NDNViber: Vibration-Assisted Automated Bootstrapping of IoT Devices

Sanjeev Kaushik Ramani
Florida International University
skaus004@fiu.edu

Proyash Podder
Florida International University
ppodder@fiu.edu

Alex Afanasyev
Florida International University
aa@cs.fiu.edu

***Abstract*—The rapid proliferation of sensors and their use in the modern Internet of Things (IoT) environment has led to a highly connected environment. For these inexpensive and connected devices to function efficiently, they need to communicate with each other as per the application they support. Communication with the correct entity and joining the correct network to share information are necessary operations. The action of pairing such devices securely so that they can trust the information exchange between them is termed as onboarding / trust bootstrapping. Bootstrapping is usually a highly cumbersome process, especially in resource-constrained and interface-less devices, which may not be accessible even physically after installation. In this paper, we propose NDNViber which compliments the existing bootstrapping techniques used in NDN based IoT networks. NDNViber provides NDN based networks with a dynamic, usable and secure out-of-band communication scheme using modulated vibrations to bootstrap multiple devices simultaneously and works in devices without any user interfaces. We implement a prototype that involves a commodity smartphone as the controller that can bootstrap many small IoT devices that possess accelerometer sensors. With NDNViber, we also analyze the bootstrapping of IoT devices that are inaccessible due to their physical orientation and deployment locations.**

***Index Terms*—NDN, Bootstrapping, OOB channel, IoT devices, vibration**

I. Introduction

The Internet of Things (IoT) vision conceives a connected world with seamless communication between humans and things [1]. One of the major challenges related to the use of IoT devices is the onboarding process, i.e., a process through which the devices ascertain trusted and secure communication with its peers in the network. With the use of a data-centric approach like NDN, such networks can be efficiently designed for enhanced functioning [2], [3]. However, there still is a need for non-architectural means to securely initiate the onboarding.

The initial stages of onboarding, which we will refer to as "bootstrapping", involves the transfer of necessary cryptographic information between the devices and the controller to facilitate subsequent secure communication. In other words, bootstrapping ensures that the device is talking to the "right" controller and the controller knows that it talks to the "right" device. Bootstrapping is usually a highly cumbersome process, especially in resource-constrained and interface-less devices, which may not be accessible even physically after installation. In addition, the traditional channels that utilize well-known radio frequencies (RF), bluetooth, etc. that we commonly use for transferring information between devices, are highly vulnerable and chaotic due to the presence of attackers capable of intercepting the signals and identifying the secrets being transferred. Thus, we need an alternate, out-of-band (OOB) channel for transferring this information.

The existing IoT devices and the associated research has explored multiple modalities of the OOB channels, including the use of visual aids (cameras, lasers, light pulses etc.); haptics (biometrics, gestures, etc.); acoustics (ultrasound pulses, modulated tones etc.), etc. All these methods however, can not be applied either directly without the need for deploying new hardware or are simply not feasible due to lack of resources and/or access to the embedded devices. Moreover, smart systems may include numerous IoT devices that effectively require group bootstrapping capabilities, which is not yet fully explored with the use of traditional techniques.

Considering the technologically advanced devices, in this paper, we propose NDNViber which highlights the use of a vibration based OOB channel for secure bootstrapping of resource-constrained devices. Figure 1 identifies the observed advantages of using NDNViber for bootstrapping. The characteristics seen in the prototype include: (a) real-time generation of initial secret eliminating the need for embedding private-keys or certificates at the time of device manufacturing; (b) requirement of a physical proximity of the devices and controller in the order of $0 - 1.5$ centimeters[1] thus reducing the attack surface significantly; (c) ability to bootstrap multiple devices simultaneously (with the availability of appropriate medium); (d) ability to bootstrap devices deployed in inaccessible locations that are hard to reach (e.g., behind a wall along the pipes sensing for any leaks etc.); (e) ease of use of commodity Android phones as controllers without requiring any additional hardware (i.e., via programing using the built-in vibration motors); and (f) no additional (when already built-in) or a meagre cost for accelerometers,[2] the only required component on IoT device side.

The contributions in this paper are as follows: (a) survey of existing and potential OOB approaches applicable for a smart-environment (Section II), (b) introduce the use of a vibration based dynamic bootstrapping technique that can compliment IoT applications that use NDN (Section III), (c) an encoding scheme for exchange of information over vibration

[1]Increasing distance, lowers the intensity of perceived vibrations and thus higher is the probability of erroneous reception.

