. The smartThing 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 smartThing 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 smartThing next extracts the five frame-level features listed in Table 1. 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 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.

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: bit

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

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

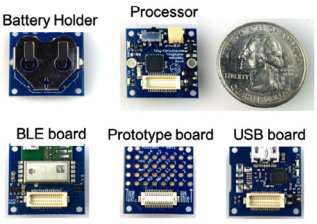
$$\text{bit} = \begin{cases} 0 & \text{when } (f_i = f_{i+1} \wedge f_{i-1} = 0 \wedge f_{i+2} = 1) \\ 1 & \text{when } (f_i = f_{i+1} \wedge f_{i-1} = 1 \wedge f_{i+2} = 0) \end{cases}$$

In case the smartThing 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 smartThing has a key (K') that it received from the vibration channel.

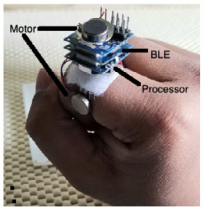
4.3. Reconciliation

Ideally, K' is identical to the key K transmitted by the smartRing. 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 smartThing knows whether it has obtained the correct key, the smartRing transmits an encrypted version of the key to the smartThing 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.

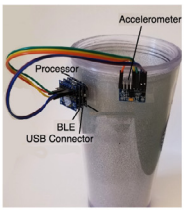
Upon receiving K' via vibration channel, and E via BLE, the smartThing decrypts the message using the decrypting function $D = g(E, K')$. If $D = K'$, then the smartThing has received the key correctly, i.e., $K' = K$. However, if $D \neq K'$, the key was corrupted during transmission. If the number of disputed bits detected is small (typically ≤ 3 in our experiments), the smartThing 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 ith combination of disputed bit values. However, if the smartThing still fails to decrypt E correctly, or if the number of disputed bits is large, then the smartThing uses the RF channel to notify the smartRing to send an Error Correction Code (ECC) via vibration, and it uses that ECC to correct more bits.



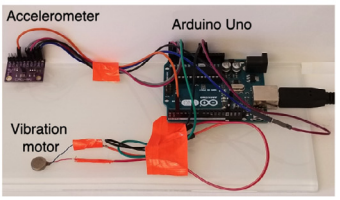
(a) Components of the prototypes.



(b) The prototype smartRing.



(c) The prototype smartThing.



(d) The prototype adversary connected to a laptop.

Fig. 5. Components and prototype of the smartRings, smartThing and the adversary made using off-the-shelf boards for evaluation purposes; they would be smaller and suitably sized if engineered as a product.

Error Correction Code: We analyzed various ECC techniques and decided to use the Hamming(m, n) code, where n is the length of the message and m is the length of the message with parity bits added, i.e., $m = n + r$. 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 bit.

5. Methods

Fig. 5 presents the prototype devices. The current prototypes are only for experimental and evaluation purposes, built using development boards (shown in Fig. 5a) from TinyCircuits [44].

Fig. 5b presents a prototype smartRing, which consists of two coin-type 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^2C 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. Although the prototype is bulky, we are confident that similar characteristics could be engineered into a comfortable and stylish smartRing; see, for example, the smart rings from Oura [8] and others.[6]

Fig. 5c presents a prototype smartThing, which uses the same processor and BLE chip as the smartRing. 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 smartThing. The smartThing can lower the accelerometer's sampling rate when it is not picked up and used. These capabilities are common (or feasible) in nearly any hand-held smart device.

The *adversary* (shown in Fig. 5d) is an Arduino Uno board connected to an accelerometer (MPU-6500 sampling at 1000 Hz) 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. We conservatively expect the adversary to have more capability than either the smartRing or the smartThing.

