Secure, Usable and Practical Authentication for the Internet of Things

By

Kyuin Lee

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2022

Date of final oral examination: 04/25/2022

The dissertation is approved by the following members of the Final Oral Committee:

Younghyun Kim, Assistant Professor, Electrical and Computer Engineering
Suman Banerjee, Professor, Computer Sciences
Kassem Fawaz, Assistant Professor, Electrical and Computer Engineering
Rahul Chatterjee, Assistant Professor, Computer Sciences

ACKNOWLEDGMENTS

I would like to dedicate this dissertation to my beloved colleagues, friends and families who have influenced and motivated me to successfully complete my Ph.D. degree.

First and foremost, I want to thank my wonderful advisor, Professor Younghyun Kim, for his great support, guidance and effort to constantly motivate and guide me in every aspects of my life for the past five years. I am honored and I always realize how lucky I am to have such a wonderful advisor. I would also like to thank every members of my defense committee, Professor Banerjee, Professor Fawaz, and Professor Chatterjee for their valuable advice, feedback and comments to make this dissertation more concrete and solid in every ways.

I would also like to thank my beloved family members, Kwansup Lee, Soyeon Ahn, Dr. Jongho Ahn, Kwangja Yu, Dongchoon Lee, Sukhyun Shin and Kyuwon Lee. I am grateful for their positive outlook, support and unconditional love in helping me accomplish my academic goals in every ways for the past 30 years.

I also want to thank all of my colleagues, Setareh Behroozi, Jingjie Li, Tianen Chen, Di Wu, Yucheng Yang, Dr. Neil Klingensmith, Jack West, Hien Vu, Aishwarya Lekshmi Chithra, Omkar Prabhune, Victoria Schrimpf, Sujin Kim, John Rupel for their genuine characters, support and our wonderful interactions together.

Last but not least, I want to thank my dearest soulmate and friends, Songhee Hong, Dr. Juhwan Lee, Dr. Yongho Kwon, Dr. Inkyu Lee, Dr. YeiHwan Jung, Dr. Wonyup Song, Dr. Iksoo Kwon, Yooyoung Ko, Yonnho Oh, Taejin Kim, Chris Kim, Bohak Yoon, Jaesung Ahn, Teddy Ahn, Sangrok Shin, Bokyeon Hwang, and Jonghwi Park, for staying by my side and helping me get through my challenging times

with our enjoyable moments together. I could not have done it without you all. Thank you.

— Kyuin Lee

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The explosive growth in the number of connected and Internet-of-Things (IoT) devices (e.g., smart speakers, lights, and thermostats) today calls for more convenient and yet secure ways to establish wireless connection between devices. Unfortunately, current device authentication method between typical IoT devices heavily involves manual human interaction by requiring the user to type in a pin or password to establish credentials between two devices. Considering highly distributed and heterogeneous nature of today's connected environment, this unwieldy authentication process particularly degrades the overall usability of IoT systems, which often causes device users to perform poor security practices such as choosing weak passwords or even reusing them. To overcome this usability challenge that leads to various security vulnerabilities, researchers have devised zero-interaction authentication (ZIA) technique which allow devices to autonomously authenticate with each other through common secret extracted from environmental contexts to prove co-existence of devices.

In this dissertation, I present series of works on designing and building novel ZIA techniques for spontaneous authentication of IoT devices based on their deployment environments. More specifically, I first propose two techniques named SyncVibe and ivPair, leveraging readily available accelerometer to sense physical vibration in the ambient environment and authenticate closely located wearable and mobile devices in various portable scenarios. Secondly, I present two authentication techniques named VoltKey and AeroKey, designed to seamlessly and continuously associate indoor IoT devices using ubiquitously observable power line noise and ambient electromagnetic radiation as a secret to authenticate co-located devices in a fully autonomous manner. Specifically tailored towards emerging mobile and resource-constrained IoT devices, the proposed works effectively result in higher overall security and usability than

traditional authentication approaches while maintaining high practicality to be directly applicable to today's already deployed devices. In addition, to address generic challenges and limitations that exist in the current state-of-the-art ZIA works, I present a framework to automatically determine proper key reconciliation parameters that provide optimal balance between security and usability.

1 INTRODUCTION

The number of connected and Internet-of-Things (IoT) devices has been experiencing explosive growth, driven by emerging applications and advanced technologies, coupled with the active standardization of connected ecosystems. It is predicted that by the end of year 2024, there will be around 84 billion connected devices throughout the world, which is more than double the number of devices that existed in 2020 [63]. According to a former chief futurist of Cisco, there are around 127 newly introduced devices that are being connected to the internet every second [62].

With this overwhelming growth in its numbers, there are various environments in which these IoT devices are being deployed. As Figure 1.1 illustrates, the deployment environments can generally be categorized into two major sectors: *consumer and industrial*. In the consumer sector, personal, wearable, and home IoT devices, such as smart watches, smart

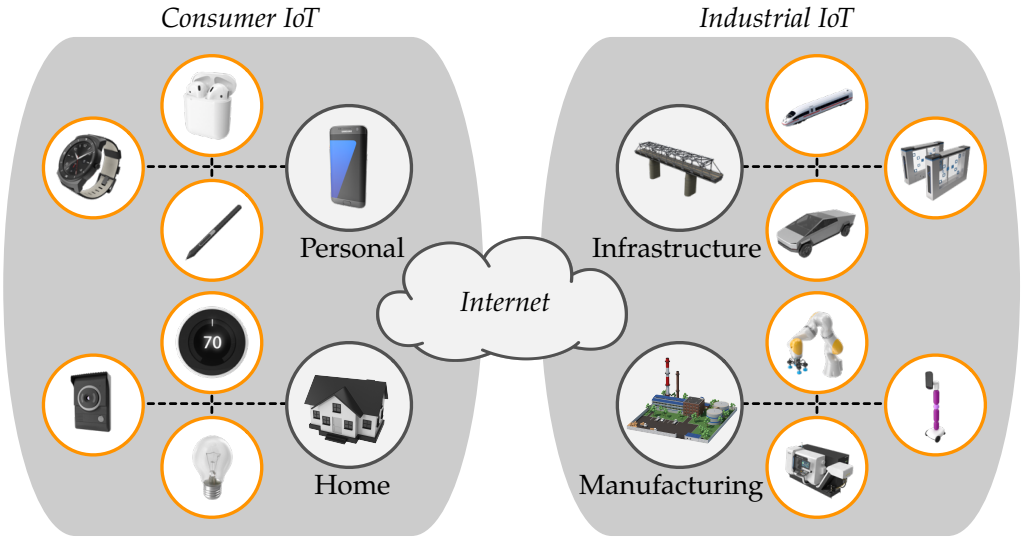


Figure 1.1: Wide deployment environments of the IoT devices are categorized into two sectors: *consumer and industrial*.

earbuds and speakers make our daily lives more efficient, effective, and healthy. On the other hand, devices deployed in the industrial sector, such as smart sensors and machines deployed in cities, warehouses, and manufacturing plants ensure higher level of automation to maintain good air quality, efficient traffic managements, and faster package deliveries. As the awareness of connected efficiency grows in the market, every industry and businesses of all sizes are thinking of ways to adopt IoT into their day-to-day activities. However, while this increased connectivity brings far too many benefits to our lives, adaptation of proliferating IoT devices leads to scalability challenges much like any other technology systems.

1.1 MOTIVATION

One of the most paramount scalability challenges that has continued to vex researchers is the question of how to quickly, securely, and seamlessly verify the authenticity between connecting devices. Unfortunately, conventional device authentication methods profoundly rely on manual human involvement. For example, Bluetooth pairing activity or connecting a wireless device to an access point in a WiFi network require human verifiers to type in a pin or password to verify that associating devices are trustworthy. Compared to traditional computing systems, this process can be considered particularly more labor-intensive and usability degrading when applied to current IoT systems for two reasons: i) Stringent constraints in the cost and the form factor have forced the manufactures to build IoT devices with limited or no usable interfaces such as touchscreens or keyboards (i.e., smart pens, smart bands). As a result, users are often forced to introduce and use secondary device, most commonly the user's smart phones, to configure devices or to establish secure connection with authorized devices. This makes traditional password- or pin-based authentication significantly more tedious, difficult, and time-consuming [15].

ii) Because device manufacturers do not consider overloading number of devices from the user's perspective, there currently exist no efficient and usable form of collective device management, and the burden of managing collection of devices falls on the user. For instance, in a typical home IoT scenario, communication network comprises of multiple edge devices connected to a single, centralized access point (WiFi router). To newly introduce or to re-authenticate devices into this network setting, users need to individually interact with different apps for different devices, which results the overall device configuration and management process to be labor-intensive and mentally overloading.

Usability is a key aspect of the authentication mechanism for IoT systems that are deployed and maintained by non-professional users. The lack of usable authentication scheme has forced many personal and mobile devices to choose usability over security and resulted in failures in properly securing critical data. For instance, without a user interface to enter a password, Bluetooth 5 devices use a common default password to encrypt the communication messages used to establish an authentication token [47, 22]. If a malicious adversary manages to gain physical access to a Bluetooth 5 headset and unpair it, they can intercept the pairing messages and extract the authentication token, ultimately gaining perpetual access to the plaintext of all subsequent communications at a distance [60]. In the case of home IoT systems, some IoT device manufacturers have inadvertently chosen usability over security and miserably failed in providing even a minimum level of security. For instance, it was reported a few years ago that 73,000 private unsecured smart cameras, including 11,000 in the U.S. alone, were being streamed on the Internet because it was not mandated to change the default password [20]. Despite the federal government's consumer advisory [29], more than 15,000 private smart cameras are still unknowingly being streamed. In the industrial sector, according to report by Palo Alto Networks' threat intelligence team, out of 1.2 million IoT

devices in thousands of physical locations across information technology and healthcare organizations in the United States, 98% of all IoT device's traffic remains unencrypted, leaving data communications vulnerable to eavesdropping by any adversary within the wireless range [13]. This vulnerability can lead to catastrophic leakage of sensitive personal data such as medical imaging, video monitoring footage, etc. Unfortunately, authenticating devices with the password does not adequately address this concern. Difficulties of authentication results in inexperienced users opting not to change default or old WiFi passwords that leads to imminent threat as disclosed in the 2016 Data Breach Investigations Report — 63% of the confirmed data breaches are attributed to using weak, default, or stolen passwords [69]. Also, in current IoT systems, once a common password is leaked, all devices using the same password must undergo tedious and error-prone password update procedures which is burdensome and sluggish. As the number of IoT devices that each user has to manage increases, combined with their heterogeneous and distributed nature, employing traditional security solutions fails to address the current problems in terms of both security and usability.

1.2 OBJECTIVES AND CONTRIBUTIONS

In this dissertation, I propose series of novel device authentication techniques towards the goal of improving usability of current IoT device authentication, so that people who have limited or no skills to operate computers can easily keep a secure connected environment. The proposed techniques successfully give users an enhanced user experience by eliminating inconveniences of conventional methods without compromising the overall security. Furthermore, the proposed techniques require minimal or no extra hardware overheads, which implies that they can easily be adopted to wide range of resource-limited devices within today's dynamic

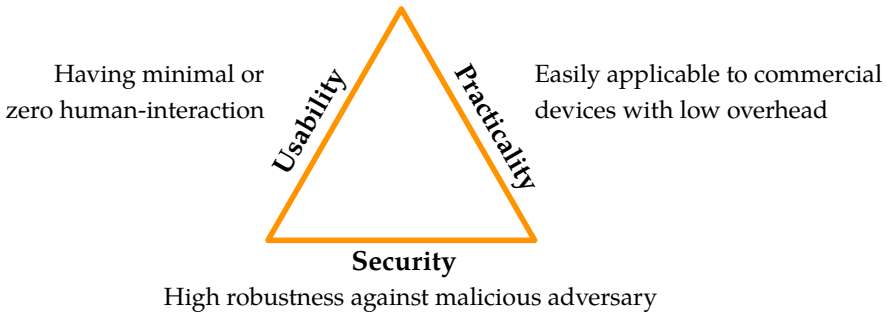


Figure 1.2: The three design aspects of the proposed device authentication techniques.

IoT deployment and usage scenarios.

Overall, the proposed authentication methods successfully improves upon conventional techniques in terms of the following three design aspects as illustrated in Figure 1.2.

1. Usability: The device authentication process can take place more quickly and efficiently with minimal or no human involvement.

2. Practicality: The authentication technique requires minimal or no hardware modifications to seamlessly be implemented into deployed devices with various form factors.

3. Security: The authentication technique provides robust protection against malicious adversaries attempting to gain unauthorized authentication.

With the three design aspects in mind, the proposed techniques are systematically evaluated in various real-world settings to demonstrate their effectiveness. Moreover, while building these series of techniques, I find that there currently exist no efficient way for the end users to balance the trade-off between usability and security of existing authentication techniques. As such, this dissertation additionally presents the generic

framework to automatically determine proper authentication parameter that balances the security and usability of device authentication schemes.

This dissertation presents different techniques designed for IoT devices that are classified into two deployment categories: *mobile* and *indoor*. Devices that are in the mobile category refers to the battery operated devices that are freely carried around by users such as smart phones, smart watches, and wearable devices. On the other hand, devices within the indoor category include all types of devices (including mobile) that are used indoors, including constantly powered devices such as smart thermostats or speakers that are designed to be mounted or not moving.

The rest of this dissertation is outlined as the following. Chapter 2 presents the background information and previous efforts of the research, as well as defining the commonly used terminologies. Next, Chapter 3 details the two device authentication techniques named SyncVibe [40] and ivPair [39] designed towards authentication between mobile and wearable devices using physical vibration that can be sensed with ubiquitously available accelerometers. In Chapter 4, I describe the techniques named VoltKey [38] and AeroKey [41] to address indoor IoT device authentication using power line noise and ambient electromagnetic radiation that are omnipresent in the indoor environment. Followed by presenting solutions to balance between security and usability of usable device authentication [37] in Chapter 5, I discuss the future works and conclude the dissertation in Chapter 6 and Chapter 7, respectively.

2 BACKGROUND

This Chapter provides the underlying background information and define the terminologies used to describe the presented works. In addition, I comprehensively provide and discuss previous efforts to improve the usability and security of device authentication through an emerging notion called zero-interaction device authentication.

2.1 DEVICE AUTHENTICATION

Device authentication is a fundamental security process where two or more devices that share no prior knowledge of each other build trust to establish a secure wireless channel to communicate with each other [76]. Traditionally, the most basic form of device authentication leverages shared secret (e.g., pre-shared key (PSK) or password), a something that only the owner "knows", to verify each others identity [21]. For IoT devices, device authentication is most commonly used to establish communication protocol such as WiFi or Bluetooth, which is the standard backbone of the wireless communication in IoT systems. For instance, in a typical smart home setting, WiFi access points and IoT end-point devices mutually authenticate each other through WiFi PSK (i.e., WiFi access point password) to agree on cryptographic keys used to encrypt and decrypt packets within the secure wireless channel [79]. Another example of device authentication which takes place between mobile devices (e.g., smart phone, smart watches) is during the Bluetooth pairing procedure. Similar to WiFi PSK, devices leverage a manually typed secret called pairing pin, typically much shorter in length, to mutually agree on the cryptographic keys for secure communication between devices.

However, secrets like PSK and passwords are usually recycled or easily gets passed around, which makes the entire network susceptible to

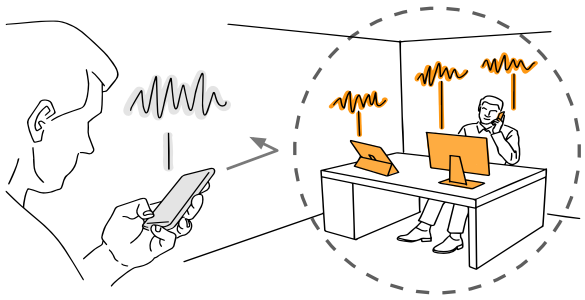


Figure 2.1: In ZIA, co-located devices are autonomously authenticated based on the ambient contextual information which can only be observed within the close region.

various forms of attacks such as man-in-the-middle (MITM) and impersonation attacks [60]. To augment security, researchers has devised two- or multi-factor authentication which requires another layer of secret such as something you are (e.g., fingerprint or iris) or something you have (e.g., one-time password generator) in addition to the password. In IoT scenario, above mentioned human-assisted device authentication remains as a big usability challenge due to lack of usable interfaces embedded in the small devices [15]. Furthermore, highly distributed nature of the smart devices today makes it impractical for users to manually manage individual devices quickly and efficiently.

2.2 ZERO-INTERACTION AUTHENTICATION (ZIA)

Recently, *Zero-interaction Authentication* (*ZIA*) has emerged as a promising solution to enable fast and convenient yet secure device authentication. It exploits spatiotemporal randomness within the device's surrounding environment, often called *contextual information* [15] to prove *co-location* of the devices. In this dissertation, contextual information refers to the device's surrounding environmental contexts, such as ambient audio, light, or radio waves, that are captured with various sensors embedded in devices,

whether it is arbitrarily generated or naturally existing within the ambient environment. The term *co-location* is defined as a physical space that are enclosed or bordered by walls such as room, house, or vehicle, depending on the ZIA's different use cases. Devices that use ZIA take advantage of the fact that the common contextual information is shared only by a limited group of co-located devices. The presence of common contextual information is evidence that the devices are located in the same place at the same time, which implies that they legitimately belong to the same user. In order to authenticate two devices, ZIA extracts *keys* (sequence of bits) from the contextual information and if the keys exhibit bit-wise equality (100% matching bits), two devices are successfully authenticated. The contextual information are excellent source of randomness for key generation because they can be easily and inexpensively be measured with sensors that are readily available on many IoT devices. Therefore, the keys generated from contextual information can be used to establish initial trust (as a pairing pin) or to protect subsequent communication (as a cryptographic key).

Compared to password-based authentication, ZIA is principally advantageous in terms of security and usability. Unlike password-based authentication where the entropy is derived from a user-provided text string, ZIA leverages the randomness that exists in the environment and requires no input from the user. Therefore, it eliminates the need for human involvement for making, entering, and managing a password, which dramatically improves the overall usability of IoT systems. It security advantage is that the time-varying nature of contextual information allows devices to use a new key for each authentication attempt or periodically update the key, which significantly reduces the attack window for adversarial agents. The Figure 2.1 illustrates the notion of ZIA. Co-located devices close to the user are autonomously authenticated without any human involvement, based on the ambient contextual information that can only

be observed within the region. The adversary's device (not located within the user's region), cannot authenticate because the contextual information is significantly different.

2.3 PREVIOUS WORKS ON ZIA

Table 2.1: Physical context and sensors used in various ZIA works.

Work	**Sensor**	**Contextual information**
Walkie-Talkie [75]	Accelerometer	User's gait acceleration
Mayrhofer et al. [49]	Accelerometer	Shaking acceleration
Groza et al. [23]	Accelerometer	Transportation vibration
Convoy [25]	Accelerometer	Vehicular vibration
Kim et al. [36]	Accelerometer	Induced vibration
Ahmed et al. [2]	Accelerometer	Gesture acceleration
Belkhouja et al. [5]	Accelerometer	Skin vibration
Li et al. [43]	Buttons, Knobs	Input timing
iBEP [77]	ADC	Body electric potential
H2B [46]	ECG	ECG interval
H2H [55]	Piezo	ECG interval
Yang et al. [78]	EMG	EMG signal
Jin et al. [31]	Antenna	Radio frequency noise
Varshavsky et al. [68]	Antenna	Wireless signal's Fluctuation
Proximate [48]	Antenna	Wireless signal's RSSI
Xi et al. [72]	Antenna	Wireless signal's CSI
Schurmann et al. [58]	Luminosity	Brightness
Miettinen et al. [50]	Microphone	Audio
Saxena et al. [57]	Camera	Visual image
Han et al. [24]	Mic., Accel.	Inter-event timing

ZIA for IoT devices has actively been studied, leveraging different

context information to establish a secure communication channel using various on-board sensors. In Table 2.1, I list previous work's sensors as well as their captured contextual information used to extract identical keys. Several prior works have leveraged accelerometer readings to authenticate between user's trusted mobile devices [8, 9, 49, 16]. For instance, to authenticate multiple mobile devices carried by the walking user, Walkie-Talkie [75, 74] proposes key generation method from walking characteristics (gait) of the user, based on the acceleration signal measured from different parts of the user's body. Additionally, Groza et al. [23] introduces secure authentication of devices within multi-modal transport (i.e., train, tram, bike, and vehicle), leveraging acceleration signal measured within various transportation modes. Similarly, Han et al. [25] proposes to use vibration measured by multiple vehicles in the same lanes (a platoon of vehicles) to authenticate a newly joining vehicle for vehicle platooning purposes.

For wearable mobile devices such as smart watches and smart bands, contextual information leveraging user's biometric properties such as ECG (heartbeat data) [46, 55, 80], EMG (produced by skeletal muscles) [78], and skin vibration [5] has been used to authenticate between low-cost wearable devices and implantable medical devices. By harvesting random keys extracted from the variations in heartbeats measured by ECG, [46] is able to authenticate medical devices only when they are in direct physical contact with the human body using low-cost piezo sensors. For implantable medical devices, Kim et al. [36] proposes to induce vibration with vibrating motor to authenticate resource constrained devices. Other solutions require the users to touch, wear, or interact with the devices to establish a common key based on the body potential [77, 31] or explicit gestures [45, 2]. These protocols require some explicit or implicit user action and do not generalize to non-wearable devices.

IoT devices that are designed to be mounted or used in stationary

position leverage readily available contexts such as audio, humidity, luminosity, and visual channels [50, 58, 57, 51, 59, 35]. Schürmann and Sigg [58] propose to use a microphone to capture audio sample to extract a secret key based on differences between energy on adjacent frequency bands. However, due to a large amount of entropy extraction in a small time interval, time synchronization between devices using commercial-off-the-shelf IoT devices has been identified as a possible drawback in their approach. Their technique also requires authenticating devices to be within audio range of one another, which is not generally the case for multiple IoT devices spread among several rooms. While ZIA scheme using longitudinal audio and luminosity data does not require exact temporal alignment of measured data, visual, luminosity, and audio channels impose an additional hardware burden (i.e., camera, microphone, and luminosity sensors) that might not be readily available in most low-cost or small devices [50]. Another line of work presents universal operation sensing [43], allowing IoT and wearable devices to authenticate through user's physical operations such as pressing a button or rotating a knob.

A similar approach is taken with RF-signals to prove co-presence of multiple devices by relying on the received strength signal indicator (RSSI) value or physical layer features of the radio environment [68, 34, 26, 30, 72]. Moreover, ProxiMate [48] utilizes any radio technology to extract a secret key based on small scale temporal variations in the perceived wireless signal. However, limitations of these works include small authentication radius (less than 1 m) that may pose severe challenges during large scale deployment scenario. Additionally, RSSI is very susceptible to malicious attacks in that it can be predictable by a distant adversary with access to trusted device's exact location [48]. While these works successfully explore secure and usable methods to autonomously authenticate devices, they require external antenna that is usually readily available in many miniature devices due to cost or form factor constraints.

While most previous methods usually rely on homogeneous sensor pairs to capture matching contexts across authenticating devices, Han et al. [24] authenticates heterogeneous devices located in secure boundary (indoor) by leveraging inter-event timing contexts from different sensors types. Although this method effectively addresses the limitations of requiring identical sensing modalities, its source of entropy still needs to be derived from human activities within the environment, which may not be observable at all times.