[2]Typical accelerometer sensors cost less than 1 USD

978-1-7281-7440-2/20/$31.00 ©2020 IEEE

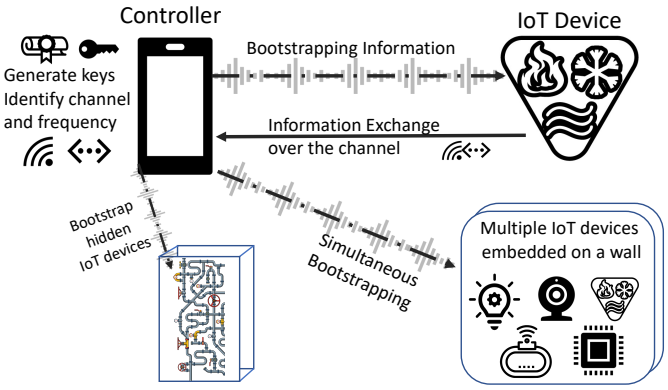

Fig. 1. NDNViber based secure IoT device bootstrapping

channel (Section IV), (d) evaluation of security properties and performance when using NDNViber (Section V).

II. Taxonomy of Out-of-Band Approaches

Out-of-band (OOB) channels use auxiliary, unused / uncommon frequencies different from that of the main frequency of communication. The main motive to use an OOB channel is for exchanging cryptographic information and thus enhance the systems' security. Figure 2 depicts a subset of the possible modalities in which OOB channels can be used in an IoT based smart environment with emphasis on securely bootstrapping these devices. In this section we survey such techniques and discuss the security properties they provide along with a note on the vulnerabilities they expose.

a) Bluetooth Low Energy (BLE): is a common choice for OOB communication. Device Provision Protocol (DPP) [4] describes an instance of its use in secure on-boarding of devices. BLE based approaches involve the use of (a) a *configurator* that broadcasts intent and (b) an *enrollee* that responds when within the communication range. Post the initial exchange, a *BLE* enabled auxiliary channel is used for exchange of cryptographic information. An important vulnerability of using BLE based pairing is the large communication range leading to a possible leak in information to eavesdropping malicious nodes. Also, the BLE protocol itself does not have a proof of possession of the bootstrapping keys in the auxiliary channel.

b) Haptics/Touch: Button Enabled Device Association (BEDA) protocol [5] describes the use of haptics technique with a reliance on physical button press patterns for bootstrapping. This is a very common approach with the patterns translating to the shared secret. As an attacker, the pattern is hard to identify unless with the use of visual aids. However, the usability is a major drawback. A legitimate user will have to identify a pattern, executing the pattern on multiple devices with the same touch sensitivity, etc. which can be cumbersome and lead to larger occurence of false negatives. The reliance on a touch enabled interface like a screen or a button makes it infeasible to use in IoT systems devoid of interfaces.

c) Magnetic field technique: Pairing devices using magnetic field values is discussed by Jin et al. [6] where smartphones were paired using their magnetometer readings. The magnetometer data, device orientation and position at that instant of time is recorded and with the addition ambient noise, a unique correlated message is generated which plays a crucial role in pairing the devices. To foil eavesdroppers, the method makes use of a proposed *Interlock* protocol which also bolsters security against passive attacks. The protocol is secure against Denial of Service (DoS) attacks owing to the extreme difficulty to overloading the magnetometer readings without an abnormally large magnet. However, it is difficult to generate stable signals by manipulating the magnetic field. An added disadvantage is that devices that possess magnetometers can detect magnetic fields but to generate and encode messages using magnetic field, there is an additional need for bulky coils and other equipment which may not work well in small sensory devices.