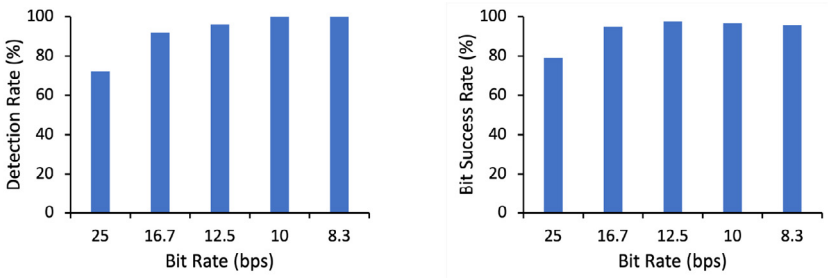
5.1. Dataset

We attached the smartThing 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 smartThings made of such materials. We collected data from a controlled study where we taped the smartRing to a smartThing and collected data, and a user study where we recruited participants who wore the smartRing prototype and picked up the smartThings.

Controlled study (CS): We performed this study to determine the feasibility of the *VibeRing* system. We attached the smartRing directly to the PT, 4 cm below the smartThing's accelerometer. The smartRing 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.

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), through word-of-mouth, or paper flyers displayed across the university campus. Most participants in the study were students in our university. The participants were aged between 18 and 30 years. 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 smartRing 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.

6 https://www.wareable.com/fashion/best-smart-rings-1340, visited 2020-11-19.



(a) PDR (b) BSR

Fig. 6. The variation of *PDR* and *BSR* at various *BR*.

5.2. Evaluation 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. A lower *BER* indicates that the smartThing interpreted more bits correctly. *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 smartThing 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?

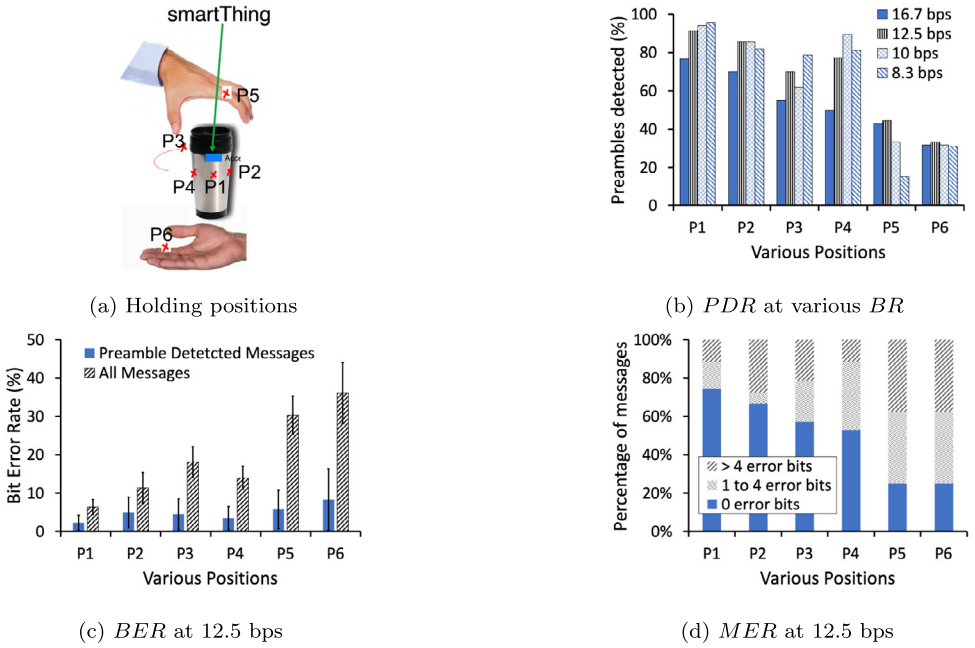
6.1. Inferring the transmitted messages

We analyze the 'CS' dataset to determine *PDR*. From Fig. 6a we see that at 16.7 bps, the smartThing 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 s.