To further understand the security aspects of ZIA, [51] comprehensively analyzes entropy loss during error-correction schemes and the level of contextual separation in the home and office environments. Additionally, [17] provides first large-scale public dataset of various devices (i.e., mobile and stationary) in multiple environments (i.e., car and office) and re-evaluates various ZIA schemes.

3 ZIA FOR MOBILE DEVICES

Short-range wireless communication technologies such as WiFi and Bluetooth have become ubiquitous in various human-operated mobile devices, such as smart phones or smart wearable devices. One of the key attributes of communications in such devices is frequent and short-lived pairing, which is a process of mutually registering two devices with no prior knowledge. During typical secure device pairing process, a human assisted device authentication takes place (typing in pairing pin or key) to establish a wireless link between a new pair of devices. This Chapter describes two usable and secure authentication protocols to address tedious and time-consuming nature of current pairing scheme between mobile devices. The first method, SyncVibe, utilizes a vibration motor and an accelerometer, that are already ubiquitously available or easy to embed in mobile and wearable devices, to transmit and receive authentication key. By simply keeping two devices in direct contact, the user can bootstrap a secure, high-bandwidth wireless connection without going through tedious pairing procedure. The second method, ivPair, specifically addresses pain of authenticating mobile devices within intra-vehicular scenario. Using ivPair, users can authenticate a mobile device equipped with an accelerometer with the vehicle's in-vehicle infotainment system or other mobile devices by simply holding it against the vehicle's interior frame.

3.1 SYNCVIBE: FAST AND SECURE DEVICE AUTHENTICATION THROUGH PHYSICAL VIBRATION ON COMMODITY SMART PHONES

Unfortunately, even in the latest wireless standards, the lack of an intuitive and simple device authentication method significantly degrades the user

experience. For example, authenticating a new smart phone with wireless earbud often requires a sequence of steps of discovering nearby devices, selecting the target device, and entering a passkey, which may be too cumbersome to do just for spontaneous use.

Authentication methods that are commonly supported by state-of-the-art protocols and devices can generally be categorized into three types. First, a new authentication request can be accepted without any authentication, which is called "Just Works" operation in Bluetooth. This is common for devices with limited or no user interface (UI) and provides no defense against MITM attacks. Second, as in the car infotainment system example, the user may be prompted to enter a passkey generated by one device on the other device. While this method is generally secure from MITM attacks unless the attacker is able to obtain the passkey via some means, e.g., shoulder surfing, the inconvenient user intervention often thwarts the user. It also requires hardware components for implementing the UI, e.g., an LCD and a keypad, which may not be feasible for small, low-cost devices, such as Bluetooth headphones. Lastly, out-of-band (OOB) authentication utilizes a secondary channel, such as near-field communication (NFC), for exchanging authentication information. The key advantage of OOB authentication is the convenience that the user does not need to manually enter a passkey, and thus a longer authentication key can be used to enhance security. As long as the OOB channel is protected from eavesdropping and MITM attacks, it can be assumed that the wireless channel is also protected from the same kinds of attacks.

One of the promising OOB channels for secure authentication is *physical vibration* generated by a vibration motor or a piezoelectric vibrator, which can be captured using an accelerometer. As shown in Figure 3.1, authentication key required for radio frequency (RF) communication can be transferred using a physical vibration channel. Physical vibration has several unique advantages for OOB authentication: i) It is a proximity

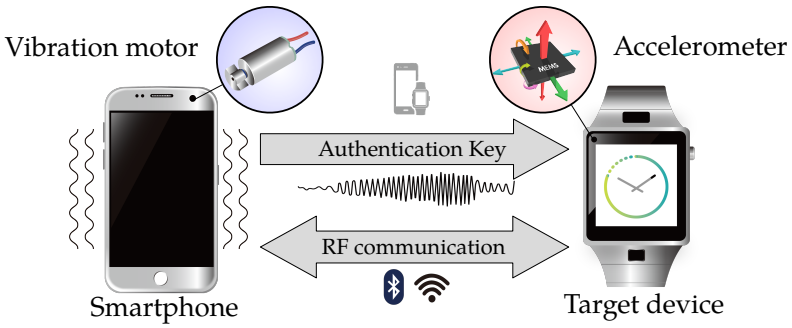


Figure 3.1: The smart phone transmits authentication key to the target device through the vibratory channel using a vibration motor to bootstrap a high-bandwidth wireless connection.

channel that requires a direct contact between the transmitter and the receiver, which makes eavesdropping and MITM attacks significantly difficult than RF channels. As demonstrated in earlier works, it can also be protected from acoustic eavesdropping attacks by generating acoustic noises to mask audio leakage from vibrating device [3, 36]. ii) Vibration motors and accelerometers are ubiquitously available in most mobile and wearable devices. iii) Finally, vibration motors and accelerometers are low-cost and small-footprint components that can be easily adopted.

This section proposes a novel scheme, named SyncVibe, for enabling accurate vibratory communication for fast, secure, and convenient device authentication on commodity smart phones. Compared to previous vibratory communication schemes, the proposed scheme significantly improves the effective throughput by maximizing bitrate and minimizing synchronization overheads. To achieve this goal, SyncVibe uses *vibration clock recovery*, which extracts timing information from the non-ideal vibration waveform of data bits by detecting the activation and deactivation of the vibration motor. Only when the data bits do not contain a bit pattern that can be used for clock recovery, a short synchronization pattern is inserted

to recover clock with a minimal overhead.

Challenges

Since all built-in vibration motors in today's smart phones are originally designed for haptic feedback and user notification, not for data communication, the Android API does not provide applications with the ability to control the amplitude and frequency of vibration in a fine granularity, nor is the motor driver circuit designed to support that. Instead, the API takes an array of integers as an input parameter that represents the vibration pattern where each value indicates durations in milliseconds to turn the vibration motor on or off, while the amplitude and the frequency of the vibration are solely determined by the physical characteristics of the vibration motor and the driver circuit.

Slow Vibration Motor Response To transfer data through this vibration channel, the data bits are encoded into a series of on-off patterns, also known as OOK scheme; turning on or off the vibration motor for a pre-defined time interval, which will be referred to as *vibration period*, represents a bit `1` and a bit `0`, respectively. Although the minimum vibration period supported by the API is 1 ms, the slow response of the motor and the driver's lack of precise timing control at low vibration periods limits the minimum time for the actual vibration period in practice, resulting in low bitrate.

Figure 3.2 shows some examples of signal envelopes of single-axis accelerometer readings as a motor being turned on for different vibration periods. Ideally, the rising slope of each signal should be completely limited to the vibration period during which the motor is activated. In other words, the starting point of the downward slope should not exceed the given vibration period to prevent next symbols from being incorrectly demodulated. However, for vibration periods of 20 ms and 50 ms, the

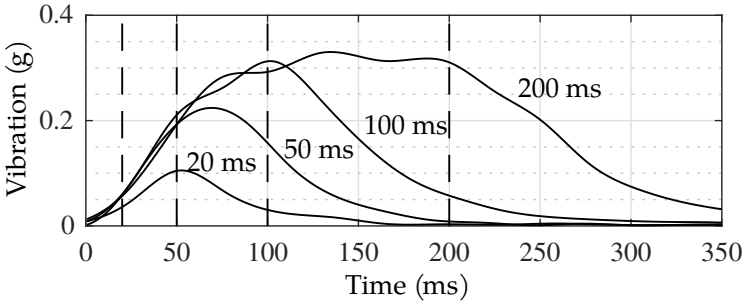


Figure 3.2: Examples of envelopes of vibration pulses for different periods: 20, 50, 100, and 200 ms. Vertical lines denote the ideal pulse widths.

signal's amplitude starts to decrease at 50 ms and 65 ms, respectively, which is far beyond the ideal end point of the pulses. As a result, this slow response of short vibration pulses hinders high-bitrate vibratory communication.

Lack of Synchronization Vibratory communication is intrinsically asynchronous communication. Having no external clock signal for synchronization, the start and the end of each bit have to be aligned between the transmitter and the receiver. For example, when each bit is encoded with 100 ms vibration period and captured by an accelerometer at 100 Hz sampling rate, each bit segment should consist of exactly ten samples. However, Android, in its current state, is not meant to be used for real-time purposes [54] and therefore cannot be guaranteed to vibrate for exact given amount of time. In addition, the slow response of the vibration motor discussed earlier can cause misalignment of bit segments, which can result in significant decoding errors.

Figure 3.3 shows a comparison of an ideally aligned vibration signal and an actual signal when 51 bits are transmitted with a vibration period of 40 ms (i.e., 25 bps). At early bit positions, bit segments, consisting of fixed accelerometer sample counts, are well aligned between the transmitter

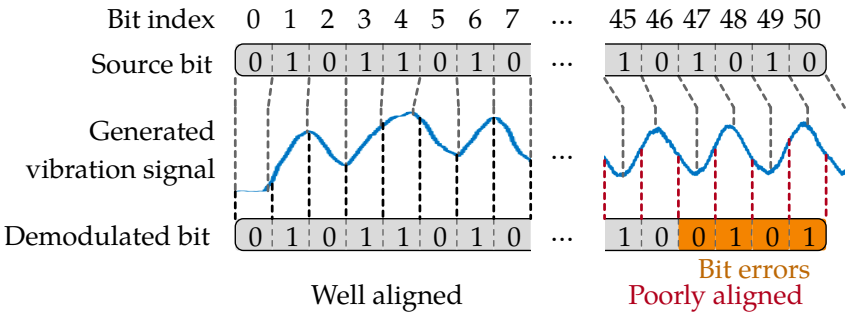


Figure 3.3: Example of bit errors due to the loss of synchronization in a long bit stream.

and the receiver, and each bit is correctly demodulated. However, as bits get demodulated further down the stream, small misalignment of segments at early stages starts to accumulate, causing later segments to decode unsynchronized samples, resulting in erroneous demodulation of bits.

3.1.1 System and Threat Models

System model: SyncVibe assumes a device authentication scenario between a smart phone, equipped with a vibration motor and a mobile or wearable device (e.g., smartband, wireless earbud), equipped with an accelerometer. The two devices does not have any prior trust established between each other. To authenticate two devices, the user must put two devices in direct contact while smart phone transmits the authentication key to the target (mobile or wearable) device through the vibration channel. In order for the devices to be successfully authenticated, the key should be transmitted with no error, so that the key can be used to establish secure wireless network (e.g., WiFi, Bluetooth) using symmetric key encryption. In terms of device's hardware capabilities, the smart phone is equipped

with either the linear resonant actuator (LRA) or eccentric rotating mass (ERM) type motor and the target device is equipped with accelerometer which is integrated at the manufacture time with zero additional hardware modifications. Additionally, two devices are not modified in terms of their kernel-level software.

Threat model: SyncVibe considers threat model which is in line with prior pin- or password-based pairing or authentication schemes. In this work, the adversary is fully aware of the SyncVibe's underlying protocol and may be located close to the legitimate authenticating devices within each other's wireless coverage. However, the adversary does not have physical or remote access to the legitimate smart phone or the target device. The adversarial device may be a device that is accidentally trying to authenticate with legitimate user's device, or it can be a malicious device that is intentionally attempting to authenticate as a stepping stone for additional threats. The adversary can eavesdrop and access any unencrypted wireless packets exchanged by the legitimate devices during SyncVibe's protocol.

3.1.2 Proposed Approach

Modulation

SyncVibe employs OOK as the main modulation scheme to enable fast vibratory communication for both ERM and LRA motors. Having no external clock, the modulation should ensure that the receiver precisely detects the start of the vibration signal, segments the vibration signals, and recovers a correct data bit from each segment, while achieving high throughput.

The first step of the modulation is modifying the data bit stream to ensure that the receiver can synchronize itself to the transmitter and cor-

rectly segment the vibration signal. As described in the next subsection, the receiver recovers the clock from a clear transition from the off state to the on state of the motor. As shown in Figure 3.2, the slope is steeper at the beginning of a pulse and becomes flatter gradually, so the moment that the motor is turned on after it is fully damped is the optimal point to recover the clock. A *synchronizable pattern*, therefore, is defined as a bit pattern of several bit `0`'s followed by bit `1` (i.e., `0...01`), where the number of bit `0`'s is set to the minimum that ensures that vibration is fully damped after turning off the motor. To prevent synchronization failure, the synchronizable pattern should appear in the vibration signal before synchronization breaks down. Otherwise, the misalignment of bit segments will accumulate, resulting in a burst of bit errors. More specifically, the synchronization pattern must be present at least once every k bits, where k is a pre-agreed synchronization interval, i.e., the maximum number of bits that the receiver can keep bit alignment without synchronization. If the data bit stream contains an unsynchronizable bit stream (i.e., a bit stream without the synchronizable pattern) longer than k bits, SyncVibe explicitly inserts a *synchronization marker* equivalent to the synchronizable pattern every k consecutive unsynchronizable bits. On the other hand, if the synchronization pattern appears in the data bit stream itself at least once every k bits, no synchronization markers are inserted.

Next, a *pilot marker* is added to the beginning of the vibration signal. A pilot marker in this scheme serves two purposes: i) allowing the receiver to measure the maximum amplitude of the vibration, the amplitude of a single vibration period, and the slope threshold values used for demodulation and ii) indicating the starting point of data transmission. The vibration signal is attenuated as it propagates from the transmitter to the receiver, and the attenuation rate varies by the medium (e.g., smart phone cover and protective case), the force applied between the transmitter and the receiver, etc. To account for the variation in the attenuation rate, com-

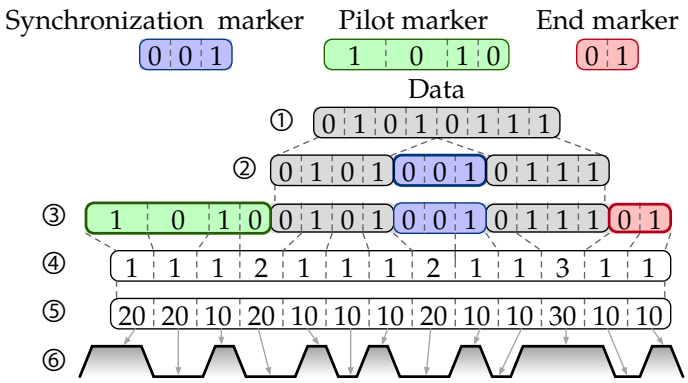


Figure 3.4: Modulation example. Synchronization marker is `001`, $k = 4$, vibration period (t) is 10 ms, and long pulse period (t_l) is 20 ms.

munication session starts by sending a maximum amplitude vibration prior to sending actual data, similar to a preamble used for automatic gain control (AGC) in RF communication. A pilot marker consists of one long and one short pulse. From the first long pulse, the maximum amplitude of vibration is measured, and the starting segment of the data bits is located. The second short pulse is generated by turning on the motor only for one vibration period. This pulse provides a reference amplitude and the increasing and decreasing slopes used for synchronization and demodulation. Finally, an *end marker*, a 2-bit pattern of `01`, is appended at the end of the bit stream to mark the end of the bit stream in case the length of the data bits is not fixed.

Figure 3.4 shows an example of the modulation process. In this example, SyncVibe assume that the synchronization marker is `001` and $k = 4$. The vibration period (t) is 10 ms. The long pulse period (t_l) for the pilot marker is 20 ms. (①) First, 8-bit data `01010111` is given. Note the synchronizable pattern `001` does not appear in the data. (②) Therefore, one synchronization marker `001` is inserted after $k = 4$ bits. (③) The pilot marker and the end marker are added in front and at the end, respectively.

(④) The number of consecutive bit `0`'s and bit `1`'s are counted. (⑤) To these numbers, t or t_l is multiplied to convert into a vibration pattern in milliseconds. (⑥) Finally, the vibration motor is turned on and off for the specified time duration to generate a vibration signal.

Demodulation with Clock Recovery

Vibration signal generated by the transmitter is measured by the receiver using an accelerometer and demodulated into data bits. First, a band-pass filter is applied to the raw accelerometer readings. This removes bias due to gravity, low-frequency noises caused by external vibration sources, such as the user's body motion, and high-frequency measurement noises. Next, an envelope detector is applied to obtain a smooth signal envelope. Figure 3.5 shows an example of a vibration signal envelope and its demodulation. The pilot marker is used for initial synchronization by locating the starting point of its first pulse. From the second pulse, SyncVibe measures the reference slopes and the reference amplitude. The reference slopes and amplitude are used for the demodulation of the data waveform that follows the pilot maker.

The data waveform is divided into segments of a fixed length equal to the vibration period. Each waveform segment is approximated as a linear function of time. The slope of the linear function is compared to the reference slopes retrieved from the pilot marker. If the slope is closer to the increasing reference slope, the segment is demodulated as a bit `1`. On the other hand, if it is closer to the decreasing reference slope, the segment is demodulated as a bit `0`. Otherwise, if it is closer to zero, the bit is demodulated as the previous bit. To account for possible changes of the attenuation of vibration due to varying pressure applied by the user, the reference slopes are continuously updated as demodulation progresses.

During demodulation, clock is recovered whenever a synchronizable pattern appears in the data. Clock recovery is done in the same way as

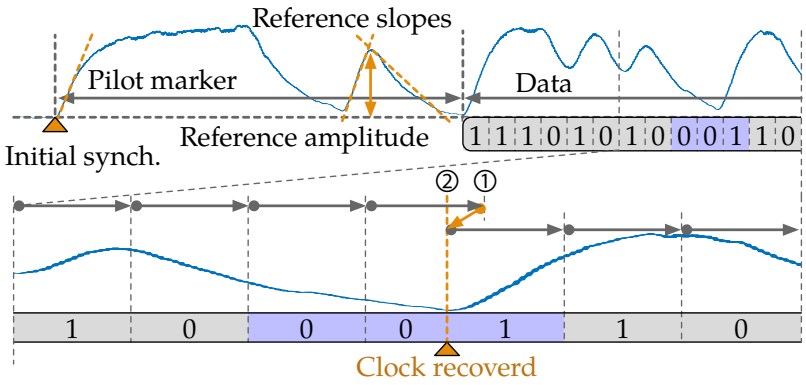


Figure 3.5: Demodulation and synchronization example for k=8. After detecting consecutive bit 0's followed by a transition to 1, the number of samples in the corresponding segment is adjusted based on the measured slope and reference amplitude of the waveform. Segment boundary is adjusted from ① to ②.

the initial synchronization of the pilot marker; the starting point of the bit 1 after the consecutive bit 0's is detected, and rest of the waveform is segmented again from switching point. The sharp increasing slope after consecutive bit 0's makes it possible to precisely detect the starting point of the segment. A synchronizable pattern that appears only after k unsynchronizable bits is treated as a synchronization marker and removed after clock recovery; otherwise, it is kept as data. Therefore, synchronization markers inserted on purpose during modulation stage are not misidentified as data bits. The magnified waveform at the bottom of Figure 3.5 shows an example of the proposed clock recovery. The synchronizable pattern 001 is highlighted. The starting point of its last segment is adjusted from ① to ②, and the subsequent segments are also adjusted. If the number of unsynchronizable bits since the last synchronizable pattern is equal to k, 001 is a synchronization marker to be removed; otherwise, it is a part of the data bits.

In case the length of data bits is unknown, the receiver will continue demodulating the absence of vibration as bit 0's even after the transmission is completed. When the number of consecutive 0's without synchronization exceeds k followed by no synchronization marker, the receiver can detect the exact end of the transmission by finding the last appearance of the end marker pattern. After detecting the end of the communication session, the end marker is removed to leave the data bits only.

Effective Bits per Second for Authentication

As a simplex communication channel without any error correction or detection scheme, the transfer of authentication key should be done with minimal error for OOB authentication. High bit error rate will result in a authentication failure and require a retransmission of the authentication key. Therefore, it is important that the OOB channel has a high rate of success of authentication attempts, which will be referred to as (*authentication*) *success rate*. To account for success rate in addition to the actual bitrate of data transmission, *expected bps* is defined as:

$$\text{bps} \times \text{effective bit ratio} \times \text{success rate} = \frac{1}{t} \times \frac{l}{l+s} \times r \text{ (bps)},$$

where t and r are the vibration period and the success rate, respectively; and l and s are the number of total transferred bits and the number of overhead bits (pilot marker, synchronization markers, and end marker), respectively. Expected bps is directly related to user experience since it is inversely proportional to the expected time needed to complete authentication.

Expected bps is proportional to effective bit ratio and success rate, which are both functions of k. Effective bit ratio and success rate are in a trade-off relationship. A small k will increase the chance of adding synchronization markers, increasing the synchronization frequency during

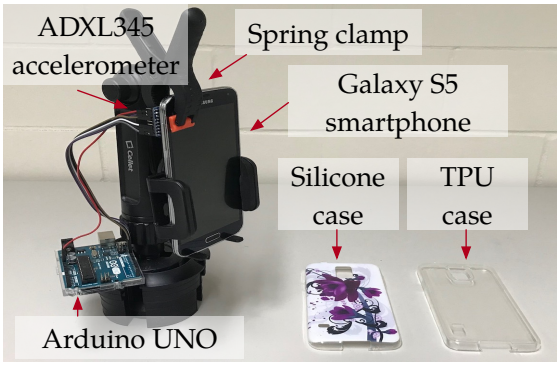


Figure 3.6: Experimental setup. The transmitter (Galaxy S5) and the accelerometer (ADXL345) of the receiver under a constant pressure using a spring clamp.

demodulation, which leads to a high success rate. As a trade-off, frequent appearances of synchronization markers will increase the number of overhead bits and reduce the effective bit ratio. Therefore, the value of k should be carefully selected to maximize expected bps.

3.1.3 Implementation and Evaluation

Implementation of Prototype

Based on the modulation and demodulation technique, a prototype of the transmitter and receiver of SyncVibe is implemented using a commercial off-the-shelf smart phone and its hardware components. Tranmitter uses Android application running on a Samsung Galaxy S5 smart phone with Android version 6.0. The application takes four inputs: bit length of authentication key (L), vibration period (t), synchronization interval (k), and synchronization pattern. The experiments use $t = 40$, 50, and 60 ms, which corresponds to 25, 20, and 16.7 bps, respectively. A 5-bit pattern of `00001` is used as the synchronizable pattern as well as the synchronization

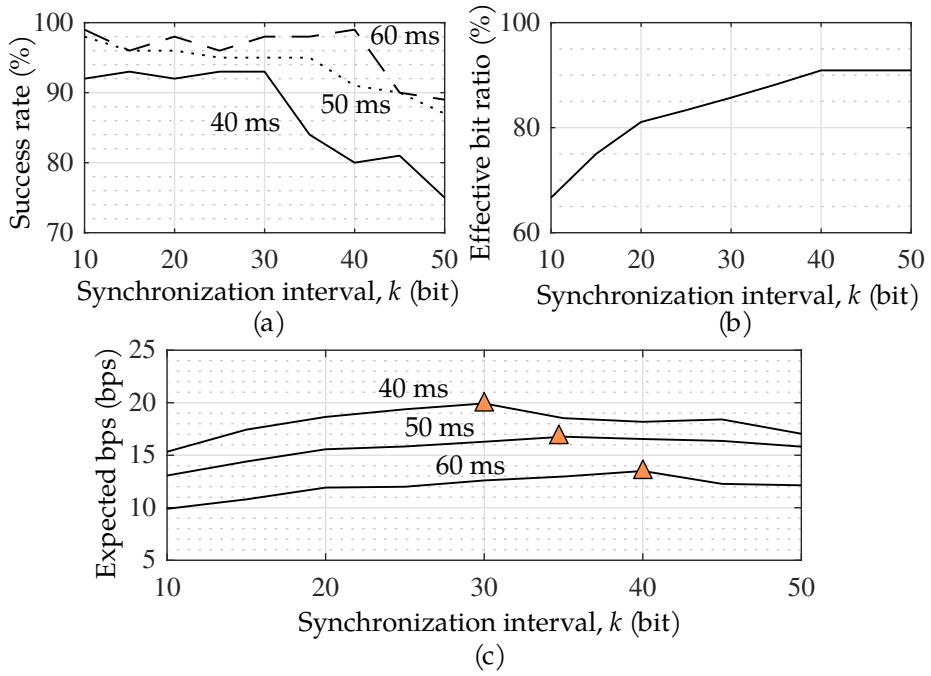


Figure 3.7: (a) Authentication success rate for varying synchronization intervals (k) between 10 and 50 bits and different vibration periods (t) of 40, 50, and 60 ms. (b) Worst-case effective bit ratio for varying synchronization intervals (k).

marker. The receiver prototype is implemented using the Arduino UNO with the ADXL345 MEMS accelerometer (embedded in many mobile and wearable devices today) at a sampling rate of 1600 Hz The transmitter and the receiver are clamped together using a spring clamp, as shown in Figure 3.6, to apply constant pressure. To evaluate the performance of SyncVibe under realistic usage conditions, experiments are conducted with two smart phone cases and two ambient vibration noises in addition to the baseline condition (without a case and noise).