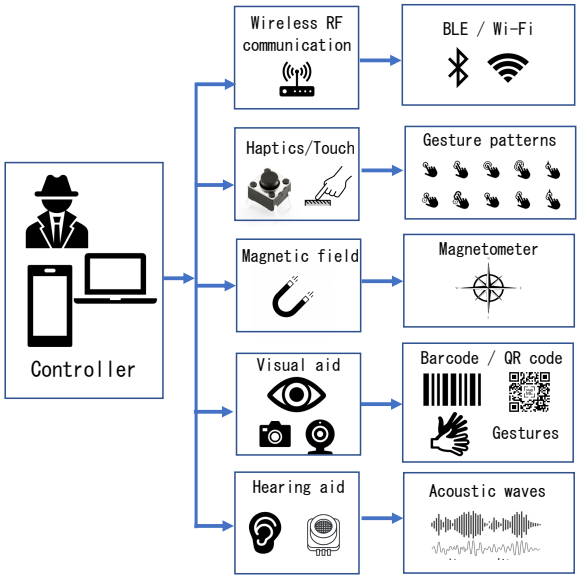

Fig. 2. Existing OOB communication methods

d) Visual techniques: An account of visible light based approach is provided by Kovacevic et al., in [7]. In order to utilize this protocol, devices must contain photodiodes and light sensors. Controllers serve as a light source and transmit cryptographic information by flashing a sequence of light pulses that other devices can decrypt to get the shared secret. The main vulnerabilities with this method of bootstrapping is that a malicious onlooker could view the flashing sequence and inject or interfere with the sequences by flashing light from a distance. Also, the need for specialized light sensors or even cameras in some cases leads to bulky upgrades to devices.

e) Audio techniques: Modulated sonic frequencies and audio patterns can also be used to perform bootstrapping. Soriente et al. [8] discuss a technique where bootstrapping information is exchanged using different codecs that generate audio that is nonsensical to humans. Audio techniques can be vulnerable to DoS attacks where the attacker can disrupt the communication using noise that interferes and modifies the encoded audio. Another commonly used acoustic technique is

the use of ultrasonic frequencies as described by Mayrhofer and Gellersen in [9]. Ultrasonic approaches however need a highly controlled environment for effective pairing and can easily be tampered with by third-party devices or physical obstacles in the vicinity. The devices also need a clear line-of-sight with each other. There is a possibility for an attacker in close proximity to the devices to be able to perform a MiTM attack, impersonate another device, eavesdrop on transmissions or even perform a DoS attack.

f) Vibration techniques: Prior attempts to use vibration as a mode of communication for security purposes can be seen in [10]–[14]. Lee et al. [15] describe an approach to enhancing the communication rate when using a vibration channel to pair devices. These articles discuss the benefits of the use of vibration for secure communication and its role in alleviating the threats other techniques. Controlled vibrations can be effectively used if the devices are very close to each other thus creating a reduced attack surface. However, there are trade-offs related to the data rate and the bit errors that can occur due to lack of synchronization. In NDNViber, using the NDN naming advantages and a specialized encoding method, we provide a solution that uses vibration based OOB for securely bootstrapping IoT devices. NDNViber observes a reduced computational load on the IoT devices and manages to reduce bit error.

III. Bootstrapping Techniques in ICN / NDN

For the following discussion, let us assume a futuristic home equipped with (a) smart climate control and (b) flow control along the gas, water, and sewer pipes, checking for potential leaks. Let us also imagine that the smart climate control system is composed not just of a smart thermostat such as Google Nest [16], but also includes hundreds of tiny temperature sensors embedded along the house walls, air ducts, windows, and doors. One of the special properties of such flow control and temperature sensors is the fact that they may not have any user interface and be completely inaccessible after the installation, yet would require bootstrapping and, potentially, re-bootstrapping (e.g., after upgrading the controller or selling the house). We assume the accelerometer associated with the constrained device to either always be active or woken up by a pilot sequence and record changes in orientation and vibrations. For the device that orchestrates and manages our smart home (controller), we will use a commodity Android smartphone.

For any of the above mentioned IoT devices to securely operate in this system, they need to [17]:

- discover which WiFi/BlueTooth/ZigBee network they should be connected to and what are the network credentials;
- be told what is *the trust anchor* of the system (a cryptographic certificate of the trusted controller) and any associated trust schemas [18];
- learn what is the *namespace* under which they can publish data; and
- obtain *a certificate* so the data created by the IoT device can be properly authenticated in the smart home network.

The initial work on NDN bootstrapping schemes either (a) assumes the existence of a pre-shared symmetric key between the device and the controller which is used to achieve initial mutual authentication [19]; or (b) expect the private key of the PKI-based approach to be embedded and installed in the device when it is being manufactured [20] and the controller scanning either a QR code or other static patterns to initiate the onboarding. While it can work perfectly in the corresponding target scenarios, these assumptions have limitations in the smart system cases we consider in this paper related to limited or no access to the bootstrapped IoT devices and the need to perform group bootstrapping.