Fig. 6b shows the *BSR* for messages whose preamble were detected. 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. Overall, 76% of messages had 3 or fewer disputed bits, which the smartThing could correct using the dispute-resolution approach. The effective *MER* at 12.5 bps (after modification of disputed bits and applying error correction) was 12%, indicating that the smartRing could transmit a 64-bit secret to the smartThing in less than 6 s 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. [45], but requires no effort from the user.

6.2. Parameters affecting message detection

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



(a) Holding positions

(b) PDR at various BR

(c) BER at 12.5 bps

(d) MER at 12.5 bps

Fig. 7. Effect of holding position on the *BER* and *MER*.

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 Fig. 7a. For each position, three messages were transferred at all five bitrates. We empirically observed that for positions $P1$ to $P4$, usually the user's palm and fingers wrapped around the tumbler. For $P5$, only the finger was in contact with the tumbler, while for $P6$, only the palm was in contact with the tumbler's base. Fig. 7b portrays *PDR* at various *BR*. Overall, *PDR* deteriorates as distance from the accelerometer increases. On average, for position $P1$, the smartThing 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 smartThings in the future.

Fig. 7c 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 smartThing 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. We also observe that for $P1$ to $P4$, the *BER* was less than 5% for messages whose preamble was detected.

Fig. 7d 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 smartRing and the smartThing ($P5$ and $P6$), 36.1% of messages had 4 or more error bits. For $P1$, the system achieved a $MSR = 88\%$ after error correction. This result indicates that to achieve high *MSR*, the smartRing and the smartThings 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 smartRing can receive the message.

6.2.2. Effect of different materials

In 'US', participants picked up each of the six smartThings described in Section 5.1. During every pick-up, the smartRing transmitted two messages at all specified bitrates.

The line plot in Fig. 8 presents the *BER* for the different material types when $BR = 12.5$ bps. The smartThing 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 smartThing 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 smartRing during the pick up gesture, thus further advocating the need for proper contact between the smartRing and the smartThing.

Fig. 8 also presents the *MER* as a stacked bar plot. For all three tumblers, where the smartRing was in contact with the smartThing, we observed that at least 70% of the messages were received with no error bits. In fact, the GT extracted 78.2%

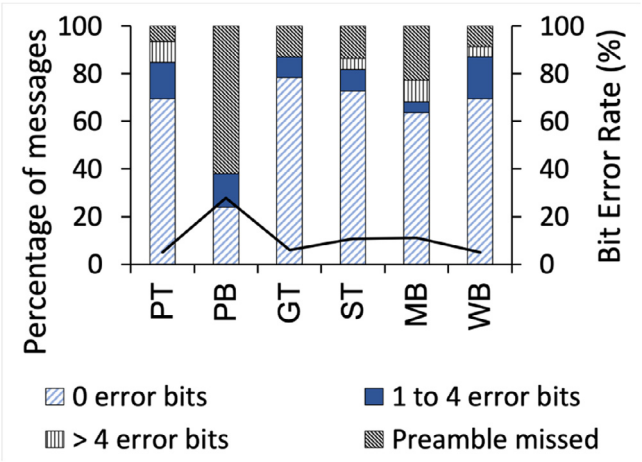


Fig. 8. Variation of BER and MER for Plastic Tumbler (PT), Plastic Box (PB), Glass Tumbler (GT), Steel Tumbler (ST), Metal Block (MB), Wooden Box (WB).

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 smartRing is in contact with the smartThing, 70% of the message exchange can occur unobtrusively in less than 6 s, 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 smartRing can be decoded only by the smartThing, (ii) data from the smartThing can only be decoded only by the smartphone, and (iii) the smartThing can verify that the message it receives via vibration is from the smartRing, and not from Addy the adversary.

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 smartThing (a) while the smartThing rested on a glass tile, and Addy also placed an accelerometer on the same glass tile, 20 cm away from the base of the smartThing, 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 smartThing was not in contact with the tile (case (b)), Addy could not infer any transmitted bit.