Trade-off in Expected Bits Per Second

First, impact of k to the expected bps is evaluated. The authentication success rate is measured using 100 samples of 150-bit random data bits ($L = 150$), comparable to the typical length of Bluetooth's 128-bit link key, for varying values of k and t. Figure 3.7(a) and (b) shows the measured success rate and effective bit ratio, respectively for $10 \leqq k \leqq 50$. The effective bit ratio is independent of t, and it increases as k increases, showing over a 90% of effective bit ratio for $k \geqq 40$. For example, when $k = 40$, the 5-bit synchronization marker is inserted three times, for every 40 bits, resulting in 15 bit overhead (9.1%) in addition to 150 data bits. On the other hand, the success rate decreases as k increases. It also decreases as the vibration period decreases since the segments are more likely to be unsynchronized when the vibration motor is switched more frequently.

This trade-off between the success rate and the effective bit ratio results in that the expected bps is maximized at $k = 30$, 35, and 40 for $t = 40$, 50, and 60 ms, respectively, as shown in Figure 3.7(c). The figure shows that the maximum expected bps of 13.5, 16.7, and 19.9 bps is achieved when t is 60, 50, and 40 ms, respectively. The rest of the experiments use these optimal k values.

Authentication Success Rate of Different Data Bit Length

This experiment examines the success rate of SyncVibe for varying L. The transmitter attempts to send 25 to 150 bits of random data ($25 \leqq L \leqq 150$) at different t, with the optimal k obtained in previous experiment. The data bits generated may contain synchronizable patterns. The success rate shown in Figure 3.8(a) is around 95%. It does not exhibit a significant dependency on L thanks to the proposed clock recovery performed during demodulation that constantly synchronizes the receiver at least once every k bits. Also, t is not a significant factor to the success rate. SyncVibe con-

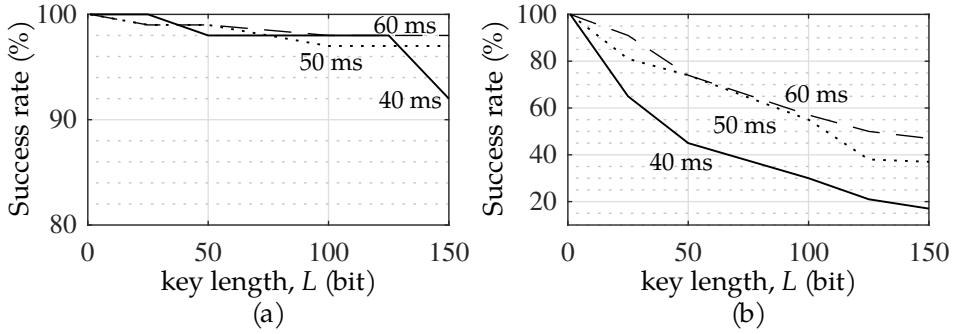


Figure 3.8: Authentication success rate (a) with and (b) without clock recovery. Note the different y-axis scale.

sistently achieves a high success rate of 98%, 97%, and 92% with expected bps of 16.1, 19.0, and 22.2 bps for $t = 60$, 50, and 40 ms, when $L = 150$ bits.

On the other hand, without the clock recovery, the success rate significantly drops as L increases, as shown Figure 3.8(b). For $L \leqq 50$ bits, the success rate is above 70% for $t = 50$ or 60 ms even without clock recovery. However, as bits get demodulated further down the stream, the probability of segment misalignment increases and the success rate decreased to below 60% when $L = 150$ bits. In particular, for $t = 40$ ms, synchronization mismatch propagates down the bit stream, resulting in less than 20% success rate when $L = 150$ bits. Also note that, unlike SyncVibe, the success rate decreases as t decreases due to more frequent segment misalignment.

Transmission Medium and Environment

It is common to use a protective case that covers the back of the smart phone, which can affect the propagation of the vibration signal. To evaluate the impact, success rate and expected bps of SyncVibe are evaluated for different smart phone case materials. The transmitter sends 100 samples of 150-bit random data bits ($L = 150$) at different t while enclosed in a protective case. Two most commonly used materials are tested: silicone and

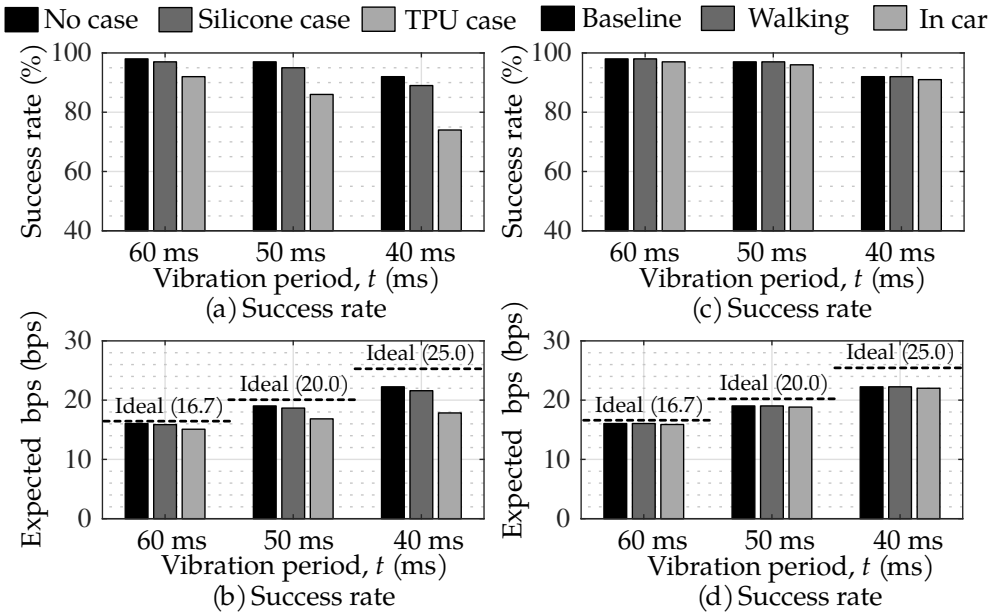


Figure 3.9: (a) Authentication success rate and (b) expected bps for varying vibration periods (t) with two different protective cases: silicone case and TPU case. L = 150. (c) Authentication success rate and (d) expected bps for varying vibration periods (t) under two different noise conditions: walking motion and moving car vibration. L = 150.

thermoplastic polyurethane (TPU). Similar to the previous experiments, t is set to 40, 50, and 60 ms. As shown in Figure 3.9(a), the overall success rate with a protective case results in a slight decrease compared to that without it. In particular, the silicone case shows a 2% decrease on average, and the TPU case exhibits a 5% decrease on average. This is due to that the protective cases absorb vibration, and the vibration signal measured by the receiver is attenuated. On average, the amplitude attenuated by 28% and 33% with the silicone and TPU cases, respectively, compared to the baseline without any case. Figure 3.9(b) presents the expected bps for varying t with different protective cases. Compared to the baseline without a protective case, the expected bps is reduced by less than 1 bps for

Table 3.1: Effective bit ratio, bit error rate, and expected authentication time. L=150.

t	k	**Eff. bit ratio**	**Bit error rate**	**Auth. time**
40 ms	30 bits	97.4%	0.95%	6.74 s
50 ms	35 bits	98.2%	0.61%	7.87 s
60 ms	40 bits	98.8%	0.67%	9.34 s

all t with the silicone case. The maximum loss in expected bps is 4.1 bps, which is caused by the TPU case at a 40 ms vibration period. Overall, SyncVibe maintains high expected bps at short vibration periods even in the presence of vibration-damping materials.

SyncVibe is evaluated under common daily noisy environments that can affect vibratory authentication. Similarly to the previous setup, the transmitter attempted to exchange 150-bit long random data bits ($L = 150$) at different t. Two vibration noises are tested: vibration due to walking motion and vibration in an operating car. As presented in Figure 3.9(c), the success rate of SyncVibe is almost unaffected by these vibration noises. The average reduction in the success rate under the noisy conditions compared to the baseline for all t is less than 1%. The vibration frequency of typical vibration motors is mainly centered above 100 Hz, while the vibration frequency of these noises is centered around 0.5–3 Hz, which can be easily removed. Therefore, the initial band-pass filter applied to raw accelerometer readings before demodulation removes most of the low-frequency noises caused by the walking motion and car. The transmission under different environment shows consistently high expected bps for $t = 40$, 50, and 60 ms, as shown in Figure 3.9(d).

Expected Authentication Time

Finally, SyncVibe's authentication time is evaluated. The experiment evaluates 100 samples of 150-bit authentication key ($L = 150$). The results are shown in Table 3.1. The bit error rate denotes the erroneous demodulation percentages of individual bits, including all overhead bits. Comparable to the worst-case effective bit ratio of authentication key that does not contain any synchronizable patterns (85%, 88%, and 91% for k=30, 35, and 40, respectively) as shown in Figure 3.7(b), the average effective ratio of key remains over 97% for all t due to random data bits naturally containing synchronizable patterns. The high success ratio of SyncVibe is enabled by the low bit error rate of less than 1% at all t, as well as the low overheads. The error rate is 0.95% at $t = 40$ ms and decreases as t increases to $t = 50$ ms and 60 ms.

Using SyncVibe, the user can expect average authentication time of 6.74 s to complete a authentication process for 150-bit key. Under circumstances where transmission channel is noisy due to different transmission mediums and noise conditions, users can flexibly decide to operate with higher t, guaranteeing higher success rate at a cost of higher transmission time. Previous methods of data transfer through vibratory signals without proper synchronization would take up to 19.23 s at $t = 60$ ms with 7.4 expected bps, resulting from low success rate. In comparison, with SyncVibe, the user can expect 9.34 s with $t = 60$ ms for authentication process to complete, achieving 2x faster time.

3.1.4 Conclusion

SyncVibe is a low-overhead, simplex OOB authentication scheme to transmit and receive authentication key through physical vibrations using a vibration motor and an accelerometer, which are widely available in today's mobile devices. For usability, SyncVibe removes the hassle of manually

discovering target device and passkey entering procedure while allowing users to securely bootstrap wireless connection with fast, close-range vibration-based data transfer. For low-error data transmission, SyncVibe's modulation scheme inserts only a minimal amount of synchronization markers so that the receiver can successfully synchronize to reduce bit demodulation error. Additionally, with initial transmission of the pilot marker, SyncVibe can dynamically adjust its demodulation thresholds on different transmission mediums and conditions. The proposed modulation and demodulation schemes are not limited to authentication purposes but can be used in any other short data exchange processes where RF-based communication is not feasible. When transmitting 150-bit key, the prototype of SyncVibe shows a reliable success rate of 92% with average authentication time of 6.74 s, achieving up to 2x faster time compared to previously proposed vibration based communication methods.

3.2 IVPair: Zero-interaction fast intra-vehicle device authentication for secure wireless connectivity

Connecting a mobile device to an in-vehicle infotainment (IVI) system has long been a problem that is much more bothersome than it sounds. To do this, the user has to navigate through multiple steps to discover the device to pair and enter a randomly generated pin or key to verify the device's authenticity. This process is often tedious and lengthy, and sometimes not so user-friendly and unsafe to do while driving that it even discourages users from using it unless the connection is expected to have a long lifetime. When the connection is deemed necessary, since this inconvenient procedure is not considered to be an everyday task, the vehicle's on-board computer system would remember the authenticated device and reuse the

pre-negotiated key, which can be vulnerable to number of attacks [12].

Unfortunately, this authentication mechanism remains surprisingly outdated in spite of the emerging advanced IVI systems, such as Apple CarPlay and Android Auto, that acts more like a smart phone embedded in the car. As the sensitivity of personal data exchanged within the network is much more higher than just an audio playback or personal contact information, today's IVI systems demand higher level of security than conventional car audio systems. Additionally, the utility of such systems would be maximized by inter-operating with mobile devices, often not only the driver's device but also (sometimes anonymous) passengers' devices with short-lived authentication. To meet this emerging demand, a secure and usable mechanism for spontaneous authentication is required to eliminate the inconveniences of conventional methods.

ivPair proposes a secure and usable authentication protocol to extract entropy from the road conditions to automatically generate authentication keys for multiple devices within the same car. While such *mechanical vibration* has been used in some previous work for device authentication [36, 42] since it is easy to measure using commercial mobile devices, there has been little investigation in the context of in-vehicle usage. More specifically, ivPair investigates the use of vibration simultaneously measured by a vehicular computer and a mobile phone in the same vehicle to subsequently establish a secure wireless connection between them. The contributions of ivPair are as follows:

- Presents an intra-vehicle device authentication technique called ivPair, which exploits simultaneously measured vibration to generate a common key to establish a secure wireless connection.

- Designs integral techniques to overcome challenges in realizing ivPair on commercial mobile devices, such as lack of time synchronization and sampling frequency mismatch.

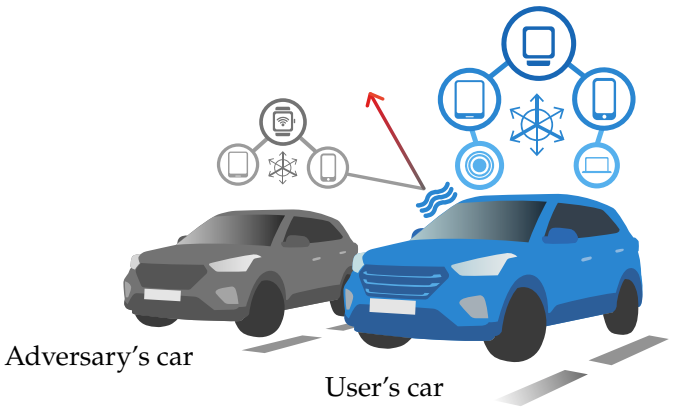


Figure 3.10: Passenger-owned mobile devices in the legitimate user's vehicle are authenticated to the vehicular computer (host), and the devices from the adversary are rejected to legitimate user's vehicular computer.

- Conducts real-world experiments under various driving environments and demonstrate successful key generation and its robustness against adversaries.

3.2.1 System and Threat Models

System model: ivPair considers a scenario where passenger-owned mobile devices within a vehicle are trying to establish a secure high-bandwidth wireless connection (e.g., Bluetooth or WiFi) to the vehicle's computer system by generating identical key while the vehicle is actively in motion, as illustrated in Figure 3.10. Additionally, it assumes that there exists an on-board reference accelerometer attached within the center console of the host vehicle.

Threat model: To be considered as a secure authentication scheme, some common attack scenarios need to be taken into account. ivPair considers an active adversary that is maliciously or unintentionally trying

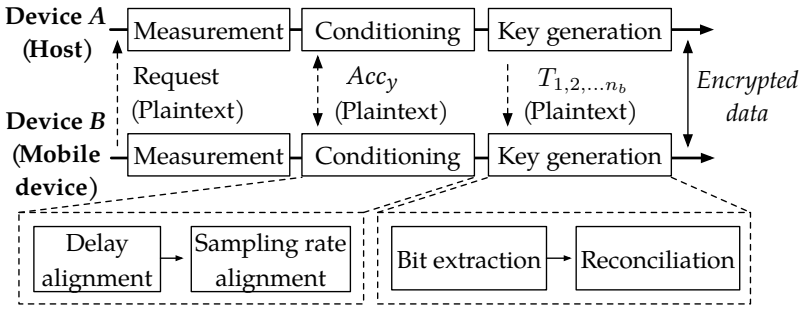


Figure 3.11: Overall protocol to extract identical keys on two devices to bootstrap high-bandwidth wireless connection.

to authenticate to the legitimate victim's vehicle or their mobile devices to tamper with or control the system. The adversary does not have direct physical access and is not present within the victim's vehicle but knows the type of the car and can drive closely to the victim within its wireless range. Additionally, the adversary can eavesdrop on any plaintext wireless messages that are used in the legitimate authentication process.

3.2.2 Proposed Approach

Overview

The only additional hardware component required by ivPair is a reference accelerometer embedded in vehicle in the direction parallel to the car, with y-axis pointing towards the direction of travel and z-axis pointing downwards through the chassis. To authenticate with a mobile device on the go, the user simply holds the device (embedded with an accelerometer) against the moving vehicle's interior door frame closest to his/her sitting position.

Figure 3.11 illustrates the authentication process, which consists of three phases: i) measurement, ii) conditioning, and iii) key generation. To initiate, Device B (user's mobile device) transmits an authentication

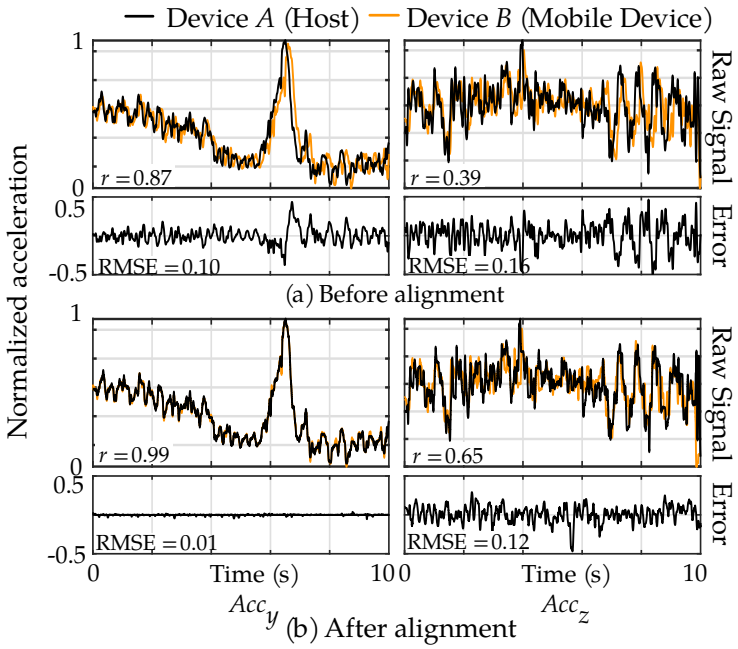


Figure 3.12: Measured Acc_y and Acc_z, sample-wise error, and correlation coefficient r between two devices (a) before and (b) after sampling frequency alignment using DTW.

request to Device A (host vehicle), and both devices independently measure its own acceleration in y- and z- axes. For simplicity, ivPair assumes that the user contacts the accelerometer of the mobile device oriented the same direction as the reference accelerometer since the orientation of two accelerometers observing identical linear acceleration in two orthogonal directions can be easily aligned [7, 75]. Following the conditioning and key generation phases, the two devices generate identical key to communicate through a secure encrypted channel. Note that all information exchanged between two devices until the completion of the key generation phase is in plaintext that can be eavesdropped by the adversary.

Measurement and Conditioning

The main source of entropy to generate a key is the vibration response of the moving vehicle perpendicular to the direction of travel. That is, the acceleration signal from z-axis, $Acc_{z,u}$, is utilized to extract keys from both devices, where u denotes one of the devices, A or B. However, the raw measurement cannot be directly used to extract bits due to significant temporal misalignment caused by: i) time offset resulting from the transmission delay of the authentication request message and ii) sampling frequency mismatch caused by variation between the devices. To achieve temporal alignment without revealing secret, two devices leverage the acceleration in y-axis, $Acc_{y,u}$, which represents the vehicle's linear acceleration towards the direction of travel resulting from the driver's behavior of accelerating and breaking. Note that, since Acc_y is more predictable by an external observer, it is used for signal conditioning only, but should not be used for actual key generation.

More specifically, the devices utilize the sliding window approach to find the index of a sample that exhibits the highest correlation between its own Acc_y and the other's Acc_y. However, while this process may synchronize the starting points, the sampling rate variation between the devices results in additional misalignment as the number of measurements accumulate. Therefore, after aligning the starting points, ivPair corrects the sampling frequency discrepancy by adopting dynamic time warping (DTW) [6] on synchronized $Acc_{y,A}$ and $Acc_{y,B}$ to calculate the optimal correspondence between them. Each device extracts its non-linear warping path i_A and i_B which represents the indices of $Acc_{y,A}$ and $Acc_{y,B}$ with minimum distance with respect to each other. Then the devices independently apply their warping path i_A and i_B on $Acc_{z,u}$, to obtain tightly aligned fingerprints F_u with respect to each device.

Figure 3.12 compares Acc_y and Acc_z measured from two devices and their sample-wise error before and after the proposed signal condition-

ing. As shown in Figure 3.12(a), without the DTW-based alignment, two signals on both axes are severely misaligned even within a 10 s measurement. The sample-wise error plot shows that the magnitude of the error gradually increases as the sampling time increases, due to the sampling frequency variation between two devices. On the other hand, after they are aligned using the warping path calculated from Acc_y, as illustrated in Figure 3.12(b), the error rate of two axes is drastically reduced (from root mean square error (RMSE) of 0.10 to 0.01, and 0.16 to 0.12, respectively), resulting in a significant improvement in correlation between $Acc_{z,A}$ and $Acc_{z,B}$ from 0.39 to 0.65.

Key Generation

The two time-aligned fingerprints F_u obtained by two devices are the main source of randomness to harvest identical bit sequences. To quantize F_u into bit sequences K_u, the key generations phase employs the noise-based random bit generation method, where time-series fingerprint signals are uniformly segmented into several subsections n_b, and the index of the maximum absolute value, T_b, in each subsection is exchanged to be converted into bits. If the signal value at each index is greater than the mean of the subsection, a bit 1 is extracted; otherwise a bit 0 is extracted. Because there exists no periodic nature in the high-frequency components of F_u, the phase segment the entire fingerprint into n_b subsections which represents the number of extracted bits and extract bits as follows:

$$K_{u,b} = \begin{cases} 1 & \text{if } F_{u,b}(T_b) \geqq \text{mean}(F_{u,b}) \\ 0 & \text{if } F_{u,b}(T_b) < \text{mean}(F_{u,b}). \end{cases} \tag{3.1}$$

This bit extraction scheme results in nearly identical sequences but may exhibit occasional bit errors due to remaining timing mismatch. To resolve these errors without leaking any information about the key itself,

the following reconciliation phase utilizes error-correcting code (ECC) to map equivalently segmented bit sequences to one of the pre-computed codewords. For instance, when using Hamming(n, k) as a base ECC, the results of equally segmented n-bits from the bit extraction phase will map to a n-bit codeword that exhibits the minimum Hamming distance. Note that during this phase, no information about the key itself is exchanged between two devices.