IV. NDNViber Approach

A. Overview

The complete NDNViber bootstrapping includes four stages (Figure 3, three of which include communication over the vibration channel: *pilot sequence* (vibro), *trigger* (hybrid), *anchor* (WiFi/Bluetooth/ZigBee), and *ndncert* (hybrid) exchanges.

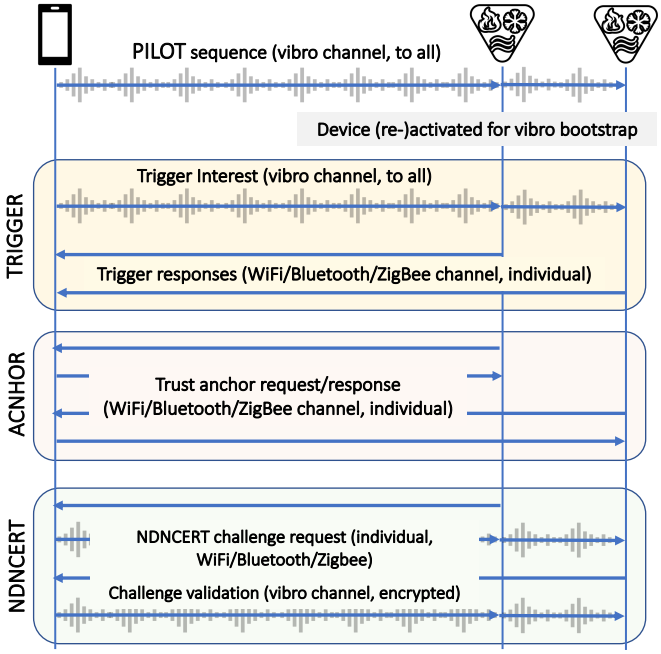

Fig. 3. NDNViber Bootstrapping Overview

The *pilot sequence* is targeting all devices within the vibro range and acts as an activation or re-activation mechanism for the IoT device bootstrapping. In other words, it simply acts as a mechanism to wake up the device to actively observe the vibration channel and attempt to decode the modulated messages.

The *trigger exchange* consists of an NDN Interest targeting all devices over vibro channel. The name of this Interest, as explained further in Section IV-B, includes the environment identity information (e.g., namespace used for the smart

house), auxiliary information (including temporary encryption key), and necessary information for the device to connect to the desired WiFi, Bluetooth, or ZigBee network along with the request for the device identifier. After decoding such a trigger Interest, the IoT devices can connect to the target network and individually respond with the unique device information, including their serial numbers, temporary encryption keys, etc.

The *anchor* exchange which follows, is realized entirely over the primary networking channel using the information and temporary encryption keys mutually obtained from the trigger exchange. The purpose of the anchor exchanges, initiated by the devices, is to obtain public key and certificate of the network, i.e., to ensure the device can successfully be authenticated for any future exchanges.

Finally, the *NDNCERT* exchange, again initiated by the individual IoT devices, runs the NDNCERT protocol [21], [22] to retrieve the assigned namespace for the device, generate the corresponding private key, and obtain the NDN certificate for this key/namespace. While most of the protocol exchanges are realized over the traditional channels (not fully shown in the illustration in Figure 3), the key security part: security challenge to ensure vibro-proximity of the device, is done using vibrations. Note that even though vibration channel will reach all devices in the range, each response is unique to the challenge-requesting device and is properly encrypted with device-specific keys.

B. Naming scheme

NDNViber directly uses NDN names (more specifically, Interest packets for the named data) as a trigger sequence to initiate (re-)bootstrapping ("`TRIGGER`"), control sequences to send system's trust anchor ("`ANCHOR`") and initiate NDNCERT vibro-challenge ("`NDNCERT`"). Following the naming convention is important for the efficient working of the protocol. The generalized naming convention we use with NDNViber follows "`/ndnViber/[sequence-type]/[house-name]/[device-name]/[params...]`". The details of the components are:

- "`/ndnViber`" is the prefix name that the device and controller use to identify the communication to be a part of NDNViber bootstrapping protocol.
- "`[sequence-type]`" identifies the sequence in progress to being either *TRIGGER, ANCHOR* or *NDNCERT*. This is very important for the exchange of appropriate information especially when we use this technique to bootstrap multiple devices simultaneously.
- "`[device-name]`" denotes the unique name for the device in the operating environment. Examples are FIU-PG6/142/duct-temp001, AA-house/washer-temp002 etc. The granularity can vary based on the number of devices deployed in the vicinity. We can thus have CPW-SR/kitchen, FIU-PG6/MeritLab/142 etc.
- "`[params...]`" here represent the other parameters including the nonces, timestamps, information regarding channels for communication etc.