(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 [31]. 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 smartRing (Addy sends a key to the smartThing and creates a secure channel between itself and the smartThing), 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 smartThing and the smartRing, Addy transmitted a random message at the same bitrate and vibration intensity as the smartRing's transmission. We found that due to the difference in the intensity of reception, the smartThing 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 smartRing can be decoded only by the smartThing, (ii) data from the smartThing can only be decoded only by the smartphone, and (iii) the smartThing could verify that the data they receive is not from Addy. Our experiments demonstrated that Addy could not collect the secret shared between the smartRing and the smartThing. Without the secret message, Addy cannot decode the encrypted messages exchanges between the smartRing, smartThing and smartphone, thus addressing security goals (i) and (ii). For security goal (iii), we showed that the smartThing ignored Addy's messages.

7. Discussion and future work

In this paper, we describe the design and performance of *VibeRing*. However, several aspects still require further investigation.

Energy Drain: Energy drain of the smartRing 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 h battery life when a 3 V, 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 smartRings, the Oura ring [8], performs complex behavior-monitoring tasks, yet its battery lasts for days. Our vision for smartRings and smartThings include a wake-up circuit (e.g., as used in [46]) to ensure they selectively 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 smartRings. We conducted a small study in which we attached a motor (representing smartRing) 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 that is known not to dampen vibration signals substantially. However, in a free-living setting, the vibration will be further attenuated while traversing multiple mediums. Indeed, Kim et al. performed a similar experiment with the vibration motor places on a medium that highly dampens the vibration and they observed that the key exchange could not succeed beyond a distance of 10 cm [18]. In fact, in their experiment, the drop was more than 50%, for a distance change of 10 cm, which is much shorter than what we observed.

An adversary might also eavesdrop using the visual channel [47]. Prior work has suggested techniques to mitigate such a leakage source – e.g., adding a pseudo-random vibration message [19]. *VibeRing* can use such existing techniques.

Usability: A major concern with vibration-based systems is their impact on usability. Although we have currently not tested the system on individuals in free-living conditions, at the end of the user study, we asked the participants to comment if they felt uncomfortable using the system. Most participants commented that they felt that the vibration was similar to a smartwatch's vibration and they felt that it was not disturbing. Because it is well known that participants usually comment positively to researchers in user studies, in the future we plan to run a study with a treatment and control group to understand the usability of *VibeRing*.

System performance: Currently, we have tested *VibeRing* in a semi-controlled environment. An individual's behavior and actions in free-living conditions will be different from what he/she exhibits in the lab setting. There can be several objects that individuals interact with (sometimes, possibly with multiple objects at the same time) and the duration of interaction might vary. To understand the possibility of realizing *VibeRing* in free-living conditions, in future, we plan to carry out a user study where we will attach the smartThing prototype to several objects in the participant's house. Such a deployment will also allow us to determine whether multiple smartThings in the environment affects the system's performance. Although we do not anticipate a deterioration in performance because of multiple smartThings in the environment, however, a free-living study will allow us to understand any free-living challenges that might exist.

8. Conclusion

In this paper, we show how *VibeRing* can bootstrap a secure wireless (RF) communication channel between an individual's trusted device (smartphone) and a transiently used hand-held device (smartThing) by using the individual's smartRing to share a fresh secret to each device. This process involves no user interaction other than wearing the smartRing and holding the smartThing in the same hand. This secret helps the smartphone and smartThing to establish a secure RF channel, allowing them to exchange data or identity information, even during transient interactions. Our method does not require the smartThing and personal device to have encountered each other previously, nor does it

require the establishment of a long-term relationship (such as pairing). We demonstrate that the smartRing, because of its on-body position, can transfer this secret to a hand-held smartThing without affecting the user's natural interactions, thus ensuring unobtrusiveness.