3.2.3 Implementation and Evaluation

Implementation

The performance of ivPair is evaluated with different body types of vehicles driven on various types of roads. In total, more than 3-hour worth of real-world driving data is collected using triple-axis ADXL345 MEMS accelerometer connected to Arduino Uno boards at a sampling frequency of 800 Hz. For each authentication attempt, 10-s long accelerometer measurement is used to extract 14-bit keys. The protocol employs Hamming$(7,4)$ as the ECC for reconciliation to resolve bit errors. To simulate the user holding the mobile device against the interior panel of the vehicle, experiments use adhesive tape to fix a reference accelerometer (representing the host) to the center console as well as different positions within the car. For all driving environments, the driver maintained safe driving behavior without any aggressive or abrupt accelerating and breaking activities to intentionally improve signal-to-noise ratio.

This work primarily focus on two evaluation metrics: the *bit agreement rate* and the *success rate* of authentication attempts. Bit agreement rate refers to the rate of equal bit-wise comparison results between two generated keys before reconciliation, and success rate represents the rate of successful authentication that exhibits a perfect (100%) bit agreement rate after reconciliation. Additionally, as a measure of user experience ac-

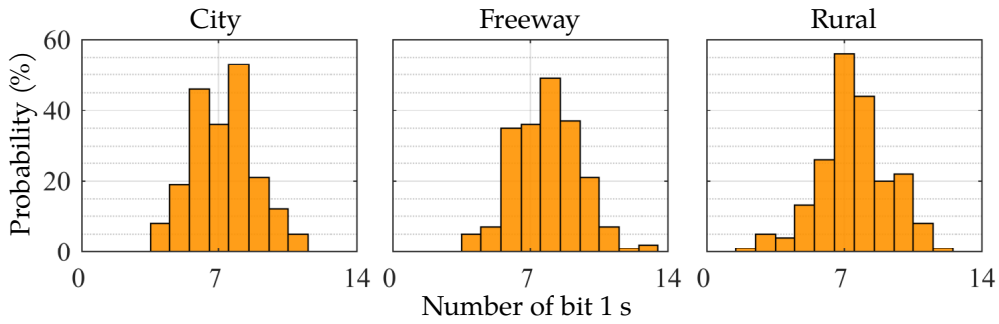


Figure 3.13: Histogram of 14-bit key based on their number of bit 1s.

counting for authentication failure scenarios, *expected authentication time* is defined to be inversely proportional to the success rate times the duration of the measurement (10 s).

Bit Randomness

First, in order to investigate the quality of bit sequences generated from the ivPair, experiments record the histogram of the frequency of bit 1's in the generated bit sequences. To prevent an adversary from randomly guessing the key, a high-quality key should contain statistically equal number of bit 0's and 1's. If the sequence dominantly embeds more number of 1's than 0's or vice-versa (i.e., biased), the contexts that are used for the fingerprint extraction is not considered ideal.

Since ivPair determines each bit by the relative magnitude of random noise and the mean value based on (3.1), the probability of appearance of a bit 1 and a bit 0 are equally likely. Ideally, in the case of a 14-bit key, the number of bit 1's should be around 7 to indicate frequent contextual changes within the measured fingerprint signals. Figure 3.13 illustrates the histogram of 100 sequences generated from the city, highway and suburb driving conditions. All three distributions exhibit a binomial distribution centered around 7 with no sequence that exhibits continuous

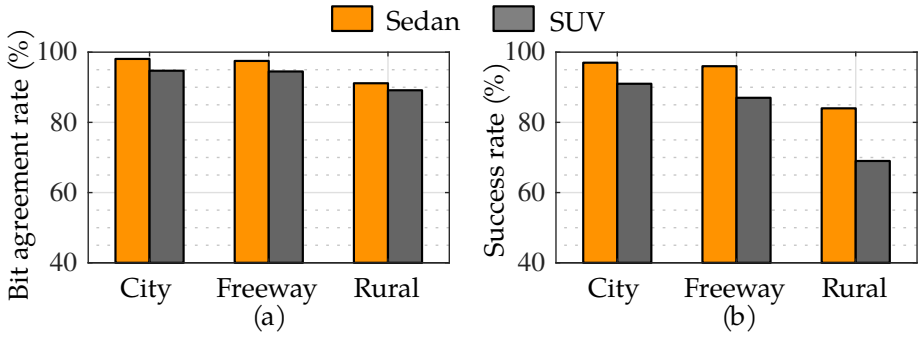


Figure 3.14: (a) Bit agreement rates and (b) success rate on sedan and SUV driven on different roads.

0s or 1s, which indicates the presence of entropy and randomness in the fingerprints that makes it difficult for the adversary to randomly guess the established key.

Vehicle and Road Types

Different vibration responses resulting from different types of vehicles, roads, and traffic conditions can affect the overall authentication process. In order to investigate these variations, experiments are conducted using a sedan and a sport utility vehicle (SUV) driven on the city, freeway and rural roads. One accelerometer fixed to the driver side door frame is requesting to authenticate to the host fixed to the center front console. For each road type, 100 authentication attempts are made. Overall, as Figure 3.14(a) illustrates, both types of vehicles show high bit agreement rates. In particular, the sedan type vehicle achieves 98.1% bit agreement rate in the city, while the SUV type exhibits 95.0%. This is due to the fact that the higher chassis and clearance height of the SUV results in a higher sensitivity to road and traffic conditions that leads to a slight difference in vibration responses between two devices. The results also show overall high bit agreement rates for all types of roads, close to exceeding 90%.

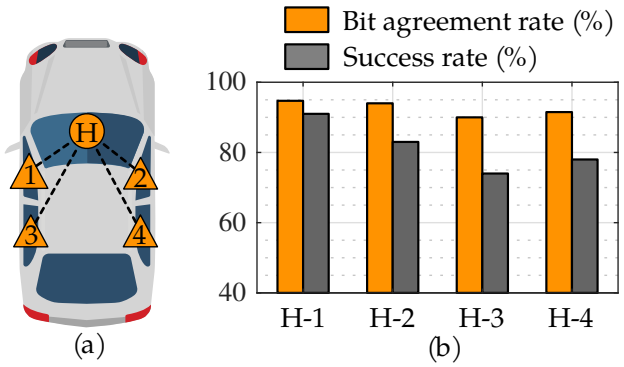


Figure 3.15: (a) Location of devices (H: host, 1–4: mobile devices). (b) Authentication success rate and bit agreement rate between pairs of devices.

Rural driving exhibits slightly lower agreement rates in both vehicle types compared to the freeway and city driving due to unstable accelerometer data from frequent and larger bumps and cracks on unpaved road surfaces.

As illustrated in Figure 3.14(b) the high bit agreement rates lead to high success rates above 85% for all freeway and city driving in both vehicle types. While the SUV case in rural driving shows the lowest success rate out of all cases at around 69%, city driving exhibits high success rates of 97% and 91% for the sedan and SUV, respectively. These results indicate that even under variations caused by different roads and vehicle types, a high success rate is maintained at 87% on average.

Location of Mobile Devices

Next, experiments are conducted to investigate the performance of ivPair at varying locations within the vehicle. Figure 3.15(a) shows the locations of the host as well as other mobile devices' location. For each device pairs, 100 authentication attempts are made on the city roads. Figure 3.15(b) shows the bit agreement rate and the authentication success rate for four different location pairs. The devices placed in the front seats, closer to the

Table 3.2: Expected authentication time and mean correlation coefficient before and after conditioning.

Device pair	Correlation coefficient	Expected time
Host-1	$0.11 \rightarrow 0.79$	11.0 s
Host-2	$0.06 \rightarrow 0.78$	12.0 s
Host-3	$0.32 \rightarrow 0.65$	13.5 s
Host-4	$0.09 \rightarrow 0.61$	12.8 s

host accelerometer, show an average agreement rate of 94.7% and 94.0% for the driver side (H-1) and the passenger side (H-2), respectively. The devices that are located in the rear seats achieve slightly lower agreement rates due to the natural location variation that leads to a slight difference in its fingerprints. However, the experiments suggest all the passengers in the vehicle will experience an acceptable success rate of above 70% with a mean of 85%, regardless of their seat position.

It also shows that ivPair's conditioning process (DTW-based sampling frequency alignment) significantly improves the correlation between fingerprints generated by device pairs. As presented in Table 3.2, all fingerprint pairs exhibit low mean correlations (0.15 on average) before conditioning. However, after conditioning, the mean correlations are dramatically improved to 0.71 on average, enabling successful authentication at a high probability.

Adversarial Scenarios

ivPair considers two adversarial attack scenarios according to its threat model. The *same-lane* scenario is where the adversary is actively driving with the target vehicle in the same lane. To maximize the adversary's bit agreement rate against the victim vehicle, the adversary is driving in front of the victim vehicle and delays its measured fingerprint signal to

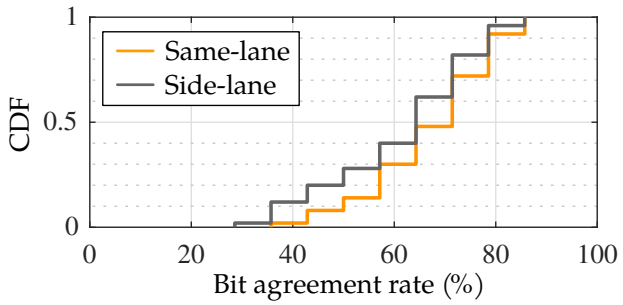


Figure 3.16: Bit agreement rate achieved by the adversary under two different attack scenarios.

account for slight timing difference caused by the distance between two vehicles. Additionally, in the *side-lane* scenario, the adversary is driving the car side-by-side in a multi-lane road. Experiments are conducted with two sedan vehicles within the city driving condition and assume that the adversary is equipped with identical hardware settings as the victim. In total, the adversary attempts 50 authentication requests in each scenario, utilizing the acceleration measured from the adversary's vehicle. The bit agreement rates resulting from the two attack scenarios are presented in Figure 3.16. Overall, the adversary conducting the side-lane attack is able to achieve a mean bit agreement rate of 61.3% compare to the legitimate key. The adversary was able to guess only up to 85.7% of the legitimate key in two out of 50 attempts before reconciliation, which is below the threshold to authenticate with the victim vehicle. On the other hand, in the same-lane attack scenario, the adversary achieves a slightly higher mean bit agreement rate of 70%. This is because the adversary mimicking the legitimate fingerprint in the same-lane is more likely to experience the same bumps and cracks than in the side-lane scenario. However, under both attack scenarios, none of the attempts successfully authenticates with the victim's vehicle.

Computational Overhead of DTW

The main computational overhead of ivPair is the execution of DTW algorithm in the signal conditioning phase to achieve sampling rate alignment. In order to validate the feasibility of computing DTW on commercial mobile devices, ivPair is implemented an Android application running on LG Nexus 5X (Android 5.0) with 1.8 GHz processor, which is a mid-range smart phone nowadays, and measure the algorithm's computational run time. The application takes in two discrete time series of 8,000 samples (10-s long fingerprint at 800 Hz) and matches the samples of one series to another. On average, computing the alignment path takes only 564 ms, which indicates that the computational overhead does not significantly affect the usability of ivPair.

Discussion

Vehicle must be in motion: For ivPair to work, it needs a common source of linear acceleration in one direction to obtain tightly synchronized fingerprints, which requires the vehicles to be in motion. All results presented in the evaluation come from moving vehicles and do not include data gathered while the vehicle has stopped due to a traffic light or a stop sign because the accelerometers would produce meaningful waveform from bumps and tears only when the vehicle is moving. This requirement makes ivPair slightly constrained in terms of its usability to authenticate anytime. However, considering that the need for a convenient device authentication method is more imperative when the driver should not be distracted, ivPair would be useful in practical scenarios.

Entropy vs usability trade-off: In the evaluation, it takes about 10 s to extract a 14-bit key—roughly the same amount of information as in a four decimal-digit Bluetooth pin—while the vehicle is in motion. If the length of bits extracted from the same duration (10 s) of the fingerprint

increases, the bit agreement rate would decrease due to the differences in the fingerprints caused by sensor variation and locality. This phenomenon is intrinsic to all ZIA schemes that relies on an ambient entropy source, and further investigation is needed in order to evaluate the trade-off between security (bit length) and usability (expected authentication time).

Human factors: In practice, substantial change in the user's handling of the mobile device (i.e., pressure or orientation) may affect the fingerprint that can cause the authentication attempts to fail. Ideally, the process should be seamless—the user should be able to authenticate a device while holding it in their hand or in the pocket. To deal with this problem, I could treat the user's body—including the seat and their hand grasping the phone—as a linear time-invariant system that filters bumps from the road before the bumps can be measured by the device's accelerometer. Additionally, the experiments are conducted in the vehicle with no background audio or music. I imagine that loud sounds can cause the door frames to vibrate since the speakers are usually embedded in the frames.

3.2.4 Conclusion

In conclusion, ivPair proposed a fast and convenient method for authenticating devices within the same vehicle. The results from extensive experiments show that ivPair can complete device authentication within a reasonable time at a high success rate in various vehicle types and road conditions as validated with extensive real-world experiments. It is shown that it can successfully reject nearby adversaries in the same or next lane. The proposed method would enable seamless and more secure connection between mobile devices and emerging IVI systems, potentially facilitating innovative mobile applications with short-lived device authentication.

4 ZIA FOR INDOOR IOT DEVICES

In this Chapter, I present two device authentication techniques designed for devices that are used within the indoor environment. The first work named VoltKey is a method that transparently and continuously generates secret keys for colocated devices, leveraging spatiotemporally unique noise contexts observed in commercial power line infrastructure. The second work named AeroKey leverages ubiquitously observable ambient electromagnetic radiation to autonomously generate secret key that can only be derived by devices that are closely located to each other. The two works successfully leverage the omnipresent contexts within the indoor environment, and therefore is considered complete zero-interaction because they do not require any human-assisted touch or any explicit action. This can principally be advantageous in terms of security as it allows devices to autonomously and periodically update the network authentication key, significantly reducing the attack window and increasing the usability during key update process.

4.1 VOLTKEY: CONTINUOUS SECRET KEY GENERATION BASED ON POWER LINE NOISE FOR ZERO-INVOLVEMENT AUTHENTICATION

In this work, I introduce device authentication named VoltKey, which can be used to realize ZIA, leveraging the plug-in power source of devices to extract a secret key from the dynamic characteristics of *electrical noise* present on the power line infrastructure. More specifically, VoltKey takes advantage of the fact that devices that are powered by colocated electrical outlets, or those that are within the same *authenticated electrical domain*, observe similar *noise fingerprints* caused by the nearby electrical environment

which is temporally and spatially unique. VoltKey can be embedded in standard USB power supplies that are pervasively used in personal and domestic IoT devices or embedded in the IoT devices themselves. Because it exploits standard power line infrastructure that is ubiquitously available virtually everywhere, VoltKey does not require additional supporting infrastructure for installation. Using VoltKey, devices that wish to associate with one another can simply be plugged into an existing power outlet to automatically generate (and periodically regenerate) a unique key and associate themselves with no involvement from the user.

In VoltKey-enabled networks, a device's ability to authenticate itself is dependent on its physical proximity to the host access point. The boundaries of an authenticated electrical domain are determined by the electrical interconnect of the site. Specifically, devices being authenticated must be connected to a single circuit breaker. A typical single-family house or a medium-size office has a few circuit breakers, hence a few authenticated electrical domains. While this requirement bars benign devices from being authenticated across a circuit breaker, this is a acceptable range considering the physical distribution of IoT device managed by a person. This also means that a malicious attacker who has physical access to the authenticated electrical domain cannot be thwarted by VoltKey. Therefore, existing secure authentication protocols will be required to augment security if such an attack is expected.

In summary, this work makes the following contributions:

- I introduce a technique to extract randomness from power line noise measurements and convert it to random bit sequences that enable secure authentication without user involvement.

- I propose a protocol as well as a suite of techniques for establishing time and sampling rate matching among pairs of IoT devices that attempt to authenticate with each other.

- I implement a low-cost hardware prototype of VoltKey and evaluate it in a variety of environments: office, home, and lab settings. Moreover, VoltKey demonstrates that devices can reliably authenticate each other within the same authenticated electrical domain for all environments and reject potential adversarial devices outside of the domain.

Power Line Noise

VoltKey generates secret keys by harvesting randomness from the power line. The important characteristics of the power line noise that VoltKey exploits are (1) it encodes enough randomness to generate authentication keys and (2) the noise is similar within a small set of nearby outlets but different in outlets that are in electrically distant locations. Generally, power line noise is dependent on the local environment, including other devices drawing power from the same electrical bus and electromagnetic radiation absorbed by the power lines.

The first, caused by nonlinear circuit elements drawing power from the power line, produces baseband impulsive noise in either transient or continuous form [53, 10]. Transient noise results from switching activities of electrical devices as it power cycles from off to on state or vice versa and typically lasts for up to few milliseconds. On the other hand, continuous noise is constantly produced by operating devices that utilize motor (i.e, fans and hair dryers) or silicon controlled rectifiers for the duration that the device is operating. Generally, these nonlinear elements inject noise at a harmonic of the fundamental frequency of either 50 or 60 Hz. This type of conducted noise tends to last for the duration of several microseconds up to a few milliseconds of random variations [81].

The second effect, caused by electromagnetic radiation from nearby devices, is known to generate electromagnetic noise signals that are weak and noisy compared to sinusoidal AC voltage [44]. This noise present on

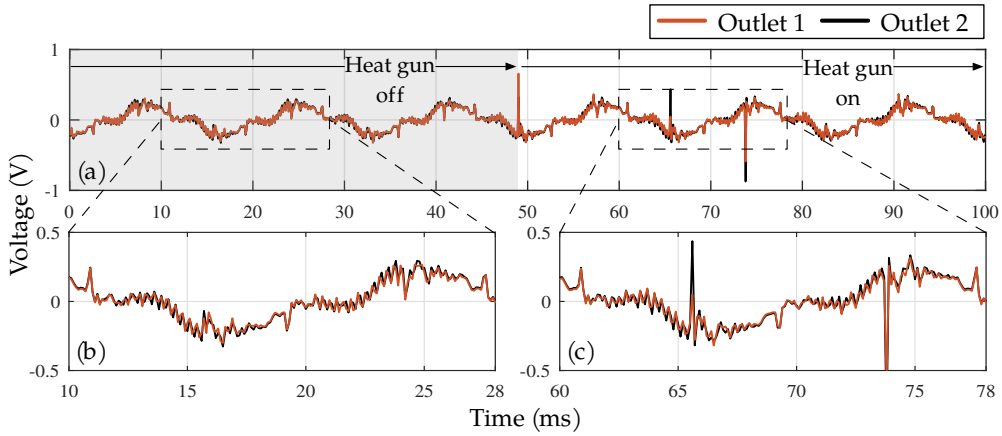


Figure 4.1: (a) Measurement of voltage signal on two colocated outlets using a USB DAQ at a sampling rate of 10 kSPS. Single period of 60 Hz signal (b) when the heat gun is off and (c) when it is on.

the power bus is generated by electromagnetic interference (EMI) both from nearby and distant radiant sources. Power lines—long stretches of conductive copper—are excellent antennas that can be excited by a broad range of radio frequencies [11]. Electromagnetic noise from nearby radiation sources is dense with randomness, and it is strongly dependent on number and types of surrounding electrical devices as well as the specific geometry and interconnects of the power wires in the walls [44]. Unlike conducted impulsive noise, this noise is not periodic. This type of noise usually lasts over periods of seconds up to several hours and is classified as background noise in power line [81].

The combination of the two previously mentioned noises makes VoltKey perfectly suited for key generation purposes because it is temporally and spatially unique, difficult to fake, and it generally requires physical access to measure. To verify the characteristics of electrical noise generated from nearby sources and its similarity from two colocated outlets, I measure the voltage signals on two outlets (less than 20 cm apart) and power cycle a heat gun from an electrical outlet located 1.0 m away from the measuring

point. Figure 4.1 illustrates two measured voltage signals using the National Instruments USB-6218, a multi-channel USB data acquisition (DAQ) device with 16-bit resolution at a sampling rate of 10 k samples per second (SPS). I use an analog notch filter to attenuate the 60 Hz fundamental frequency, but non-idealities in the analog components, such as series resistance in the capacitors, do not completely eliminate 60 Hz component. In Figure 4.1(a), noise signal that is superimposed from surrounding active power supplies from computers, LED light bulbs, etc. shows a close correlation between two colocated power outlets with root mean squared error (RMSE) of 0.03 V. As nearby heat gun switches on at 48 ms, the period shows a significant difference in peak amplitude with RMSE of 0.11 V compared to the period without the heat gun's noise component. The single period of 60 Hz signal when the heat gun is off and on is illustrated in Figs. 4.1(b) and 4.1(c), respectively. While significant peak difference in the periods indicates different noise signatures generated by nearby active sources, the measured signal encodes relatively little randomness with distinct sinusoidal behavior. To address this issue, further signal processing is done to extract noise components. Note that the structure of power line noise is local and time-variant, and depending on the way a building is wired, the noise structure may vary considerably.

Challenges

There are two key challenges that must be addressed in order to realize practical power line-based ZIA. First, generated keys should be *random and unpredictable*. The most dominant signal in the power line is the deterministic sinusoidal wave with a frequency of 60 or 50 Hz. Moreover, each electronic device generates a unique and consistent noise pattern that is distinct enough to be used for identifying one from each other [53]. Therefore, the key generation method should be capable of producing random bit sequences in the presence of strong predictable signals. Second,

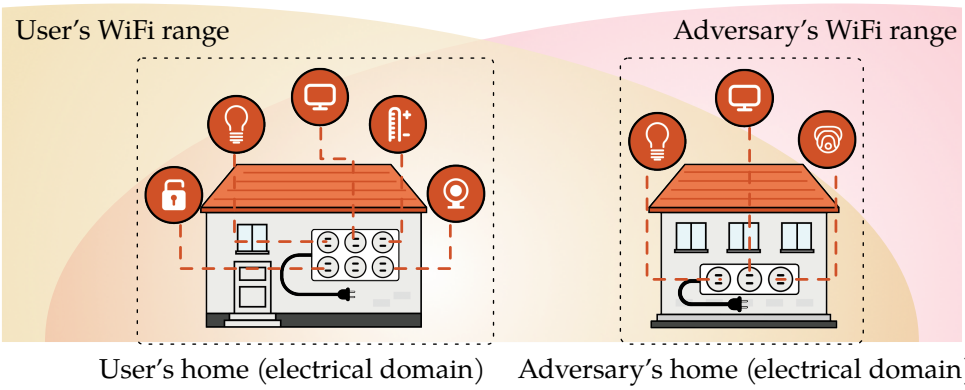


Figure 4.2: System and threat models of VoltKey. A number of IoT devices are installed in each home. WiFi range of each home can reach neighboring homes, potentially the adversary's.

it should impose *minimal hardware and software overheads*. Low-cost IoT devices cannot afford to embed an expensive high-precision measurement circuit. Inexpensive measurement circuits are more prone to process and temperature variability, which leads to significant inconsistency between different devices' measurement results. Therefore, the key generation method should be able to mitigate the hardware limitations with a minimal software effort without compromising security. VoltKey successfully addresses these challenges with novel key generation and device synchronization techniques achieved with low-cost hardware design.

4.1.1 System and Threat Models

System model: VoltKey assumes a scenario where a number of IoT devices are colocated in their owner's home, as shown in Figure 4.2. In each home, all wall outlets are connected to the same load center (or circuit breakers) that defines an electrical domain. Each home has a WiFi access point, and its coverage can reach neighboring homes. Stationary

devices, such as WiFi access points, smart thermostats, and smart light bulbs, are constantly powered by VoltKey-enabled power adapters that periodically generates secret keys for each device. In this scenario, an IoT device that has no prior trust with the wireless access point tries to establish trust and join the secure WiFi network using a symmetric cryptographic key, and the cryptographic key needs to be periodically updated. Additionally, although VoltKey is not limited to a specific power line voltage or frequency, I assume 120 V and 60 Hz throughout this work.