C. Vibration Coding Scheme

The pilot sequence initiating the NDNViber approach includes vibrations from controller for a duration of 250 milliseconds followed by a idle state of 25 ms. It is common for IoT devices and other sensors to go into an idle state when not sensing to save power and the pilot sequence triggers their monitoring of the channel. NDNViber trigger sequence follows the pilot. The information exchange is performed using a variation of the on-off keying (OOK) technique with duration being altered instead of the amplitude[3].

TABLE I
VIBRATION DURATIONS MAPPING TABLE

Quad number	00	01	10	11
Decimal Equivalent	0	1	2	3
Vibration Duration (ms)	50	60	70	80

Our initial version of NDNViber uses a commodity smartphone running Android OS because of the possibility of controlling the vibration motors while designing applications. The information containing the "`TRIGGER`" interests or data packet responses to "`ANCHOR`" and "`NDNCERT`" interests, the controller encodes them as a sequence of time-varying vibrations using the "Vibrator" class [23][4]. The bits of each octet of the packet's wire encoding is grouped and independently converted to a vibration duration value using a simple lookup table as shown in Table I. The mapped durations start from $50ms$ because commodity android phones come with vibration motors from different vendors and thus are not precise for vibrations $\leq 50ms$. To identify the vibration durations accurately, between subsequent vibrations, we introduce an *idle period* of $20ms$. This idle period helps in (a) determining the end of a vibration and (b) providing the receiver enough time to process the received vibrations.

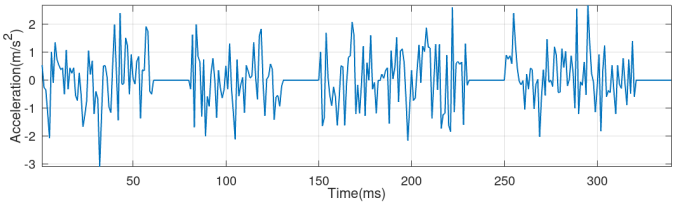

Fig. 4. NDNViber Encoded Information as detected by the accelerometer associated with the receiver

At the receiver end, the device measures the presence of an acceleration value between two idle periods and records them[5]. The recorded vibrations are then rounded off to the nearest integer value using the "round" function. The mapping table (Table I) is subsequently used to identify the corresponding equivalence of the decoded value. On completion

[3]OOK is one of the forms of amplitude-shift keying (ASK) modulation used to represent data based on the presence or absence of a carrier

[4]The vibration channel is used only for transmitting the information from the controller to the device(s)

[5]Acceleration values $< 0.1m/s^2$ is considered part of the idle period and not a part of the encoded information

of the transmission, the device now has the entire information decoded. Figure 4 depicts the vibration pattern as detected by the accelerometer associated with the device that shows the vibrations that are a part of the encoded information and the introduced idle time. The vibration durations in the figure decode to a value of "`1032...`".

The use of duration of vibrations instead of the amplitude of vibration has the following advantages:

1) All android phones can be used as a controller since the duration is programmable in all versions of android whereas the amplitude can be programmed only in phones running newer versions of android.
2) The transmission error reduces significantly. A carefully selected value for the idle period based on the sensitivity of the available controller and device can provide optimal results.
3) The computational load on the resource-constrained device decreases as processing is limited to the round-off function.
4) Time synchronization does not become too critical after the pilot sequence is identified.

V. Evaluation and Discussion

To check the feasibility of the proposed NDNViber approach, our initial experimentation was with an android phone as both the controller and the device. Our future exploration will be conducted on a thermostat like setup built using Arduino Uno, a temperature sensor, an accelerometer and a wireless interface.

A. Security Properties Analysis

The use of vibrations as a mode for communication ensures that the controller and the device are in very close physical proximity of $< 1.5cm$. Increasing this distance leads to a drop in the accuracy of reception which leads to the exchange of corrupted information. Figure 5 depicts the drop in received signal power as the controller and device move apart in the absence of a medium (like wood). The requirement of the device and controller to be in very close proximity, ensures that the probability of an attacker to intercept messages or make any attempts to tamper with the messages is very low.