We provide details of our prototype software and hardware, and demonstrate the system's effectiveness by conducting a user study, in which we show that *VibeRing* can share the secret reliably – with message success rate of 85.9%. Additionally, this communication is robust to the way a person holds the smartThings, and the smartThing's constituent material. *VibeRing* will enable secure communication in smartRings of the future without disturbing natural human actions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research results from a research program at the Institute for Security, Technology, and Society at Dartmouth College, supported by the National Science Foundation, USA under award numbers CNS-1329686 and 1955805. 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] A.B. Rabiah, K.K. Ramakrishnan, E. Liri, K. Kar, A lightweight authentication and key exchange protocol for IoT, in: Workshop on Decentralized IoT Security and Standards (DISS), 2018, pp. 1–6, http://dx.doi.org/10.14722/diss.2018.23004.

[2] D. Kumar, H.S. Grover, Adarsh, A secure authentication protocol for wearable devices environment using ECC, J. Inf. Secur. Appl. 47 (2019) 8–15, http://dx.doi.org/10.1016/j.jisa.2019.03.008.

[3] C. Buenaflor, H.-C. Kim, Six human factors to acceptability of wearable computers, Int. J. Multimedia Ubiquitous Eng. 8 (3) (2013) 103–114, URL https://www.earticle.net/Article/A202284.

[4] N. Akinyokun, V. Teague, Security and privacy implications of NFC-enabled contactless payment systems, in: International Conference Proceeding Series, Vol. Part F1305, ACM, 2017, pp. 1–10.

[5] W. Berchtold, P. Lieb, M. Steinebach, Secure communication protocol for a low-bandwidth audio channel, in: European Signal Processing Conference (EUSIPCO), IEEE, 2017, pp. 2206–2210.

[6] S. Banerjee, V. Odelu, A.K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, K.K.R. Choo, A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment, IEEE Internet Things J. 6 (5) (2019) 8739–8752, http://dx.doi.org/10.1109/JIOT.2019.2923373.

[7] Motiv ring, 2020, -, http://mymotiv.com/. Last Accessed: 11-19-2020.

[8] Oura, Oura ring, 2020, https://ouraring.com/ Last Accessed: 11-19-2020.

[9] C. Cornelius, R. Peterson, J. Skinner, R. Halter, D. Kotz, A wearable system that knows who wears it, in: International Conference on Mobile Systems, Applications, and Services (MobiSys), 2014, pp. 55–67.

[10] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, M. Gruteser, Whose move is it anyway? Authenticating smart wearable devices using unique head movement patterns, in: IEEE International Conference on Pervasive Computing and Communications (PerCom), 2016, pp. 1–9.

[11] S. Bagchi, T.F. Abdelzaher, R. Govindan, P. Shenoy, A. Atrey, P. Ghosh, R. Xu, New frontiers in IoT: Networking, systems, reliability, and security challenges, IEEE Internet Things J. 7 (12) (2020) 11330–11346, http://dx.doi.org/10.1109/JIOT.2020.3007690.

[12] G. Drosatos, K. Rantos, D. Karampatzakis, T. Lagkas, P. Sarigiannidis, Privacy-preserving solutions in the Industrial Internet of Things, in: Proceedings - 16th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 219–226, http://dx.doi.org/10.1109/DCOSS49796.2020.00044.

[13] D. Abbasinezhad-Mood, M. Nikooghadam, Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended chebyshev chaotic maps, IEEE Trans. Ind. Inf. 14 (11) (2018) 4815–4828, http://dx.doi.org/10.1109/TII.2018.2806974.

[14] A. Triantafyllou, J.A.P. Jimenez, A.D.R. Torres, T. Lagkas, K. Rantos, P. Sarigiannidis, The challenges of privacy and access control as key perspectives for the future electric smart grid, IEEE Open J. Commun. Soc. 1 (2020) 1934–1960, http://dx.doi.org/10.1109/ojcoms.2020.3037517.