Threat model: The adversary is the owner of an IoT device located outside of the legitimate user's home, potentially in a neighboring home. The user and the adversary are within the range of each other's WiFi coverage. The adversarial device refers to any device that is trying to illegitimately gain unauthorized authentication to the legitimate device for additional threats. The adversarial device can intercept unencrypted packets within its WiFi range and listen to public discussion. Also, VoltKey assumes that the adversary has physical access to any other electrical domains (e.g., neighboring home), but not to the user's electrical domain. The adversary does not have the ability to install a rogue device in the user's electrical domain and leave it there without the user's knowledge. Under normal circumstances, such a device would be immediately noticed by the user, unless it was hidden (e.g., inside a circuit breaker panel), which would require a tremendous effort. In addition, the adversary knows the daily usage pattern of the dominant electrical loads of legitimate user that are active during a specific time of the day.

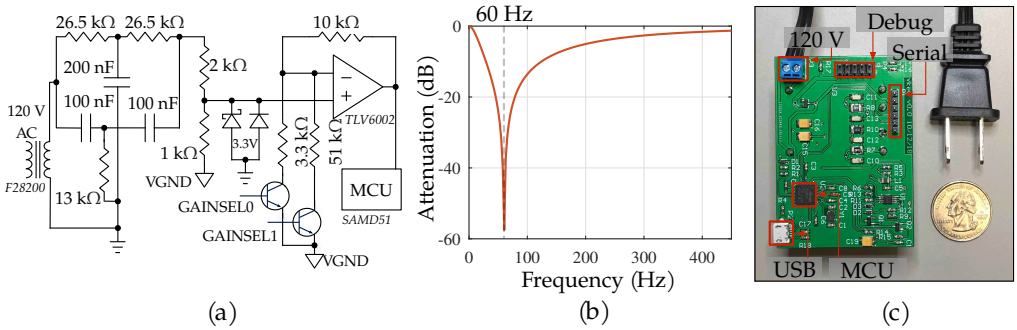


Figure 4.3: (a) VoltKey's Analog front-end schematic. MCU's power regulation, debugger and serial communication circuitry is omitted for simplification purposes. (b) Frequency response of the twin-T notch filter used in the prototype. (c) Top-view of VoltKey prototype.

4.1.2 Proposed Approach

VoltKey Hardware Design

VoltKey is designed as a modular addition to standard USB or AC/DC power supplies shipped with IoT devices. In addition to supplying power, the module also generates keys from superimposed noise on the power line and transmits the keys to the device over a wired interface for authentication purposes. VoltKey consists of two main components: (1) the analog input circuitry for filtering and amplifying power line noise and (2) microcontroller unit (MCU) which includes an analog-to-digital converter (ADC) for noise measurement and key extraction procedures.

Analog front-end The analog front-end, illustrated in Figure 4.3(a), consists of an isolation transformer, a twin-T notch filter, and a differential amplifier. The purpose of this circuit is to amplify high-frequency noise from the power line and attenuate the 60 Hz fundamental. The transformer steps the 120 V AC power signal (between hot and neutral) down to a lower voltage and isolates the VoltKey circuitry from the power line. The

prototype uses a split-core transformer with two secondary coils: one to generate power for the circuitry and the host and another to measure noise. As illustrated in Figure 4.3(b), the twin-T notch filter attenuates the 60 Hz fundamental frequency component from the voltage waveform. The 60 Hz component is an unwanted signal in the context of VoltKey because its harmonics carry a deterministic signal that repeats almost identically from period to period. Therefore, attenuating it improves the signal-to-noise ratio (SNR). After the signal has been filtered, it has an amplitude of 200–300 mV and an average value of 0 V. The amplifier's job is to shift and amplify the filtered signal so its range is within 0–3.3 V, the limits of the ADC. The diodes at the end of the filter clip the filtered analog voltage waveform between 0–3.3 V to avoid damaging the op-amp and the ADC of the MCU. VoltKey uses an op-amp to generate a virtual ground of 0.7 V, and the output of the twin-T notch filter is referenced to the virtual ground using a voltage divider (immediately to the left of the diodes in Figure 4.3(a)). The amplitude of the noise varies considerably depending on active electrical loads. When the signal amplitude is too small compared to the ADC dynamic range, the prototype may get poor measurement results due to large quantization error; if it is too big, the peaks of the noise will be clipped by the diodes and lost. To deal with this issue, prototype builds an adjustable gain amplifier to allow software to dynamically adapt to changing noise conditions, adjusting the gain accordingly. The adjustable gain amplifier is built from a standard configuration of a non-inverting op-amp circuit with bipolar junction transistors in the feedback loop between the inverting input and VGND. The `GAINSELx` signals are connected to the microcontroller's GPIO lines via bias resistors, allowing the software to modify the amplifier's gain.

MCU and ADC VoltKey uses a low-cost MCU to measure and process the voltage signal on the power outlet. The hardware prototype is

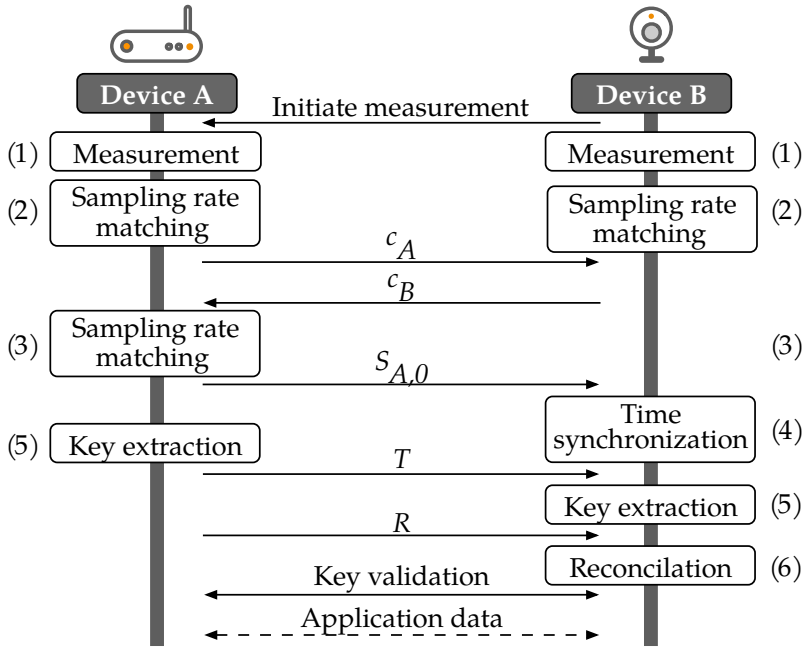


Figure 4.4: Overview of VoltKey's key establishment protocol. Solid lines denote plaintext messages exchanged on a public channel and dotted lines represent encrypted messages.

equipped with the Microchip's ATSAMD51, a 32-bit ARM-Cortex M4 [27], running at 120 MHz with an on-chip ADC capable of a sampling rate of up to 1 MSPS at a 12-bit resolution. The MCU also has a USB device functionality which can be used to transfer the computed keys to the host over a virtual COM port (serial) interface. The MCU chosen is considerably more powerful than necessary—VoltKey's application uses very little memory and can run on a low-power processor.

Key Establishment Protocol Overview

The overall key establishment protocol to bootstrap a full-duplex communication channel between two devices (A and B) consists of the following

steps (illustrated in Figure 4.4). Solid lines depict plaintext messages exchanged through a public channel, whereas dotted lines represent encrypted messages. This protocol is designed to address the aforementioned challenges—extracting common secret keys while compensating for variabilities. It allows two devices to gather time-synchronized samples of the voltage waveform from the power line and ensure minimal information leakage so that an eavesdropper cannot derive the final key from the information revealed on the public channel. More specifically, the protocol consists of the following steps. (Numbers correspond to Figure 4.4.)

1. *Measurement:* Device B (e.g., an IoT device) contacts Device A (e.g., a WiFi access point) to initiate independent power line noise measurement. Let S_A and S_B be the measurement results of A and B, respectively. Note that the sampling clock (time and rate) can vary between A and B.

2. *Sampling rate estimation:* Each device independently goes through the sampling rate estimation procedure based on S_A or S_B. Let c_A and c_B be the estimated sampling rate of A and B.

3. *Sampling rate matching:* Device A or B performs sampling rate matching based on c_A or c_B to align the sampling rate (in samples per 60 Hz period) of S_A or S_B.

4. *Time synchronization:* Device B synchronizes its measurement time to that of A, using $S_{A,0}$, a short snippet of S_A received from Device A.

5. *Bit sequence extraction:* Both devices independently execute bit sequence extraction procedure based on the timestamp T provided by A.

6. *Reconciliation:* Differences in the extracted bit sequences are corrected by B with publicly exchanged data R through key reconciliation stage).

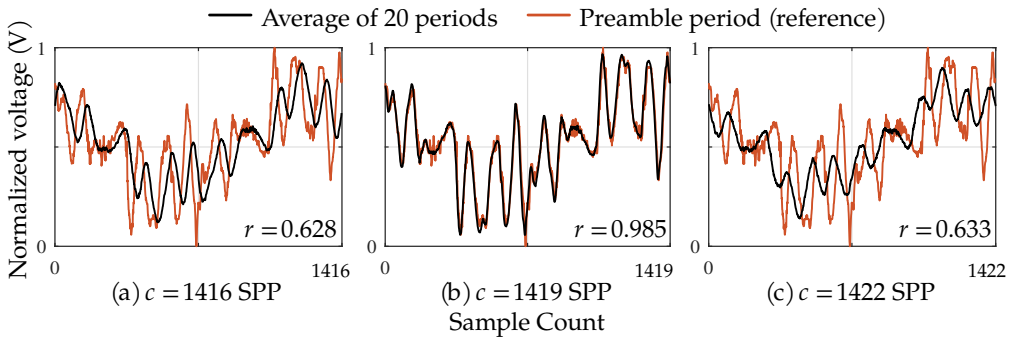


Figure 4.5: Mean of uniformly sliced signal at (a) c =1416 SPP, (b) c =1419 SPP, and (c) c =1422 SPP. The correlation coefficient is highest when c is equivalent to the actual SPS divided by 60.

When the protocol is successful, both devices A and B have identical keys that can be used for authentication and encryption. To periodically update the cryptographic key, this protocol is repeated at preset intervals. The rest of this Section describes the detailed procedure of each step.

Terminology From here forward, while describing VoltKey, the texts will refer to the random bits extracted from the voltage waveform as a *bit sequence* before reconciliation. After reconciliation completes successfully, both devices will share an identical *key*. The difference is that the bit sequences may have a few bit errors between the devices, but the reconciled keys will be identical.

Estimation and Matching of Sampling Rates

VoltKey uses an ADC on each device to sample and process the noise signal measured from the power line. Since each device samples the signals (S_A and S_B) independently, the sampling rate of the ADC on both devices must be identical before timing synchronization can take place. However, commercial low-cost MCUs often suffer from timing

variability, and moreover, the variability is time-varying. For example, the internal ultra-low-power oscillator frequency of the MCU in the prototype, ATSAMD51, can vary up to $\pm 2\%$ even at a constant room temperature [27]. Extreme temperature variation can cause more severe frequency variability ranging from -17% to $+15\%$. The measurement shows about 0.5–1% of frequency variability among only five different prototype boards.

In the VoltKey system, each device independently derives the exact rate at which their measured signal is sampled, using the periodicity of the 60 Hz sinusoidal voltage waveform from the power line as a common time base. Devices that wish to establish a key first agree on an approximate sampling frequency r, in the range of tens to hundreds of kilohertz. Each device samples several periods of the 60 Hz voltage waveform at its approximate sampling rate. Then, the measured signal S_u is uniformly sliced into sequences of length c from the starting point, where c is the samples per period (SPP), i.e., SPS of the ADC divided by 60. Among these sequences, the first sequence of the slice is referred to as a *preamble period*. Ideally, due to the 60 Hz sinusoidal nature of the power line, the index-wise average value of the equally sliced signal should exhibit high correlation compared to the preamble period if c is the exact SPP. Therefore, each device sweeps the value of c near $\frac{r}{60}$ in an iterative manner to find the accurate SPP that exhibits the highest correlation between index-wise averaged slices and the preamble period. Figure 4.5 illustrates an example of comparisons between a preamble period and the mean of 20 subsequent periods for $c = 1416$, 1419, and 1422. Clearly, among three averaged slices, the correlation with the preamble is highest when the length of the slices is equivalent to $c = 1419$. Once this procedure is executed on both devices, c_A and c_B is exchanged with each other. After the exchange, each device resamples its measured signal S_u by a common SPP c_l, typically the lower value between c_A and c_B.

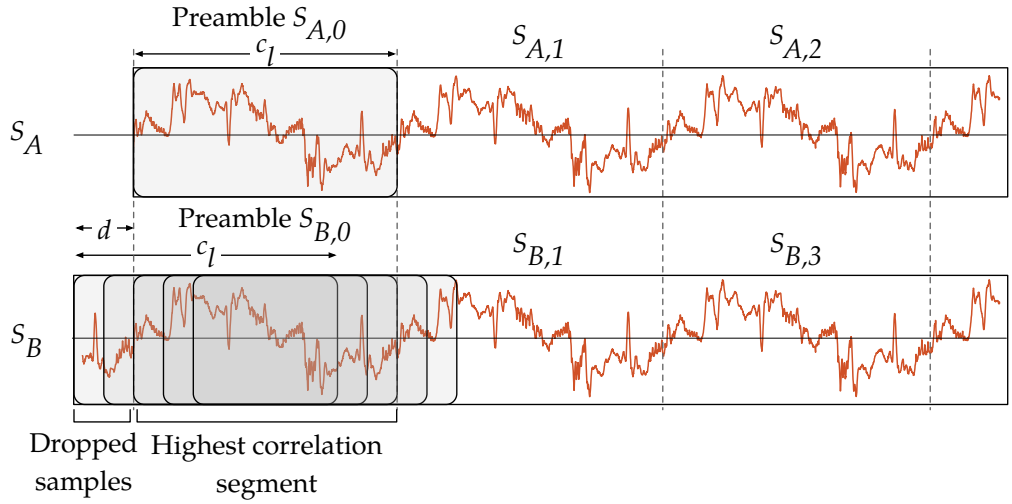


Figure 4.6: VoltKey's time synchronization. Using the sliding window approach, Device B locates the most correlated segment between received preamble $S_{A,0}$ and discards the samples up to the offset d.

Time Synchronization

After the sampling rate matching, both devices now have an equivalent SPP. However, two signal S_A and S_B exhibit lack of temporal alignment by an offset of d samples caused by the network latency during the transmission of the initiation message. That is, Devices A and B cannot start measurement at the exact same time. For example, on a WiFi network, a long latency up to a few milliseconds is common, which is significant considering the length of a single period (16.7 ms).

Considering existing solutions to establish accurate time synchronization such as GPS and atomic clocks are not feasible for low-cost IoT devices, VoltKey achieves low-cost high-precision time synchronization by exploiting the simultaneous measurement between two devices. First, Device A sends its preamble period $S_{A,0}$, which has a length of c_l samples to B. Once B receives A's preamble period, it uses a sliding window on S_B to

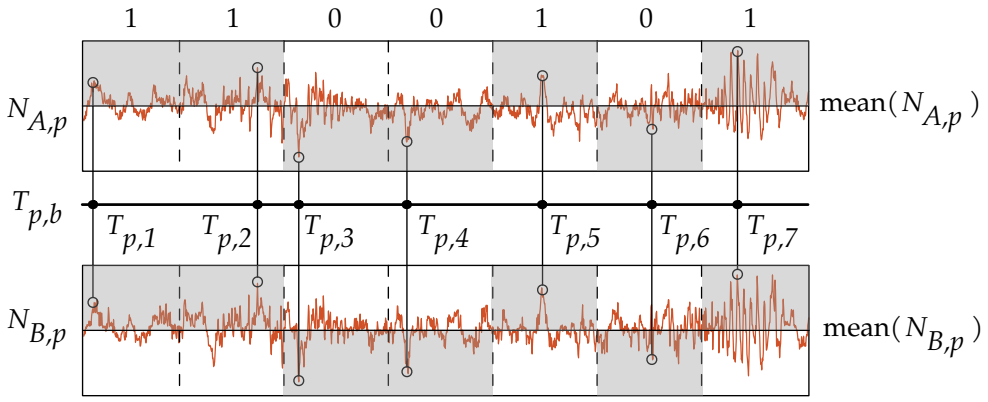


Figure 4.7: Bit sequence extraction from the p-th noise period with $n_b = 7$. The largest absolute value of each bin is converted to a bit 1 if indexed value at $T_{p,b}$ is greater than the mean of the noise period, and a bit 0 otherwise.

find the offset d that produces the highest correlation. To keep the leakage information minimal, the preamble period is solely utilized for time synchronization and not used for bit sequence extraction stage because this information is assumed to be eavesdropped by an adversary. Thus, both devices discard the samples up to the end of the preamble period. Figure 4.6 illustrates the synchronization process between two devices. After dropping the first d samples, S_A and S_B are accurately aligned with a common time base, so they can be sliced into synchronized periods $S_{A,p}$ and $S_{B,p}$, where p is the period number ($p = 0, 1, 2, \ldots, n_p$).

Bit Sequence Extraction

VoltKey exploits temporal randomness in the amplitude of S_u to obtain a random bit sequence. The dominant signal in S_u is a 60 Hz sinusoid plus its harmonics. This periodic waveform does not fluctuate with much randomness, so the goal is to remove the 60 Hz periodic portion of the signal before extracting entropy. The *noise period*, $N_{u,p}$, is defined as the noise

component that resides in each period. It is the period-to-period random variation, which is the index-wise subtraction result of two consecutive periods:

$$N_{u,p} = S_{u,p} - S_{u,p+1} \text{ for } p = 1, 2, \ldots, n_p. \tag{4.1}$$

VoltKey do not use the preamble period, $S_{u,0}$, since it is already publicly broadcasted during time synchronization. In order to extract multiple bits, each noise period is equally sliced into n_b bins, where each bin contains $\lfloor \frac{c_l}{n_b} \rfloor$ samples. First, Device A searches for the index of the sample with the maximum absolute value among all samples in each bin for every period, which is denoted by $T_{p,b}$, where $b = 1, 2, \ldots, n_b$ is the bin number. Then, a sequence of the indices, T, is shared with B through a public channel. With the common index sequence T from Device A, both devices can extract the same bit sequences by observing the value of the noise, $N_{u,p}(T_{p,b})$, at each index $T_{p,b}$. If $N_{u,p}(T_{p,b})$ is greater than the mean of the noise period, a bit 1 is extracted from the b-th bin of the p-th period; otherwise, bit 0 is extracted. That is, for $p = 1, 2, \ldots, n_p$ and $b = 1, 2, \ldots, n_b$, the bit $K_{u,p,b}$ is defined as:

$$K_{u,p,b} = \begin{cases} 1 & \text{if } N_{u,p}(T_{p,b}) \geqq \text{mean}(N_{u,p}) \\ 0 & \text{if } N_{u,p}(T_{p,b}) < \text{mean}(N_{u,p}). \end{cases} \tag{4.2}$$

Figure 4.7 illustrates an example of VoltKey's bit sequence extraction process. Thanks to the sampling rate calculation and synchronization procedure, two independently obtained noise periods from two devices exhibit a high correlation. The sequence of noise period is equivalently segmented into 7 bins (i.e., $n_b = 7$). The index of the maximum absolute value within each bin, $T = (T_{p,1}, T_{p,2}, \ldots, T_{p,7})$, is transferred to Device B. In Device B, if the value at these indices exceeds the mean of the noise period, the bit translates to a bit 1 (b = 1, 2, 5, and 7); if the value is less than the mean, the bit translates to a bit 0 (b = 3, 4, and 6). Even if the eavesdropper obtains $T_{p,b}$, without the noise periods from the power

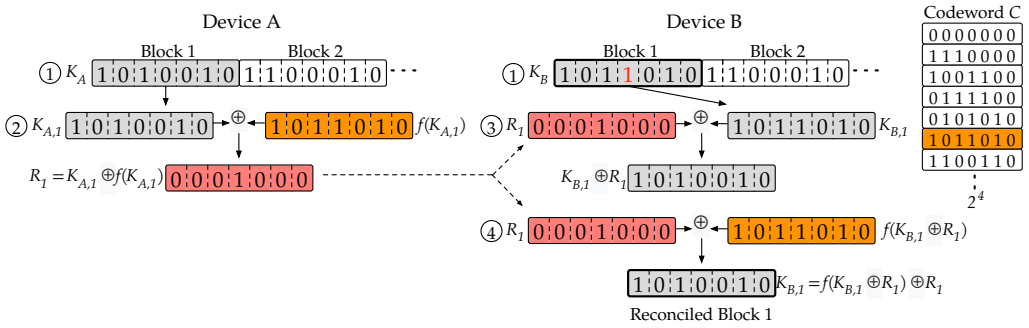


Figure 4.8: Illustration of key reconciliation process of the first seven bits using Hamming(7,4) code. ① Bit sequences extracted from both devices are divided into linear blocks of seven bits. ② Difference (exclusive or) between bits in the block and its corresponding codeword, denoted as R_1, is transferred to Device B. ③ Using R_1, Device B flips the bit differences with its own 7-bit block. ④ Result from the previous step is mapped to the codeword, and an additional bit flip with R_1 will reconcile the single-bit error between two devices. Subsequent blocks are reconciled in similar manner.

line he/she cannot properly obtain or predict the resulting bit sequence K. This bit extraction scheme is comparable to the list-encoding scheme used in [48]. However, VoltKey extracts the highest amplitude instead of finding the relative minima and maxima of the signal, which can be prone to signal misalignment. Even if the synchronization is not perfect between two devices, this technique reduces bit-wise error in the resulting bit sequences.

Key Reconciliation

The bit extraction protocol generates bit sequences that are nearly identical among nearby devices. But to serve as authentication or encryption keys, the bit sequences must have a 100% bit agreement rate. Even a single bit difference between two independently generated sequences will result

in encrypted messages that are undecipherable. In the case of VoltKey, small differences in the voltage noise pattern observed by two nearby VoltKey devices can result in occasional single-bit errors in the extracted bit sequence that render the resulting key useless for the purpose of authentication or encryption. The propensity of environmental noise to vary by the location that enables context-based key generation also creates spurious errors in extracted bit sequences. In order to use the extracted bit sequences for authentication or encryption, it must resolve bit errors first. Key reconciliation is a suite of techniques that allows a pair of remote devices to establish a common shared secret key through a public channel, starting from two similar bit sequences that may have a small proportion of bit errors. The base of the key reconciliation is based mainly on the error correcting code (ECC). For example, (n,k) ECC is based on a total of 2^k possible codewords of n-bit sequence, reducing the entropy of n bits by $n-k$ bits. I implement the *quantization-based construction* method presented in [48, 78].