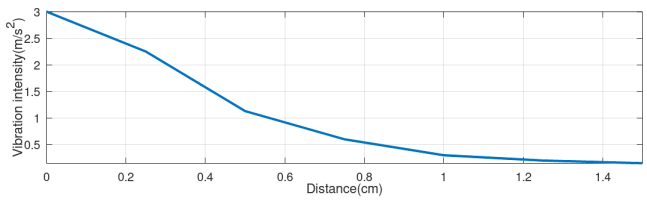

Fig. 5. Vibration intensity detected by device at varying distances from controller

An attacker can attempt to introduce random and rogue vibration signals to disrupt the system and wage a Denial of Service (DoS) attack. However, the use of specific naming conventions as described in Subsection IV-B, alleviates the impact of such rogue vibrations as they will not correspond to valid interest or data. The data centric security leveraged using NDN provides added security. The nonces and the timestamp that are exchanged with the messages also ensure the freshness and authenticity of the devices and controller while also ensuring that replay attacks are thwarted. Every time a device is to be (re-)bootstrapped, NDNViber being a dynamic approach, uses a new set of message exchanges ensuring the completed requests invalid.

Re-bootstrapping is a task that is performed infrequently in the IoT setup and for this, we assume the controller and device to have a shared secret vibro-sequence (after the device has been bootstrapped) which along with the pilot sequence will indicate to the device that the following steps are for re-bootstrapping the device. This addition will ensure that no device apart from the legitimate controller can re-bootstrap the device.

B. Performance Evaluation

The limited use of vibratory channels for communication and bootstrapping is because of the observed low throughput which is an outcome of: (a) commodity smartphones (controller in our scenario) use the vibration motors for providing user notifications and thus response times are not considered too seriously; (b) android phones have different vibration motors and thus the accuracy in terms of the duration of vibration are not precise for values $\leq 50ms$.

The target devices we consider have low computational capabilities, no interfaces, etc. NDNViber thus employs simple transformation of the data into vibration durations for encoding data. The time taken to transfer a byte using the encoding scheme described in Subsection IV-C including the 20ms idle period ranges between 260ms to 380ms.

Even though this technique is slower than the methods described in Section II, our initial experiments yielded an error ratio[6] of the order of 10^{-9}. Figure 6 depicts the vibratory signals sent by the controller (in blue) and vibrations sensed (in red) by the accelerometer in the device. The variations in the duration are because of possible addition of noise in the channel. These received vibrations are rounded off which as will eliminate the errors and lead to decoding the intended message.

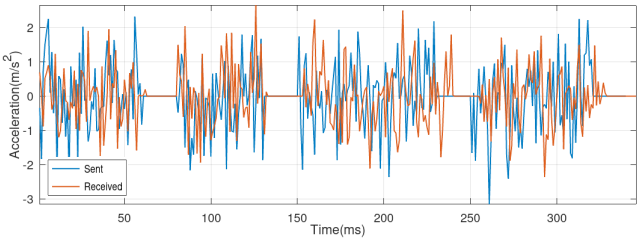

Fig. 6. Comparison of sent and received vibratory information

The time taken to pair devices with NDNViber involves the use of multiple modes of communication and thus leads to

[6]We expect the group bootstrapping to be more erroneous because of the impact the medium has on the communication and is a part of our future work

varying times for bootstrapping. Based on our "initial tests", we could pair an Android phone (Google Pixel 3a) in an average time of 8.94 seconds[7]. Accelerometers also play an important role in the overall working of this system and their sensitivity impacts the time taken for booststrapping.

VI. Future Work

An important advantage of using NDNViber is the possibility of being able to bootstrap multiple devices simultaneously. The requirement is the availability of a conducive medium that can transmit the vibrations generated by the controller to the target devices. There are inherent challenges that this brings up like (a) interference among devices; (b) induced passive vibration by the medium affecting the transmitted vibrations; (c) minor degree of acoustic leakage and attenuation; (d) orientation induced errors etc. However, the use of a controlled environment allows the controller to successfully bootstrap the devices. We intend to explore this extensively in the future.