[15] R. Chaudhary, G.S. Aujla, N. Kumar, A.K. Das, N. Saxena, J.J. Rodrigues, LaCSys: Lattice-based cryptosystem for secure communication in smart grid environment, in: IEEE International Conference on Communications, Vol. 2018-May, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1–6, http://dx.doi.org/10.1109/ICC.2018.8422406.

[16] N. Ghose, L. Lazos, M. Li, In-band secret-free pairing for COTS wireless devices, IEEE Trans. Mob. Comput. (2020) 1, http://dx.doi.org/10.1109/tmc.2020.3015010.

[17] N. Ghose, L. Lazos, M. Li, Secure device bootstrapping without secrets resistant to signal manipulation attacks, in: Proceedings - IEEE Symposium on Security and Privacy, Vol. 2018-May, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 819–835, http://dx.doi.org/10.1109/SP.2018.00055.

[18] Y. Kim, W.S. Lee, V. Raghunathan, N.K. Jha, A. Raghunathan, Vibration-based secure side channel for medical devices, in: Design Automation Conference (DAC), 2015, pp. 1–6.

[19] N. Roy, M. Gowda, R.R. Choudhury, Ripple: Communicating through physical vibration, in: Symposium on Networked Systems Design and Implementation, in: NSDI, USENIX, 2015, pp. 265–278.

[20] I. Hwang, J. Cho, S. Oh, Privacy-aware communication for smartphones using vibration, in: Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012, pp. 447–452.

[21] T. Yonezawa, J. Nakazawa, H. Tokuda, Vinteraction: Vibration-based information transfer for smart devices, in: International Conference on Mobile Computing and Ubiquitous Networking, in: ICMU, IEEE, 2015, pp. 155–160.

[22] K. Lee, V. Raghunathan, A. Raghunathan, Y. Kim, SYNCVIBE: Fast and secure device pairing through physical vibration on commodity smartphones, in: International Conference on Computer Design (ICCD), 2018, pp. 234–241.