Figure 4.8 illustrates the process of quantization-based construction using Hamming(7,4) code. Both devices use a public set of codewords $\mathcal{C}$, which consists of 16 (2^k) possible 7-bit sequences, known to all parties (even potential eavesdroppers). Let $f(b)$, where b denotes block number, be a publicly available function that maps the extracted 7-bit sequence to codeword in $\mathcal{C}$ in terms of the closest Hamming distance. ① Each device (A and B) begins by extracting a sequential block of 7-bit sequence from their extracted bit sequences (denoted K_A and K_B) from the measured mains voltage waveform. ② Device A computes $R_b = K_{A,b} \oplus f(K_{A,b})$ which is a 7-bit sequence in which each bit encodes whether there is a difference between the extracted bit sequence $K_{A,b}$ and its map in $\mathcal{C}$. Then, Device A sends R_b to Device B. ③ Afterwards, Device B uses its own 7-bit sequence, $K_{B,b}$, flips the bit differences using R_b. ④ Using function $f(b)$, the result from the previous step maps to the codeword. With an obtained codeword,

another bit flip operation is done with R_b, which will result in $K_{A,b}$ with high probability. Even if the eavesdropper obtains the R_b, without the extracted bit sequence K_A or K_B, the eavesdropper cannot derive the reconciled key without the knowledge of n-bit codeword. Since there are only 16 possibilities of $\mathcal{C}$, the resulting entropy of each block is worth only 4 bits. Although any other ECC is suitable for quantization-based construction, for simplicity, VoltKey will use two different sets of Hamming codes (i.e., Hamming(3,1) and Hamming(7,4)) as a mapping function between n-bit and the k-bit codeword [64].

Key reconciliation presents a unique problem when applied to VoltKey and other ZIA schemes. It allows a nearby imposter who does not have physical access to the authenticated electrical domain to measure a voltage noise signal that is similar to the benign waveform and try to authenticate itself to gain access to the network. Alternatively, the imposter could just pick a random bit sequence and exploit key reconciliation to transform it into the correct key. In order to avoid this exploit, VoltKey must limit the number of bit errors that are allowed to be corrected by key reconciliation. In the reconciliation scheme presented above, this can be tuned by adjusting n and k, trading off security for reliability.

4.1.3 Implementation and Evaluation

Experimental Setup and Metrics

The voltage measurements of S_u from the MCU's ADC, programmed at 85.4 kSPS using an internal oscillator, are stored on the onboard RAM and transferred to the PC via the MCU's serial interface. I use the low-power internal oscillator as main clock generating source not only because it is a common setup in low-cost IoT devices but also in order to intentionally induce natural sampling rate variation between multiple devices. I simulate bit sequence extraction and information exchange between devices

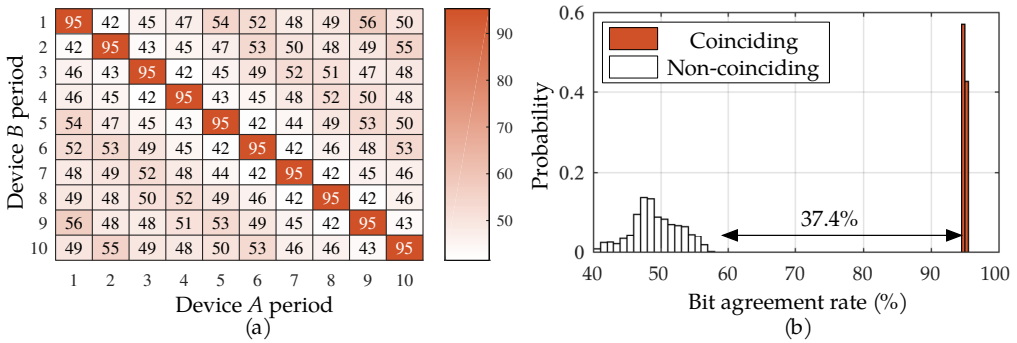


Figure 4.9: (a) 10-by-10 confusion matrix of average bit agreement rate between bit sequences generated by noise periods obtained by Device A and B. (b) Distribution of bit agreement rate between diagonal and off-diagonal pairs of noise periods.

using software written in Matlab. In order to evaluate the *bit agreement rate* in the generated bit sequences between two devices, I extract 128-bit long sequences with n_b=6, 8 and 10 bins, from n_p=22, 16 and 13 periods, respectively. Because authentication by comparing two generated keys are considered successful only when bit-wise errors between two keys are zero after the reconciliation process, I define *authentication success rate* to be a percentage of key pairs with a bit agreement rate of 100% out of all pairs of generated keys.

Uniqueness of Bit Sequences

In order for VoltKey to be a reliable authentication method, it is important that two bit sequences generated by two different devices belonging in same authenticated electrical domain exhibit high bit-wise agreement rate. More importantly, temporally unique bit sequences should exhibit low bit agreement rate compared to other bit sequence generated by different noise periods at different time. To investigate the uniqueness of bit sequences generated by each noise period $N_{u,p}$, I set up two VoltKey devices,

connected to two colocated outlets that are less than 10 cm apart, to periodically gather 10 consecutive noise periods (n_p=10) under regular daily office environment with typical usage of various surrounding electronic loads such as PCs, light stands, microwaves and refrigerators. In order to obtain uniform data throughout all day period, the data measurement process lasted three consecutive days, resulting in a total of 864 sets of $S_{u,p}$ from both Device A and B. As by the protocol, the starting index of each noise period is obtained from the agreed length of c_l samples and each noise period is set to generate 6-bit long sequences (n_b=6). The similarities of bit-sequences are presented as the rate of matching bits between two resulting sequences, or the bit agreement rate between Devices A and B. Note that in this evaluation, key reconciliation is not considered, and therefore the bit agreement rates are not expected to reach 100%.

Figure 4.9(a) illustrates a 10-by-10 confusion matrix of average bit agreement rates between bit sequences generated by Devices A and B for 10 periods. The diagonal elements represent bit agreement rates between coinciding bit sequences (generated from the same periods), whereas the off-diagonal elements represent rates between non-coinciding bit sequences (generated from different periods). It is clear that the average bit agreement rates between coinciding bit sequences are consistently around 95%; on the other hand, the bit agreement rates between non-coinciding bit sequences are close to 50%, which is equivalent to a rate of a random guess. Figure 4.9(b) shows the distribution of the bit agreement rates. The distance between the bit agreement rates of coinciding and non-coinciding bit sequences is 37.4%. The result represents that each noise period and a bit sequence generated from it are unique. It also demonstrates that VoltKey can accurately pinpoint the starting index of each noise period using common sampling rate c_l, thanks to the effective sampling rate matching and time synchronization procedures. It also contributes to the strong security of VoltKey by allowing it to use each key only once. Even

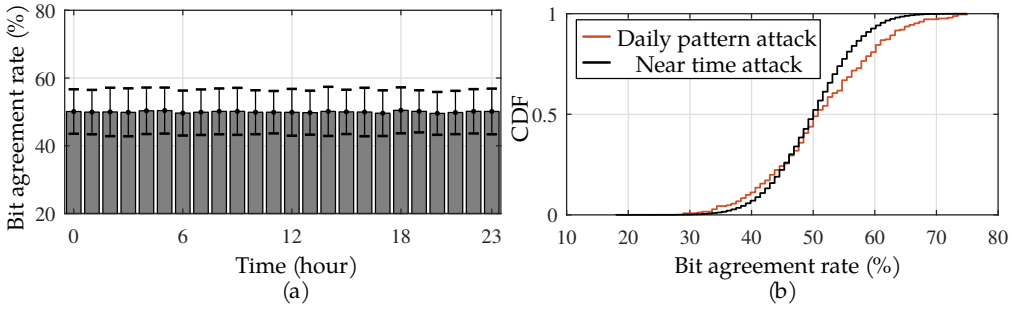


Figure 4.10: (a) Bit agreement rate between all keys pairs generated within each hour over course of three consecutive days. (b) CDF of daily pattern and near time attack.

if an adversary is able to measure the noise at one time instance, the key generated from the measured noise is not stronger than a random guess from the very next moment.

Time-based Attack

An adversary might attempt to exploit that electrical load usage has a repeated pattern in order to generate a key from a previously observed power line noise. This might lead to a malicious attacker who gets the hold of a single recently used key, trying to authenticate themselves at a near time within the same hour. This attack scenario is defined as *near time attack*. Since the attacker is equipped with a directional antenna and is able to eavesdrop plaintext packets during the initiation message, they are able to go through the reconciliation process with a single key that has already been used within the same hour. In order to validate VoltKey's robustness against near time attack, I gather a total of 864 keys (128-bit with n_b=6), from a single device over the course of three days and categorize the keys based on its extracted hour. This results in an average of 36 keys within each hour category. Afterward, all possible reconciled key pairs are evaluated on their mean bit agreement rates as shown in

Figure 4.10(a). Clearly, the agreement rate between keys generated within each hour category is consistently achieving close to 50% agreement rate after the reconciliation stage, which is close to the rate of a random guess. The distribution of the key agreement rate from near time attack is shown in Figure 4.10(b). As illustrated, the distribution of the bit agreement rate among all pairs within a single hour time period exhibit binomial distribution centered at 50.0% and the maximum agreement rate that the attacker can achieve is 74.2%. Therefore, malicious attacker getting hold of any recently used key cannot properly authenticate themselves near future (i.e., within an hour) by re-using the old key.

Additionally, two keys gathered at the same time on different days should be different to prevent malicious users from obtaining the key at one time and reusing the key later to authenticate themselves at the similar time on a different day. This attack scenario is referred to as *daily pattern attack*. In order to simulate daily pattern attack, I configured a single VoltKey device to extract a key every five minutes over the course of 2 consecutive days and used the key generated from the first day to authenticate itself against key gathered at the exact same time (hour and minute) on the second day. The resulting distribution of bit agreement rate of daily pattern attack is illustrated in Figure 4.10(b). Compared to distribution of near time attack, daily pattern attack shows slightly higher bit agreement rate compared with legitimate key because the device usage at the exact time of the day may not change much from previous days. However, the attacker was only able to achieve a mean agreement rate of 51.3% with the highest agreement rate of 75.0%, which demonstrates VoltKey's robustness from an active adversary obtaining previously used keys. This suggests that even under the circumstances of similar electrical usage pattern, VoltKey harvests different enough bit sequences from the voltage noise to prevent adversaries from carrying out timing-based attack scenarios.

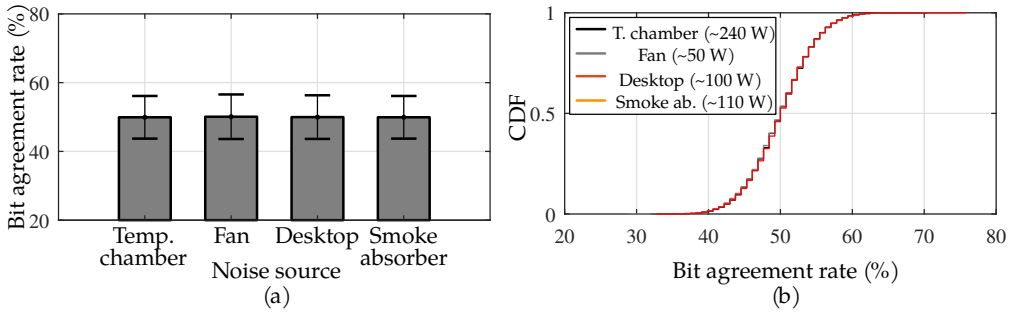


Figure 4.11: (a) Bit agreement rate between all keys pairs generated with nearby inductive electrical loads. (b) CDF of dominant noise attack using different loads.

Robustness against Dominant Noise

Loads from motors and nonlinear circuit elements that are present in the nearby power circuit may be the source of continuous dominant noise due to electro-mechanical switching from the motor brushes, rectification, etc. One might wonder if VoltKey is robust enough to generate different keys while the same set of dominant electrical appliances is operating. This is an important question because malicious agent should not be able to generate keys by creating artificial electrical noise by running the same appliances as the target's authenticated electrical domain. This attack scenario is referred to as *dominant noise attack*. First, to verify the differences in generated keys under same electrical environment, I set single VoltKey device to harvest 100 keys in the presence of four different types of high-wattage laboratory equipment operating nearby: temperature chamber, fan, desktop computer, and smoke absorber. Then, the distribution of the bit agreement rates are compared for each load. As illustrated in Figure 4.11(a), the mean bit agreement rate between all key pairs generated under the same dominant electrical loads show a mean value of 49.9%. In addition, all generated keys exhibit no overlap among all possible pairs. To simulate a dominant noise attack, I set another VoltKey device to generate 100 sets

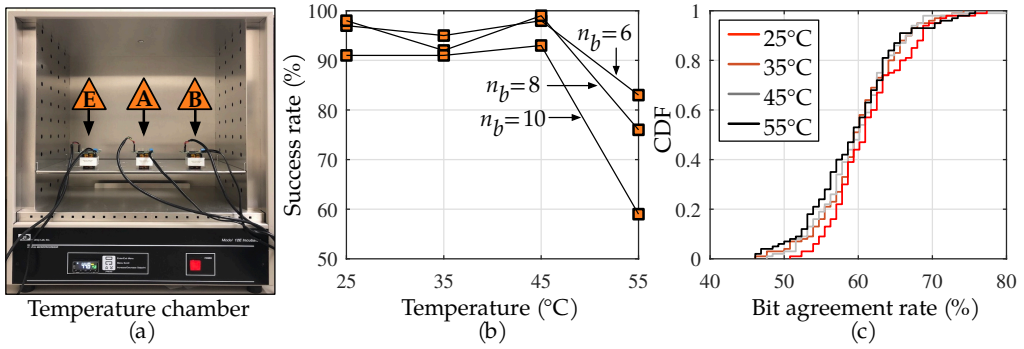


Figure 4.12: (a) Experiment setup inside temperature chamber. (b) Success rate between legitimate devices with respect to different operating temperature. (c) CDF of passive attacks with different temperature.

of keys with an identical nearby source of dominant noise at different times under different authenticated domain (nearby room at least 10 m apart). The CDF of bit agreement rate of 10,000 key pairs are illustrated in Figure 4.11(b) with power ratings of four different appliances. Clearly, the CDF of resulting bit agreement rates for all four different dominant electrical loads exhibits similar binomial distributions with a mean of around 50.0% and highest of 70.3% bit agreement rate. The malicious device was not effective in reproducing the legitimate key. Additionally, the dominant noise attack shows slightly less mean bit agreement rate compared to two time-based attacks from the previous section. This is due to the fact that even though the keys are obtained under single nearby dominant electrical loads, the general noise fingerprint of each authenticated electrical domain differs due to many other influential factors (random electromagnetic interference, geometry of power lines, etc).

Robustness against Temperature Variation

The frequency of the MCU's on-chip oscillator, used to time sample acquisitions from the ADC, is temperature-dependent [52]. Additionally,

the ADC's internal voltage reference output also depends on the temperature. Because VoltKey uses low-cost MCU with an internal oscillator, the temperature of the device can heavily affect the sampling rate, possibly resulting in the generated keys to be irreconcilable. To understand the effectiveness of VoltKey under different operating temperatures, I set multiple VoltKey devices inside the temperature chamber to attempt to authenticate itself with each other. Two Devices, A and B, are set to draw power from the colocated outlet in the same authenticated electrical domain and a single malicious Device E is connected to the outlet two rooms away which is separated with the distance of more than 30 m. Device E simulates a malicious device from the outside of authenticated electrical domain attempting to authenticate itself with the same temperature as the trusted devices under voltage readings from a nearby location within wireless range. This attack scenario is referred to as *passive attack*. Note that unlike previous attack scenarios, the timestamp of the key extraction synchronizes with the trusted devices. The experiment setup is illustrated in Figure 4.12(a). Each device pair (A-B and A-E) attempts to authenticate 100 sets of 128-bit long keys with n_b=6, 8 and 10 under four different temperatures of 25, 35, 45 and 55 °C. To ensure all devices to reach the specified temperature, devices are placed inside the temperature chamber for 30 minutes after each temperature adjustment.

Figure 4.12(b) illustrates the success rate of legitimate devices with respect to different temperatures. At 25 to 45 °C, the success rate of authentication attempt remains over 90% with average bit agreement rate greater than 91% for all n_b. However, as the controlled temperature reaches 55 °C, the success rate significantly decreases to under 85% for all n_b. Specifically, when n_b=10, only 59% of the total attempts were successful. This is due to the MCU's internal oscillator's drift, causing the ADC to intermittently measure 0 reading samples. The result of the passive attack with controlled temperature is illustrated in Figure 4.12(c). At 25 °C, the malicious device

can achieve an average of 60% of the trusted key by attempting to authenticate itself with a nearby power line. As the temperature increases, the oscillator's drift on malicious devices hinders its performance, reducing the average agreement rate down to 59.2%. Compared to previous attacks leveraging time and noise source, passive attack under room temperature exhibits higher bit agreement rate of up to 78% with a mean of 61.3% due to an exact timestamp of the key extraction synchronizing with legitimate devices.

Distance and Range

Ideally, the authenticated electrical domain should be limited to a certain space within the user's trust domain such as a home or office with physical access restrictions. Therefore, it is crucial that the superimposed noise signal is spatially unique, as a function of the distance between two authenticating devices. To validate the statement, I set four VoltKey devices to authenticate with a single access point under varying distance within a realistic laboratory environment. As illustrated in Figure 4.13(a), five devices (A, B, C, D and E) are drawing power from five different wall outlets at increasing distances. The power line attached to the outlet is clearly visible around the room as illustrated with the black line. However, to experiment with even further distance, I extend Device D and E further from the outlet with extension cords, resulting in non-equivalently increasing distance from 1 m up to 24.8 m, between (A-B, A-C, A-D and A-E) device pairs. The lab environment is under regular daily usage with electronic appliances such as smoke absorber, heat gun, personal computers and soldering stations. Similar to previous experimental settings, each device pair attempts to authenticate itself with Device A, periodically regenerating keys every five minutes for three consecutive days, resulting in a total of 864 sets of keys.

Figure 4.13(c) illustrates the bit agreement rate against the distance

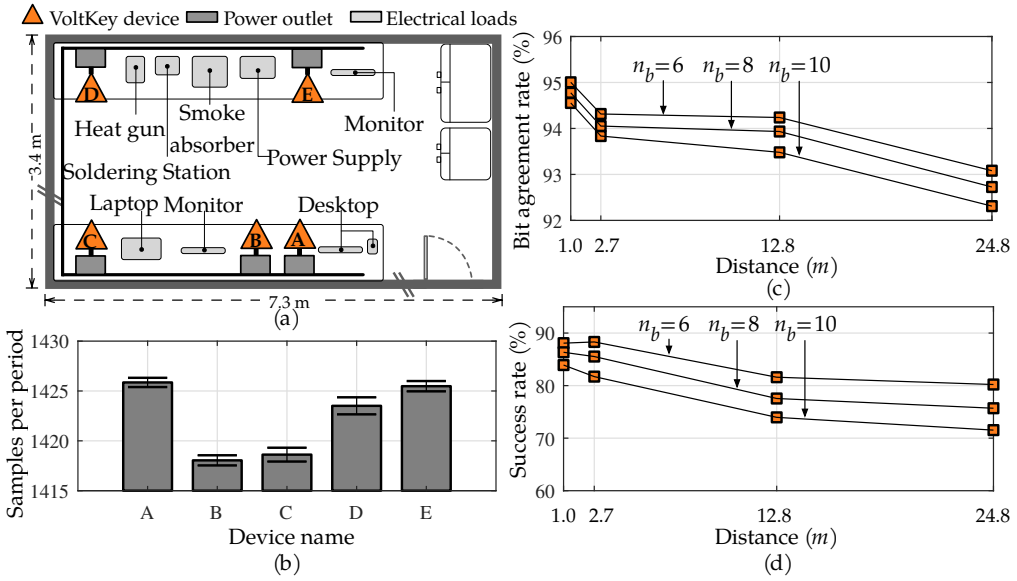


Figure 4.13: (a) Location and distance between multiple VoltKey devices (not to scale). The power line is visible around the surrounding wall of the lab. The electrical distance from Device A to B, C, D and E is 1, 2.7, 12.8 and 24.8 m, respectively. (b) SPS of five different devices. (c) Bit agreement rate between devices with respect to the distance between authenticating devices. (d) Success rate of authentication attempts with respect to distance between authenticating devices.

before performing key reconciliation. As the distance between the two authenticating devices increases, the bit agreement rate decreases. Specifically, when the two devices are in close proximity of 1.0 m apart, the bit agreement rate for of 128-bit long keys with n_b=6, 8 and 10 are 95, 94.7 and 94.5% respectively. On the other hand, when the distance increases up to 24.8 m, the bit agreement rate gradually decreases up to 93.1, 92.7 and 92.3% for all n_b. This indicates that at some point, there will be a distance that will decrease the agreement rate so that two authenticating devices will not be able to reconcile the bit differences. As Figure 4.13(d) illustrates, the success rate, which is the rate of authentication trial that results in a

perfectly matching key after key reconciliation stage, decreases with distance. In specific, for devices that are located 1.0 m apart, the success rate exhibits 88.1% with n_b=6. As the distance between the devices gradually increases to 24.8 m, the success rate significantly decreases down to 80%. Additionally, I find that for each distance, there is a trade-off between the amount of entropy, n_b, and the amount of bit agreement among nearby VoltKey devices in the authenticated electrical domain. The consequence of this observation is that if VoltKey wants to extract more bits within a single noise period—which is good for encryption strength—it has to sacrifice the bit agreement rate, which reduces the overall success rate of authentication attempts. Additionally, as the distance between authenticating devices increases, noise in the power line is translated into different bit sequences, which proves the spatial uniqueness property of the key generation algorithm. Furthermore, to illustrate the effectiveness of VoltKey under varying sampling frequency, Figure 4.13(b) shows the sample count per period that is exchanged between devices. Although the sampling rate of the MCU is programmed at an identical fixed frequency, the high sampling rate and imperfections of internal oscillator resulted in devices to sample at higher or lower frequencies compared to the programmed rate. However, despite each devices varying sampling rate, thanks to the accurate sampling rate estimation and matching procedure, authenticating devices are able to achieve high bit agreement rate which ultimately leads to a high overall success rate.