VII. Conclusion

Vibration based channel as a possible OOB channel in an NDN based IoT network is described in this paper using the NDNViber design. The NDNViber design uses an android phone as the controller thus utilizing its features in encoding the information and transmitting them to the target device. The naming scheme and its advantages are utilized in ensuring appropriate information exchange and also plays a vital role in providing the security properties of the approach. The design showcases the manner in which the NDNViber approach is resilient to common attacks like the man-in-the-middle (MiTM), DoS, replay attacks etc. with appropriate use of features that determine the freshness. NDNViber being a dynamic approach also alleviates the issues with static approaches. The proposed encoding drastically reduces the transmission error as well. Systems with minimum compute and an associated accelerometer can use this approach to (re-)bootstrap device(s) in very close proximity.

VIII. Acknowledgements

This work was supported in part by the US National Science Foundation/Intel grant CNS 1719403.

References

[1] S. K. Ramani and S. Iyengar, "Evolution of sensors leading to smart objects and security issues in iot," in *International Symposium on Sensor Networks, Systems and Security*. Springer, 2017, pp. 125–136.

[2] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang, "Named data networking of things," in *2016 IEEE first international conference on internet-of-things design and implementation (IoTDI)*. IEEE, 2016, pp. 117–128.

[3] W. Shang, Z. Wang, A. Afanasyev, J. Burke, and L. Zhang, "Breaking out of the cloud: Local trust management and rendezvous in named data networking of things," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 2017, pp. 3–13.

[4] Wi-Fi Alliance, "Device provisioning protocol specification v1.1," 2018.

[7]At the time of writing this paper, we are attempting to optimize the performance and thus the bootstrapping time in various test devices. Extensive results will be shared by the time of publication.

[5] C. Soriente, G. Tsudik, and E. Uzun, "Beda: Button-enabled device association," 2007.

[6] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "Magpairing: Pairing smartphones in close proximity using magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2015.

[7] T. Kovačević, T. Perković, and M. Čagalj, "Flashing displays: user-friendly solution for bootstrapping secure associations between multiple constrained wireless devices," *Security and Communication Networks*, vol. 9, no. 10, pp. 1050–1071, 2016.

[8] C. Soriente, G. Tsudik, and E. Uzun, "Hapadep: human-assisted pure audio device pairing," in *International Conference on Information Security*. Springer, 2008, pp. 385–400.

[9] R. Mayrhofer and H. Gellersen, "On the security of ultrasound as out-of-band channel," in *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2007, pp. 1–6.

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

[11] Y. Kim, W. S. Lee, V. Raghunathan, N. K. Jha, and A. Raghunathan, "Vibration-based secure side channel for medical devices," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2015, pp. 1–6.

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

[13] A. De Luca, E. Von Zezschwitz, and H. Hußmann, "Vibrapass: secure authentication based on shared lies," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 913–916.

[14] R. Kainda, I. Flechais, and A. Roscoe, "Usability and security of out-of-band channels in secure device pairing protocols," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, 2009, pp. 1–12.

[15] K. Lee, V. Raghunathan, A. Raghunathan, and Y. Kim, "Syncvibe: Fast and secure device pairing through physical vibration on commodity smartphones," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 234–241.

[16] [Online]. Available: https://nest.com/

[17] H. Zhang, Y. Li, Z. Zhang, A. Afanasyev, and L. Zhang, "NDN host model," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 3, pp. 35–41, 2018.

[18] Y. Yu, A. Afanasyev, D. Clark, kc claffy, V. Jacobson, and L. Zhang, "Schematizing trust in Named Data Networking," in *Proceedings of 2nd ACM Conference on Information-Centric Networking*, Sep. 2015. [Online]. Available: http://dx.doi.org/10.1145/2810156.2810170

[19] A. Compagno, M. Conti, and R. Droms, "Onboardicng: a secure protocol for on-boarding iot devices in icn," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 2016, pp. 166–175.

[20] Y. Li, Z. Zhang, X. Wang, E. Lu, D. Zhang, and L. Zhang, "A secure sign-on protocol for smart homes over named data networking," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 62–68, 2019.

[21] Z. Zhang, Y. Yu, A. Afanasyev, and L. Zhang, "Ndn certificate management protocol (ndncert)," *NDN, Technical Report NDN-0050*, 2017.

[22] [Online]. Available: https://github.com/named-data/ndncert

[23] [Online]. Available: https://developer.android.com/reference/android/os/Vibrator