[23] W. Wang, L. Yang, Q. Zhang, Touch-and-guard: secure pairing through hand resonance, in: International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), ACM, 2016, pp. 670–681.
[24] T. Yonezawa, H. Nakahara, H. Tokuda, Vib-connect: A device collaboration interface using vibration, in: International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), IEEE, 2011, pp. 121–125.
[25] N. Roy, R.R. Choudhury, Ripple II: faster communication through physical vibration, in: Conference on Networked Systems Design and Implementation (NSDI), USENIX, 2016, pp. 671–684.
[26] Y. Shen, F. Yang, B. Du, W. Xu, C. Luo, H. Wen, Shake-n-shack: Enabling secure data exchange between smart wearables via handshakes, in: International Conference on Pervasive Computing and Communications (PerCom), 2018, pp. 1–10.
[27] D. Bichler, G. Stromberg, M. Huemer, M. Löw, Key generation based on acceleration data of shaking processes, in: ACM Conference on Ubiquitous Computing (UbiComp), 2007, pp. 304–317, http://dx.doi.org/10.1007/978-3-540-74853-3_18.
[28] R. Mayrhofer, H. Gellersen, Shake well before use: Authentication based on accelerometer data, in: International Conference on Pervasive Computing (Pervasive), 2007, pp. 144–161.
[29] R.D. Findling, R. Mayrhofer, 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, 2015, pp. 131–135.
[30] S.A. Anand, N. Saxena, Vibreaker: Securing vibrational pairing with deliberate acoustic noise, in: Conference on Security & Privacy in Wireless and Mobile Networks (WiSec), ACM, 2016, pp. 103–108.
[31] S.A. Anand, N. Saxena, 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, 2017, pp. 173–183.
[32] D.Z. Sun, Y. Mu, W. Susilo, Man-in-the-middle attacks on Secure Simple Pairing in Bluetooth standard V5.0 and its countermeasure, Pers. Ubiquitous Comput. 22 (1) (2018) 55–67, http://dx.doi.org/10.1007/s00779-017-1081-6, URL https://link.springer.com/article/10.1007/s00779-017-1081-6.
[33] T.R. Mutchukota, S.K. Panigrahy, S.K. Jena, Man-in-the-middle attack and its countermeasure in bluetooth secure simple pairing, in: International Conference on Information Processing (ICIP), 2011, pp. 55–67.
[34] L. Yang, W. Wang, Q. Zhang, Secret from muscle: Enabling secure pairing with electromyography, in: Conference on Embedded Network Sensor Systems (SenSys), ACM, 2016, pp. 28–41.
[35] M. Roeschlin, I. Martinovic, K.B. Rasmussen, Device pairing at the touch of an electrode, in: Network and Distributed System Security Symposium (NDSS), 2018, pp. 1–15.
[36] H. Lee, T.H. Kim, J.W. Choi, S. Choi, Chirp signal-based aerial acoustic communication for smart devices, in: Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 2407–2415.
[37] R. Nandakumar, K.K. Chintalapudi, V. Padmanabhan, R. Venkatesan, Dhwani: secure peer-to-peer acoustic NFC, in: Special Interest Group on Communication (SIGCOMM), ACM, 2013, pp. 63–74.
[38] S. Gollakota, N. Ahmed, N. Zeldovich, D. Katabi, Secure in-band wireless pairing, in: USENIX Security Symposium (USENIX Security 11), 2011, pp. 1–16.
[39] I. You, S. Kwon, G. Choudhary, V. Sharma, J. Seo, An enhanced LoRaWAN security protocol for privacy preservation in IoT with a case study on a smart factory-enabled parking system, Sensors 18 (6) (2018) 1888–undefined, http://dx.doi.org/10.3390/s18061888, URL http://www.mdpi.com/1424-8220/18/6/1888.
[40] A. Dwivedi, G. Srivastava, S. Dhar, R. Singh, A decentralized privacy-preserving healthcare blockchain for IoT, Sensors 19 (2) (2019) 326, http://dx.doi.org/10.3390/s19020326, URL http://www.mdpi.com/1424-8220/19/2/326.
[41] A. Durand, P. Gremaud, J. Pasquier, U. Gerber, Trusted lightweight communication for IoT systems using hardware security, in: ACM International Conference Proceeding Series, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–4, http://dx.doi.org/10.1145/3365871.3365876, URL https://dl.acm.org/doi/10.1145/3365871.3365876.
[42] S. Sen, A. Misra, V. Subbaraju, K. Grover, M. Radhakrishnan, R.K. Balan, Y. Lee, I4S: Capturing shopper's in-store interactions, in: International Symposium on Wearable Computers, 2018, pp. 156–159.
[43] Y. Shaked, A. Wool, Cracking the bluetooth PIN, in: International Conference on Mobile Systems, Applications, and Services (MobiSys), ACM, 2005, pp. 39–50.
[44] TinyCircuits, -, 2020, https://tinycircuits.com/. Last Accessed: 11-19-2020.
[45] I. Ahmed, Y. Ye, S. Bhattacharya, N. Asokan, G. Jacucci, P. Nurmi, S. Tarkoma, Checksum gestures: Continuous gestures as an out-of-band channel for secure pairing, in: International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), ACM, 2015, pp. 391–401.
[46] S. Bi, T. Wang, N. Tobias, J. Nordrum, S. Wang, G. Halvorsen, S. Sen, R. Peterson, K. Odame, K. Caine, R. Halter, J. Sorber, D. Kotz, Auracle: Detecting eating episodes with an ear-mounted sensor, Interact. Mob. Wearable Ubiquitous Technol. 2 (3) (2018) 1–27, http://dx.doi.org/10.1145/3264902.
[47] M. Meingast, C. Geyer, S. Sastry, Geometric models of rolling-shutter cameras, ACM Trans. Graph. 33 (4) (2005).