Typical commercial and residential buildings have multiple outlets that are connected to different circuit breakers to protect individual electrical systems from current overloads and short circuits. To investigate circuit breakers' effects on the radius of the authenticated electrical domain, additional experiments are conducted under identical distance settings (1.0 m to 24.8 m) with a single circuit breaker attached between pair of authenticating devices. Figure 4.14(a) illustrates 120 V–15 A standard D type

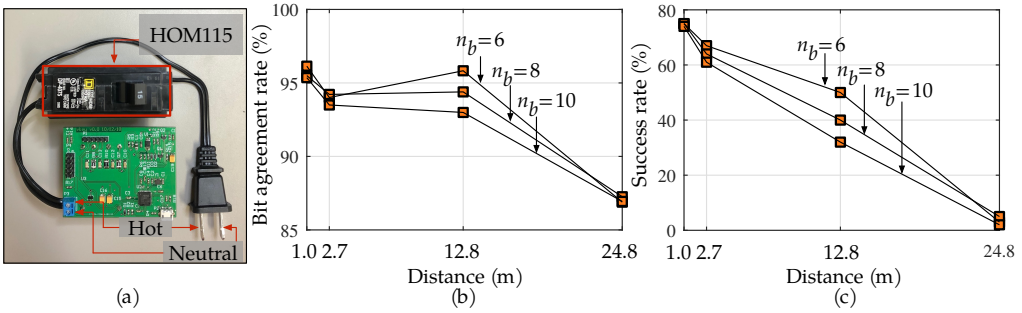


Figure 4.14: (a) VoltKey prototype with circuit breaker attached on the hot line of power cable. (b) Bit agreement rate between devices with respect to the distance between authenticating devices. (c) Success rate of authentication attempts with respect to distance between authenticating devices.

miniature circuit breaker (Schneider Electric HOM115) installed between hot lines of VoltKey's power cable [28]. For the experiment, Device B (with circuit breaker) is relocated to varying distance while Device A is fixed to stationary location as shown in Figure 4.13(a). For each distance, 100 sets of keys are generated from each device under the duration of 3 hours. As illustrated in Figure 4.14(b), the bit agreement rate between two devices is maintained above 94% when the distance is at 1.0 m apart. Up to 12.8 m, devices separated with circuit breakers exhibit high bit agreement rate similar to the agreement rate without the circuit breakers. However, as the distance increases up to 24.8 m, the agreement rate significantly decreases to less than 90%. Consequently, the success rate at 24.8 m is 3, 5 and 2% for n_b=6, 8 and 10, respectively as shown in Figure 4.14(c). This is due to the fact that circuit breaker acting as a low pass filter that suppressed the leakage of high-frequency noise between the devices. The results suggest that the radius of an authenticated electrical domain does not reach beyond a few tens of meters in the presence of a circuit breaker between two VoltKey devices. The result suggests that contextual separation between

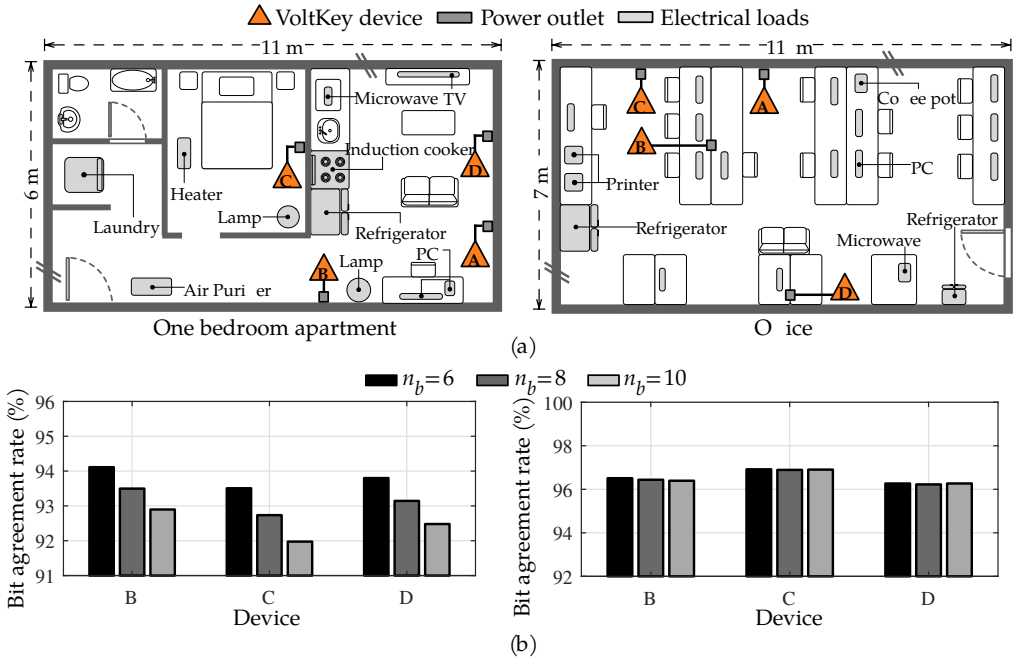


Figure 4.15: (a) Floor plans of the one-bedroom apartment and office (not to scale). VoltKey devices are connected to different wall outlets to periodically authenticate themselves with Device A. (b) Bit agreement rate of devices with different n_b before key reconciliation.

authenticated and non-authenticated electrical domain is separated with certain distance and presence of circuit breakers in between.

Realistic Deployment

To verify the overall effectiveness of VoltKey under realistic deployment scenarios, four devices are deployed within a regular daily environment to measure the success rate and the bit agreement rate of each device. I also simulate a reasonable attack scenario by placing a single VoltKey unit in the next room, outside the authenticated electrical domain, continually attempting to authenticate itself with the legitimate device. This adversarial device simulates a passive attack, periodically trying to authenticate

itself using the voltage readings from the nearby room within wireless range. Two separate experiments are conducted in a typical one-bedroom apartment and in an office environment. Figure 4.15(a) illustrates two deployment environments. In the one-bedroom apartment scenario, a single device is connected in the bedroom and three other devices are connected to various outlets spread around the living room. The adversarial device is located outside the apartment constantly drawing power from an outlet located on the apartment hallway. In the office environment, four trusted devices are spread around the room, surrounded by personal computers and various household electronics such as refrigerators and microwaves. A single adversarial device is located in the lab two rooms down the hall. Significant loads that are constantly being power cycled in the course of daily usage are marked in addition to multiple VoltKey devices and their associated power outlets. Four VoltKey devices (B, C, D, and E) are set to periodically authenticate themselves with Device A every 10 minutes over the course of six consecutive days. To increase the overall resulting success rate, each device at every key generation cycle (10 minutes) is allowed a maximum of five authentication attempts. The bit agreement for different devices before key reconciliation using different n_b is illustrated in Figure 4.15(b). For Device B, C and D, located in the one-bedroom apartment, the bit agreement with n_b=6, 8 and 10 exhibit 94.1, 93.5 and 93.8%, respectively. As n_b increases to up to 10, devices experience slightly lower agreement rate due to the higher number of harvested bits under a single noise period. On the other hand, bit agreement rate for device deployed in office environment exhibit higher rate compared to that of a one-bedroom apartment, achieving 96.2, 97 and 96.1% for Device B, C and D, respectively. As n_b increases to up to 10, devices do not experience significant lower agreement rate. This is due to that the greater number of switching activities of electronic appliances in the apartment leading to higher fluctuation on the voltage signal, leading

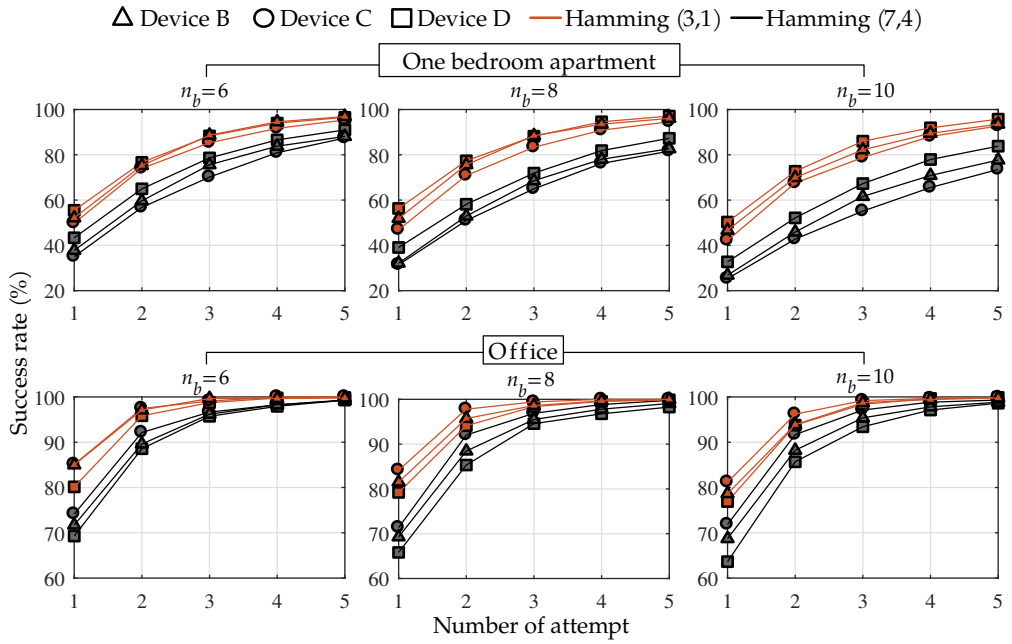


Figure 4.16: Successful authentication rate with multiple trials of authentication in apartment and office environment.

to inaccurate bit agreement.

Figure4.16 illustrates success rate of each device under different Hamming codes and n_b values. Compared to Hamming(7,4) error correcting code, Hamming(3,1) achieves higher error correcting capability due to higher rate of overhead bit (66%), resulting in a higher overall success rate. Specifically, in one-bedroom environment with n_b=6, the success rate of a single trial attempt is 55, 49 and 52% for Devices B, C and D, respectively. As devices are allowed up to five authentication attempt, the success rate increases up to 90.9, 87.4 and 99.1%. Device B, which is the closest to Device A compared to other devices, achieves the highest success rate as expected. On the other hand, Device C, located in the bedroom, achieves lowest success rate with 87% success rate due to the long electrical distance between two devices. When the overhead bit ratio decreases to 43% using

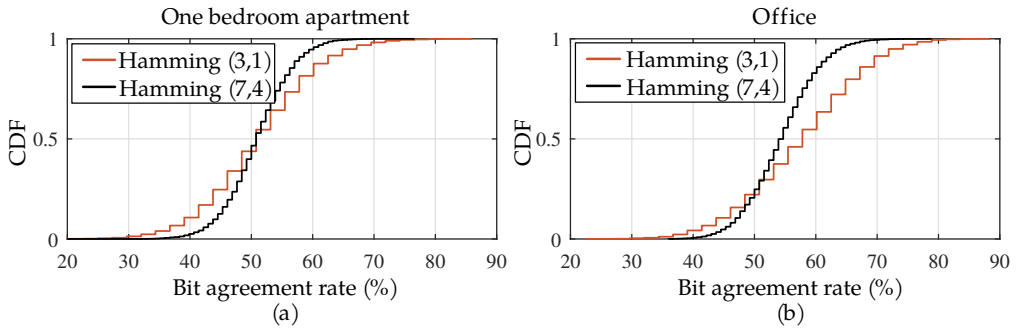


Figure 4.17: (a) CDF of bit agreement rate for passive attack (n_b=6) on (a) one-bedroom apartment and (b) office.

Hamming(7,4), the success rate of a single trial attempt is 43, 35 and 37% whereas allowing five attempts resulted in 90, 87.3 and 88% for Device B, C and D, respectively.

VoltKey deployed in the office environment exhibits higher success rates due to higher bit agreement rates. For n_b=6, three devices under Hamming(3,1) reconciliation show over 80% success rates at a single trial. As the trial attempt increases to up to five, devices exhibit success rate of 99.8, 100 and 99.7% for Device B, C and D, respectively. With the usage of Hamming(7,4) protocol, authentication was more selective, but with five attempts, devices achieve 99.3, 99.3 and 99.1% for Device B, C and D, respectively. As the number of harvested bits from a single noise period increases to 10, the success rate still remains relatively high for all devices with rate of 98.6, 99.3 and 98.8%. Overall, VoltKey shows its effectiveness both in office and home environments with a success rate of over 90% for all devices.

The results of the passive attack in the apartment and office with n_b=6 are illustrated in Figure 4.17(a) and (b), respectively. Because the apartment environment is more selective in authenticating devices with higher context separation, a malicious device with voltage readings from outside the unit is only able to achieve on average 50.9% of the key with using

Hamming(7,4) error correction based reconciliation. Moreover, utilizing Hamming(3,1) reconciliation results in a similar mean bit agreement rate of 51.1%. However, the maximum bit agreement rate that can be achieved by the malicious device is much higher using Hamming(3,1), with a rate of 85.9%. In the office environment where context separation is lower, the malicious device exhibit higher mean agreement rates than that of the one-bedroom apartment. Specifically, mean agreement rates of 57.4% and 54.3% are achieved with Hamming(3,1) and Hamming(7,4), respectively. Furthermore, with Hamming(3,1), the adversary is able to achieve the highest rate of 88.2%. Overall, out of all passive attempts, none of the malicious devices under any environment successfully authenticated itself with a measured voltage signal from outside of the authenticated electrical domain within trusted WiFi range which demonstrates VoltKey's security against various malicious attacks.

Discussion

I have demonstrated that VoltKey is practical in a variety of electrical environments. Key generation in all the environments is reliable enough to show that heterogeneous IoT devices can use VoltKey to authenticate to an access point with no involvement from the user. It also shows that VoltKey can be implemented on low-cost hardware that can conveniently communicate with its host (either an IoT device or a WiFi access point) through a standard USB interface. In the following, I discuss practical challenges and concerns for deploying VoltKey en masse in more detail.

Duration of Authentication The duration of the authentication is directly proportional to the n_b and the type of Hamming(n,k) being used for reconciliation stage. Based on the experimental data from using Hamming(3,1) error correcting code, adversarial devices are able to obtain up to 85.9% and 88.2% of correct bits in the apartment and office envi-

Table 4.1: NIST test results of VoltKey (p-value $\geqq 0.05$)

NIST test	p**-value**
Frequency	0.7399
Block frequency	0.1223
Cumulative sums	0.5341
Rank	0.3504
Non overlapping template	0.3505
Linear complexity	0.7399

ronments, respectively. Consequently, from an adversary's point of view, the resulting entropy of VoltKey is only 0.14 and 0.11 bits. According to [51], the minimum entropy for an authentication token is 20 bits and 128 bits for a cryptographic key. Considering Hamming(n,k) error correcting code loses $n-k$ bits for every n bits in entropy, VoltKey needs to extract minimum of 60 bits and 384 bits with Hamming(3,1) to be used for authentication tokens and cryptographic keys, respectively. Additionally, considering maximum entropy loss from the adversary, in the one-bedroom apartment environment, to obtain 60 bits, $\lceil \frac{60}{0.14} \rceil = 429$ bits need to be extracted and for the office environment, 546 bits are needed. Accordingly, with $n_b = 6$, it takes $\frac{429}{60 \cdot 6} = 0.119$ s and 0.151 s worth of voltage signal measurement to authenticate in the one-bedroom apartment and office, respectively. To be used as a cryptographic key, $\lceil \frac{384}{0.14} \rceil = 2743$ bits and 3491 bits are required which results in 0.762 s and 0.970 s worth of voltage signal for home and office deployments. Overall, even under the circumstances of allowing multiple authentication iterations and considering the computation time of MCUs, the authentication does not require a significant amount of time.

Key Randomness In order to validate the randomness of the harvested bits, I applied a statistical test suite provided by National Institute of Stan-

dards and Technology (NIST) to the 850,000-bit long bitstream generated by VoltKey [4]. The test results are presented in Table. 4.1. Out of 15 total randomness tests, the generated bitstream passed 6 tests with p-value greater than the threshold (generally 0.01 or 0.05). If the test does not meet the threshold for a specific test, the randomness hypothesis is rejected, and the bitstream is presumed to have too much structure to be considered as a cryptographic key. In order to be considered as a true random number generating source, all p-values should be uniformly distributed across the range between 0 to 1. Therefore, further investigation is needed in order for VoltKey to be considered as a true random bit generating source.

Authentication Radius In the experiments, I found that the reliability of key reconciliation for devices inside the authenticated electrical domain varied by the environment (i.e. circuit breakers). Depending on the deployment scenario, users may want to adjust the permissiveness of key reconciliation to control the dimensions of the authenticated electrical domain. The most obvious technique is to modify the Hamming code used during key reconciliation. More permissive codes—those that allow authentication from bit sequences with higher error rates—would result in larger authenticated electrical domains at the expense of diminished security. Through extensive experiments with VoltKey, working with Hamming(7,4) and Hamming(3,1), the practical authentication radius is one or two rooms. Another technique that can increase the radius of the authenticated electrical domain would be having multiple VoltKey access points in different rooms of the structure to authenticate all devices within the building. This is an important practical problem for ZIA methods that do not exhibit strict context separation boundaries.

Minimum Sampling Rate Because VoltKey uses the ADC on a low-cost MCU to sample the voltage signal, the sampling rate directly affects the bit agreement rate before key reconciliation. The higher the sampling

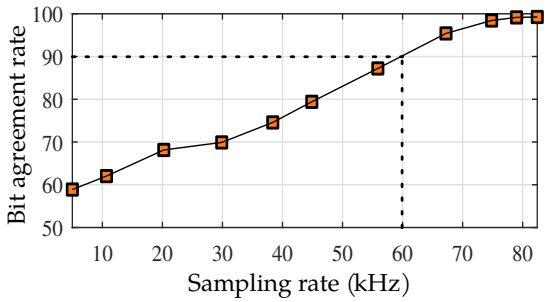


Figure 4.18: Bit agreement rate of bit sequences generated by two colocated VoltKey devices with respect to different sampling rate.

rate, the more precise its measured voltage signal, and the more accurate the bit extraction between two devices. However, after a certain accuracy threshold, the bit agreement rate will reach its upper limit. Generally, bit agreement rate of above 90% resulted in a reasonable success rate with multiple authentication iterations (less than 5). To investigate minimum sampling rate with 12-bit resolution ADC that can lead to high bit agreement rate, I downsample the measured signal from 80 kSPS to 5 kSPS to find out minimum sampling rate required to maintain high bit agreement rate. Figure 4.18 illustrates bit agreement rate between pair of generated bit sequences from two colocated (less than 10 cm apart) VoltKey devices with respect to downsampled frequency. At 82 kSPS, the bit agreement is maintained at 99.2% bit agreement rate. This suggests that the sampling frequency above 82 kSPS will not significantly increase the overall success rate of VoltKey. Starting at 60 kSPS, the bit agreement rate falls to below 90%. Therefore, to maintain bit agreement rate above 90% before key reconciliation, the minimum lower bound sampling frequency should be kept above around 60 kSPS which can easily be achieved with typical low-cost MCUs.

4.1.4 Conclusion

This Section presented VoltKey, an unobtrusive and transparent key generation method based on spatiotemporally unique noise patterns in the commercial power line. Because VoltKey involves no human effort during key establishment, VoltKey-enabled devices can autonomously and periodically update the network authentication key, significantly reducing the attack window and increasing usability in case of key leakage. Devised techniques address practical challenges in implementing VoltKey on low-cost IoT devices, and a hardware prototype was implemented to evaluate them. In the experiments, a high bit agreement rate over 95% is achieved even before key reconciliation, thanks to the precise sampling rate estimation and matching techniques. Under various realistic deployment scenarios in home, laboratory, and office environments, VoltKey successfully authenticate over 90% of trusted pairs of devices within reasonable authentication trial. It is also shown that VoltKey successfully rejects adversarial devices in different attack scenarios leveraging various temperature, time, dominant electrical noise, and access to nearby locations. Attached on ubiquitously available USB chargers and power supplies, VoltKey will allow multiple heterogeneous IoT devices to authenticate each other securely and seamlessly.

4.2 AeroKey: Using Ambient Electromagnetic Radiation for Secure and Usable Wireless Device Authentication

This Section describes the work named AeroKey, which represents a novel ZIA scheme for establishing secure wireless networks of personal devices

using *ambient electromagnetic radiation* (*EMR*)[1] as a source of randomness. AeroKey extracts entropy from low-frequency ambient EMR generated by surrounding electrical appliances and power lines that are ubiquitous indoors. I observe that the EMR noise tends to be spatially correlated only inside a small area of a few meters in radius yet temporally uncorrelated, making it ideal for authenticating spatiotemporally co-located devices in a personal area network. A pair of AeroKey-enabled devices first independently measure the ambient EMR using readily available ADCs without costly hardware-assisted signal conditioning. Then, from digitized sequences of the EMR measurements, two devices generate bit sequences which will be identical only within a small area called *personal authenticated region* (*PAR*), around the user. The two devices can make use of this identical bit sequences to form a basis of a symmetric key to authenticate each other.

AeroKey can potentially be used to enhance the usability in almost all inter-device authentication scenarios within indoor environment. An example use case would be during the authentication process between the two headless devices (with no graphical user interface or other peripherals). For instance, two AeroKey-enabled Raspberry Pis can authenticate each other by simply placing them in a close distance without the use of external monitor and keyboard. Also, relatively minimal hardware requirements of AeroKey allows it to be implemented in many small mobile or personal devices with MCU to facilitate seamless authentication between devices (i.e., associating wireless earbuds or smart pens with smartphones).

The main contributions of this work are summarized as follows:

- I present a secure and usable device authentication scheme named

[1]The radiation in this context refers to non-ionizing radiation at frequencies below visible light (e.g., super and ultra low frequency ranges), contrary to the high-energy ionizing radiation (e.g., X-rays, Gamma rays).

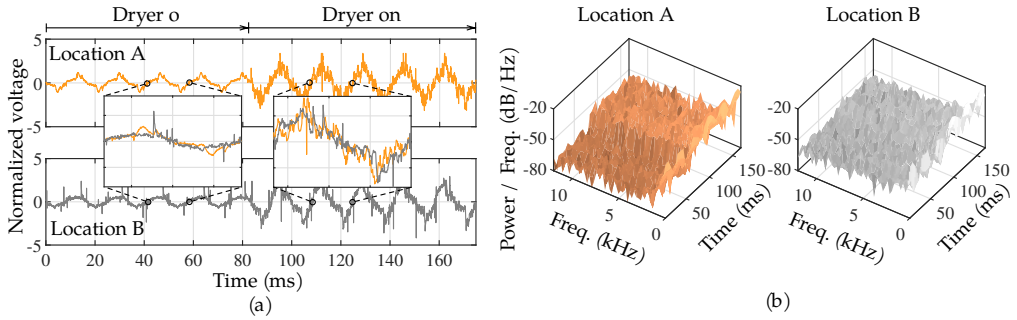


Figure 4.19: (a) Raw ADC readings from two different locations (5 m apart) as a hair dryer (1875 W) switches on at 80 ms. (b) Spectrograms of the raw readings from two different locations.

AeroKey, leveraging spatially correlated and temporally uncorrelated randomness in simultaneously measured ambient EMR using low-cost hardware.

- I propose and implement essential techniques (i.e., signal processing and noise extraction) to derive symmetric keys from EMR noise and to overcome the challenges in instrumenting AeroKey on low-cost MCUs, such as lack of time synchronization, low-precision ADC measurement, and inconsistent sampling rates.

- The real-world experiments using commercial off-the-shelf MCUs in various environments demonstrate reliable authentication rate between co-located devices and its robustness against various realistic adversaries with low EER of 3.4% or less and authentication time of under 24 s.

Ambient Electromagnetic Radiation

I first discuss the underlying physical basis of AeroKey. A power line that delivers power to appliances, lighting, and other electrical loads is essentially a charged wire carrying an alternating current (AC) and radi-

ates electromagnetic waves, which is called ambient EMR. The combined effect of the time-varying electromagnetic waves manifests in EMR, which oscillates at the same rate as the nominal AC frequency (50 or 60 Hz in most countries). As modern structures are built with complex networks of power lines and wirings, the ambient EMR can be ubiquitously observed within the buildings. In fact, the effect is so pronounced that it is often easily detectable as an unwanted "60-Hz tone" on microphones, amplifiers, and other radio-related instruments [14].

At a specific location, the observed ambient EMR is a superposition of EMRs emitted by surrounding sources, including the power lines and electric loads. While the AC power lines are the main source of ambient EMR, various appliances produce random and transient electromagnetic waves on top of the power line EMR based on their operating states and conditions (e.g., plugged or unplugged) [18, 44, 77, 56]. Therefore, EMR measurements from two locations (in the same structure) may vary significantly even when they are measured at the same time due to the different distance to the power lines, different spatial geometry of the lines [61, 18], and transient operation of the appliances. Furthermore, even at a specific location, the ambient EMR may constantly vary depending on various electrical and human activities at or near the location. In summary, an ambient EMR signal at a certain location includes two major components: i) a periodic signal, typically a sinusoid, at the nominal power line frequency, and ii) superimposed noises, which are higher frequency components that represent temporal and spatial variations in an environment.

AeroKey leverages the spatial and temporal uniqueness of the superimposed noises to allow a group of co-located devices that observe similar EMR patterns to *autonomously* generate near-identical bit sequences from the patterns. To provide an intuition of these properties, I conduct a controlled experiment in a typical home environment. First, I directly connect a short conductor wire to an ADC pin of the two off-the-shelf MCUs (Ar-

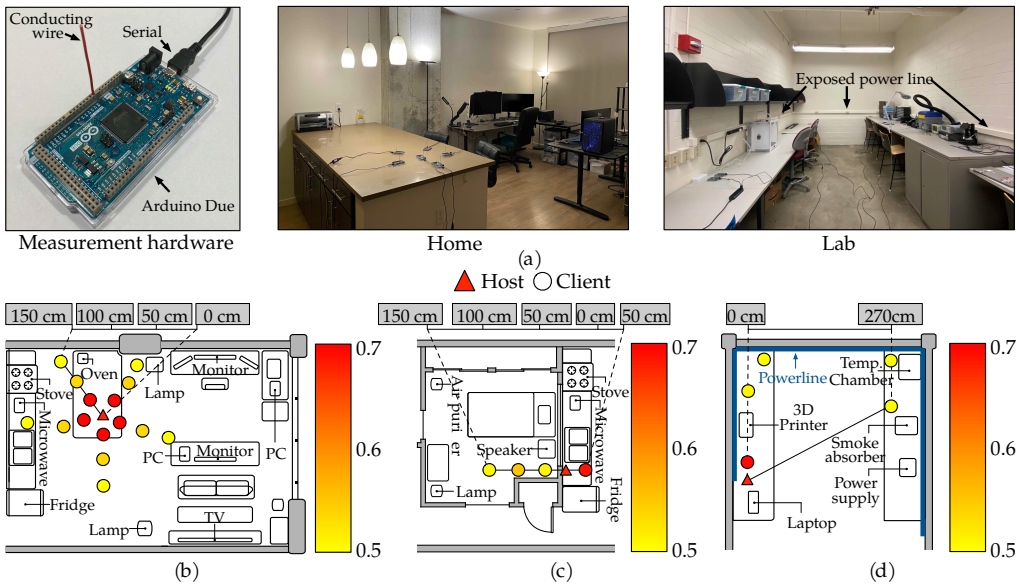


Figure 4.20: (a) Measurement hardware with a conducting wire as an antenna. (b) Correlation heatmap of superimposed noise components between host and client devices within typical living room environment. (c) Correlation between devices located in the next room. (d) Correlation between devices located along the identical power line.

duino Due). Then, devices read the ADC at two different locations that are 5 m apart, while power-cycling a 1875 W hair dryer in the middle between the two measuring points. As Figure 4.19(a) shows, the ADC readings at both locations exhibit a synchronous 60 Hz sinusoidal structure due to the power line EMR. However, a closer look at the pairs of periods captured from both signals reveals clear discrepancies in the shape and amplitude before and after turning on the hair dryer, indicating the spatial variability of EMR from two different locations at the distance of 5 m. Comparing the signals captured at the same location before and after turning on the hair dryer, the difference is even more dramatic, indicating the temporal variability of EMR caused by the state of the surrounding appliances. Figure 4.19(b) presents the spectrograms of the signals that show the random

and dynamic variabilities over time, location, and frequency. AeroKey leverages these spatiotemporal variations (on top of the 60 Hz signal) caused by various surrounding electromagnetic noises to harvest unique secret bits only between co-located devices.

To further explore the spatiality of the EMR signal, I extract the superimposed noise components and compare them between various locations within the home and lab environments. Figure 4.20(a) illustrates the measurement hardware (Arduino Due with a conducting wire connected to an input pin of the ADC) as well as the images of the experiment environments. Figure 4.20(b), (c), and (d), shows the correlation heatmap of the extracted noise components between the two devices (Host and Client) at varying locations. In typical living room environment (Figure 4.20(b)), as the distance between the devices increases, the correlation gradually decreases from 0.7 at 50 cm down to 0.5 at 150 cm apart. The decrease in the correlation is observable in all directions with increase in the distance, which leads to demonstrate the spatial difference of the EMR noise between the two distinct locations. If the devices are separated by the wall (Figure 4.20(c)), the correlation experiences slight decrease compared to an open environment, exhibiting correlation of 0.5 at 50 cm. This is because the observed signal between two devices is slightly different due to the distance variation from the surrounding appliances and the noise signal that gets attenuated by the wall. Figure 4.20(d) illustrates the effect of the power line to the observed signal by conducting an experiment in the lab. Although all the devices are located close (within 20 cm) to the same power line, the correlation decreases between devices due to the surrounding electrical loads producing varying noise signal at different locations. This leads to demonstrate that the EMR at a specific location is spatially unique, which allows AeroKey to only authenticate closely located devices.

4.2.1 System and Threat Models

System model: The main objective of AeroKey is to provide a mechanism for secure initial authentication between two headless commodity devices. As such, one of the use cases of AeroKey is to provide an alternative to the pairing in Bluetooth. The security level of pairing Bluetooth devices hinges on securely exchanging a six-digit code (32 bits of entropy) through a user utilizing the I/O interfaces. These mechanisms, however, remain as pain points for users to suffer in the initial pairing stage. AeroKey brings at least this level of security to headless commodity devices without adding additional capabilities and user interaction.

AeroKey considers typical indoor environments where pairs of headless devices located within a short distance of each other attempt to establish a secure wireless connection (authentication). The user is able to keep legitimate devices within a close range and remove any suspicious devices away immediately. If a pair of devices are within close proximity, defined as the *personal authenticated region* (*PAR*), they are considered trustworthy and safe to be connected. The maximum radius of the PAR is limited to one meter, which is a typical range that the user can keep the devices under control (e.g., personal desk).

AeroKey assumes a generic device model that is reasonable for most low-cost personal devices. In AeroKey model, headless devices are operated by a low-cost MCU with an integrated, typically low-precision, ADC. They can exchange messages over the established wireless channel (e.g., Bluetooth, Wi-Fi, or Zigbee) to process authentication requests. The authentication process is considered successful only when the pair of devices agree on a symmetric secret key. AeroKey assumes no pre-established time synchronization between the devices; their system clock is not synchronized, and the ADC sampling frequency may be slightly offset. AeroKey assumes the power line voltage of 120 V at 60 Hz; the application, however, is not limited to a specific voltage or frequency.

Threat model: AeroKey considers an adversary aiming to illegitimately authenticate its device to the victim's device as a steppingstone for additional threats. Further, the adversary's device is not located within the victim's PAR at the same time during the authentication process. While it is possible for the adversary to perform an attack by placing or hiding an adversarial device within the PAR (e.g., underneath desk or chair), it would significantly raise the bar for the attack. Furthermore, such attack scenario can be thwarted by adopting a simple confirmation or selection interface (e.g., asking the user to confirm or select the associating device) with minimal user involvement.

The adversary, however, does have access to the victim's future expected PAR when legitimate devices are not within the vicinity. Furthermore, the adversary has full knowledge of the legitimate device's location as well as their authentication timestamps (i.e., protocol initiation time with fine granularity). I assume the adversary to be fully aware of AeroKey's underlying protocol; it is capable of eavesdropping on any plaintext wireless messages that are used in the legitimate authentication process and transmitting arbitrary packets to the devices. I consider denial of service attacks, such as jamming, to be outside the scope of AeroKey as such attacks are applicable to any wireless communication mechanisms and are not special to AeroKey. The following lists four types of attack scenarios which is evaluated later in the Section.

- *Passive attack:* A nearby device that resides just outside of the victim's PAR tries to authenticate itself to the victim's device by generating bit sequences from its own EMR measurement. This scenario can be common where devices belonging to different users within the same house or space can accidentally authenticate with each other.

- *Replay attack:* An adversary with the knowledge of future PAR and their authentication timestamp gains partial access to the location and performs measurements from that specific location. For example,

the adversary has access to a desk at which the victim will sit tomorrow. Afterwards, when legitimate authentication takes place at the similar time and location, the adversary uses previously measured EMR to generate bits to authenticate to the victim's device.

- *Injection attack:* An adversary is capable of inducing strong EMR to the environment to force a common bit sequence as the victim's device, for example, using a high-wattage signal generator and a load. This can also accidentally happen when such high-wattage load is active, and an inadvertent device is nearby.

- *Machine learning (ML) attack:* An adversary with the knowledge and access to the future PAR leverages previously measured EMR as well as any eavesdropped plaintext messages exchanged between legitimate devices to train an ML model and uses it to authenticate itself or directly predict unique bit sequences derived from legitimate authentication process.

4.2.2 Proposed Approach

AeroKey Protocol

This part details AeroKey's protocol between Device A as the host and Device B as the client. The underlying authentication mechanism of AeroKey requires two co-located devices to generate similar *evidence bit sequences*, B_d (d denotes device name, A or B) to be authenticated. Figure 4.21 illustrates various notations used within the pipeline of the protocol consisting of five stages: i) Measurement (MEAS), ii) Synchronization (SYNC), iii) Preprocessing and feature extraction (PPFE), iv) Bit quantization (QUANT), and v) Key reconciliation (RECON). Each pass of the first four stages is denoted as a *cycle* and generates a partial evidence bit sequence (not necessarily by an equal amount). The cycle repeats until the number of

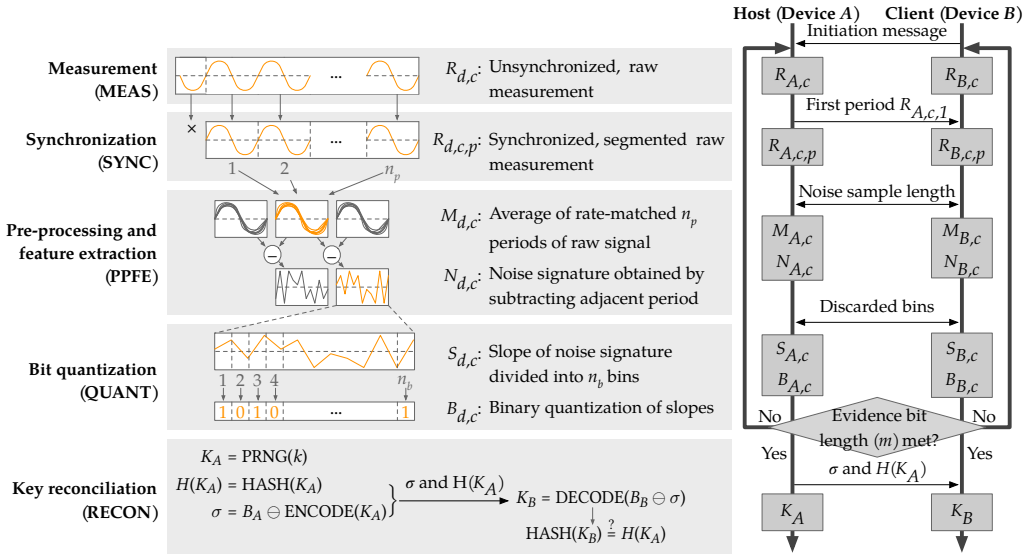


Figure 4.21: Five-stage pipeline of the AeroKey protocol.

evidence bits reaches the desired length, m. Afterwards, the RECON stage allows Devices A and B to establish an encrypted wireless channel using a symmetric key, K, of length k. The details of each stage are explained in the following.

Measurement (MEAS)

A primary design goal of AeroKey is to minimize hardware overheads. This goal is especially crucial for devices with stringent form factor constraints (e.g., wireless earbuds). In the MEAS stage, it achieves this goal by measuring ambient EMR using the ADC embedded in MCUs without any analog front-end circuitry for amplification or filtration. In particular, AeroKey considers a simple way to measure the ambient EMR: a short conductive wire (or a PCB trace) connected to a floating single-ended ADC pin in lieu of a frequency-matched antenna.

The client (Device B) commences the AeroKey protocol by sending

an initiation message to the host (Device A). Both devices independently read their ADC for a pre-defined duration to capture the electric potential on the wire excited by the ambient EMR. The raw time-series measurement signals are denoted by $R_{A,c}$ and $R_{B,c}$, respectively, where c is the cycle count ($c = 1, 2, \ldots, n_c$), and n_c is the total number of cycles. Because the input impedance of a floating ADC pin is very high, the measured signal is weak and susceptible to various sources of electrical noises. Therefore, subsequent stages of the AeroKey protocol carefully synchronize and pre-process the obtained raw signals before extracting evidence bits, as described in the following subsections.

Synchronization (SYNC)

AeroKey avoids the need for tight clock synchronization that is costly and unavailable for low-cost wireless devices [1], and as a result, the two obtained raw signals, $R_{A,c}$ and $R_{B,c}$, are not temporally aligned. The timing mismatch is mainly attributed to the transmission delay of the initiation message, which almost always leads the raw signal of Device A to lag behind the signal of Device B. If comparing the worst-case Wi-Fi transmission time (up to 250 ms [65]) to the length of a single period in 60 Hz ambient EMR (16.7 ms), the starting point of the measured signal between each device can be off by several periods.

To achieve a common notion of time, the SYNC stage leverages the measured signals' sinusoidal property and the structural similarity of the signals measured at the same time. It first leverages the 60 Hz sinusoidal property of the ambient EMR to segment the raw signals into individual periods using zero-crossing detection. Although the raw signals exhibit a strong 60 Hz component, they are considerably noisy with occasional high amplitude peaks, which may result in multiple zero-crossing points within a single period and, in turn, lead to inaccurate detection of period boundaries. To deal with this, this stage conditions the normalized raw

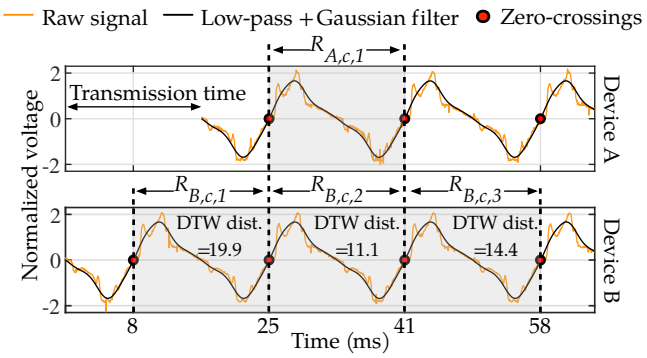


Figure 4.22: Synchronization of raw signals between Device A and B. Among three periods of Device B, the second period, $R_{B,c,2}$, shows the minimum DTW distance against Device A's $R_{A,c,1}$.

signal with a software-based low-pass filter (cut-off at 60 Hz) and apply a Gaussian filter to further smooth the signal. Figure 4.22 illustrates an example of raw and conditioned signals, as well as detected positive-going zero-crossings of two devices. From the zero-crossing indices, each device segments $R_{A,c}$ and $R_{B,c}$ into $R_{A,c,p}$ and $R_{B,c,p}$, respectively, where p denotes the period count ($p = 1, 2, \ldots, n_p$) within cycle c, and n_p is the number of periods per cycle. Note that the filtered and smoothed signal is only used to extract the zero-crossing indices used to mark the raw signal into individual periods.

Next, Device A sends the first period of the measured raw signal, $R_{A,c,1}$, to Device B. Upon receiving $R_{A,c,1}$, Device B finds the closest matching period among $R_{B,c,p}$ by sequentially comparing measurements from each period against $R_{A,c,1}$. However, the number of samples in each period is not uniform even when both devices use the same hardware and measurement settings because: i) surrounding appliances inject transient and continuous noises that cause shifting of zero-crossing points, resulting in fluctuations in the frequency measurement, and ii) oscillators of low-cost MCUs suffer from large frequency variability ranging from -17% to $+15\%$ [44, 27]. As

such, common signal similarity metrics such as root mean squared error (RMSE) or Pearson correlation coefficient are not applicable. To address this problem, VoltKey utilizes the dynamic time warping (DTW) distance method, an algorithm for measuring the similarity between two time sequences that may vary in their speed or sampling rates [6]. It calculates the optimal time-warping path between two signals and outputs the closest Euclidean distance between them.

In the example shown in Figure 4.22, the closest period in Device B is the second one, $R_{B,c,2}$, which exhibits the minimum DTW distance of 11.1 against $R_{A,c,1}$. Upon finding the closest period p (in this example, $p = 2$), both devices discard all measured samples up to end of $R_{A,c,1}$ and $R_{B,c,p}$ and re-segment the subsequent periods as $p = 1, 2, , \ldots, n_p$ as the publicly exchanged $R_{A,c,1}$ can be eavesdropped by the adversary and should not be used to generate evidence bits.

Pre-processing and Feature Extraction (PPFE)

The main goal of the following PPFE stage is to extract a portion of raw signals that is rich in randomness to be converted into evidence bits. Even between two closely located devices, the raw signals often exhibit low correlation due to slight location differences, environmental noise, and hardware variations. For a pair of signals to be used as a source for evidence bit generation, they should exhibit some correlation across co-located devices so that near-identical bit sequences can be extracted. To increase the correlation, I take the sample-wise mean of n_p raw period signals measured by device d in cycle c into a *mean signal* $M_{d,c}$:

$$M_{d,c} = \frac{\sum_{p=1}^{n_p} R_{d,c,p}}{n_p}. \tag{4.3}$$

During this process, because each period contains a different number of samples, each device linearly re-samples all of its marked periods down

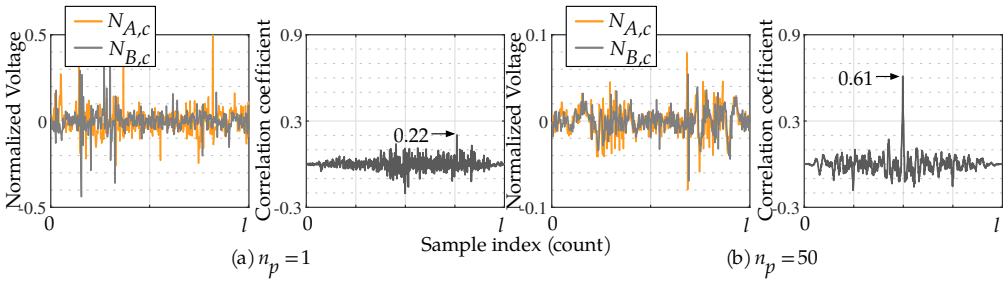


Figure 4.23: Two noise signals, $N_{d,c}$, and their cross-correlations calculated from mean signals with (a) $n_p = 1$ and (b) $n_p = 50$ on two co-located devices.

to the minimum period length to make the sample number of each period equal. The signal $M_{d,c}$ is taken as the average of the signals $R_{d,c,p}$ over n_p periods of cycle c. It represents the general shape of the EMR waveform for the duration of the MEAS stage as observed by each device. For instance, when the MEAS stage in each cycle runs for 1 s, it is expected that the mean signal captures the general shape of around 60 periods ($n_p = 60$), which will be more similar across co-located devices than a direct comparison of raw signals. The main reason that the mean signal mainly leverages 60 Hz over the higher frequency component (over 1 kHz range) is because: 1) typical low-cost IoT devices with on-chip ADC does not support high frequency sampling which only allows us to accurately measure signal bandwidth of below few KHz and 2) higher frequency component is difficult to correctly segment and measure due to lack of amplifier and precision analog front-end circuitry.

From the obtained mean signals, each device extracts the cycle-to-cycle temporal amplitude variation referred to as *noise signal*, $N_{d,c}$. This noise signal, mainly used to extract spatiotemporally unique evidence bit sequences, represents the index-wise subtraction of mean signals from

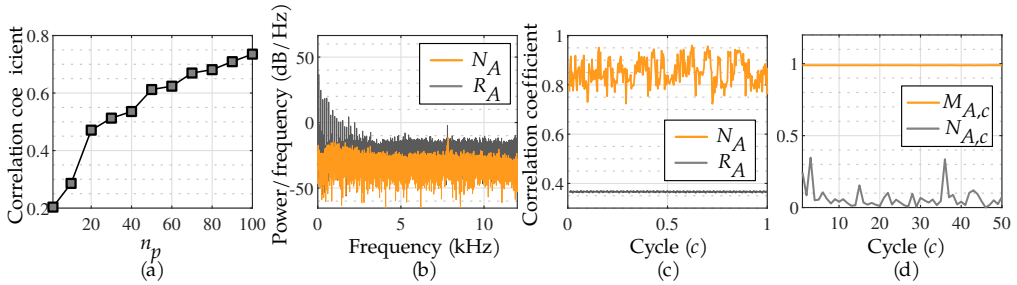


Figure 4.24: (a) Relationship between n_p and correlation coefficient between two noise signals. (b) Periodogram of noise signal, N_A, and raw signal, R_A. 60 Hz component is removed in the noise. (c) Spectral entropy of the noise signal, N_A, and the raw signal, R_A.(d) Correlation achieved between $R_{A,c,1}$ and $M_{A,c}$, as well as $N_{A,c}$.

two consecutive cycles:

$$N_{d,c} = M_{d,c+1} - M_{d,c}. \tag{4.4}$$

Because multiple mean periods, $M_{d,c}$, share a similar 60 Hz dominant structure, subtracting two subsequent mean signals essentially removes this dominance, and only captures the transient EMR variations between two subsequent cycles resulting from various electrical activities. Therefore, even if the adversary has access to the dominant structure of the raw signal (i.e., from publicly exchanged $R_{A,c,1}$ in the previous stage), it is not correlated to the resulting noise signal used for key generation. Finally, the two devices agree on the length of their noise signals by re-sampling them to match the length of the shorter one, denoted by l, called the *agreed noise sample length*.

To demonstrate the effectiveness of the signal processing technique in this stage, I show in Figure 4.23 two pairs of noise signals obtained from two co-located devices. Figure 4.23(a) represents the noise signal pair with the mean signals calculated with $n_p = 1$ (equivalent to no pre-processing), and Figure 4.23(b) with $n_p = 50$. When $n_p = 1$, the two

noise signals have a low max correlation of 0.22. On the other hand, when calculated from the mean of 50 periods ($n_p = 50$), they show a significantly higher max correlation of 0.61. In Figure 4.24(a), I further investigate the relationship between n_p and correlation between noise signals on two devices. We can observe a significant increase in correlation from 0.20 to 0.74 as the n_p increases from 1 to 100. This suggests that the longer MEAS stage runs to measure a greater number of 60 Hz periods, the two noise signals independently extracted from two devices exhibit higher similarity, which can ultimately result in a higher probability of extracting identical bits on a pair of authenticating devices. In Figure. 4.24(b), I show a periodogram (estimate of the spectral density) of the extracted noise signal, N_A, and raw signal, R_A. Because N_A retains relatively similar power between range of 0 kHz to 12 kHz, it successfully demonstrates that the PPFE stage effectively removes the 60 Hz sinusoidal property of the raw signal. In Figure 4.24(c), I compare the spectral entropy of raw signal R_A against the noise signal N_A. The pre-processed noise signal exhibits more randomness and irregularities (mean of 0.85) compared to the raw signal that mainly fluctuates in sinusoidal behavior (mean of 0.36). This indicates that the extracted noise signal is more suitable for random bit generation purposes. It is important that the extracted noise signal is not similar to the raw signal since the first period of the raw signal, $R_{A,c,1}$, is exchanged over the public channel (during SYNC stage) and is assumed to be eavesdropped on by the adversary. In Figure 4.24(d), I illustrate the correlation achieved between $R_{A,c,1}$ and the following mean signal, $M_{A,c}$, as well as the noise signal, $N_{A,c}$, with respect to subsequent cycle number. While the attacker can generally infer the shape of the mean signal with a high correlation of 0.98, there is relatively low correlation between $R_{A,c,1}$ and $N_{A,c}$ due to subtraction operation that removes the dominant shape. Because $N_{A,c}$ is the main source of entropy in generating bit sequences, this suggests that the attacker cannot leverage $R_{A,c,1}$ to infer $N_{A,c}$ used to

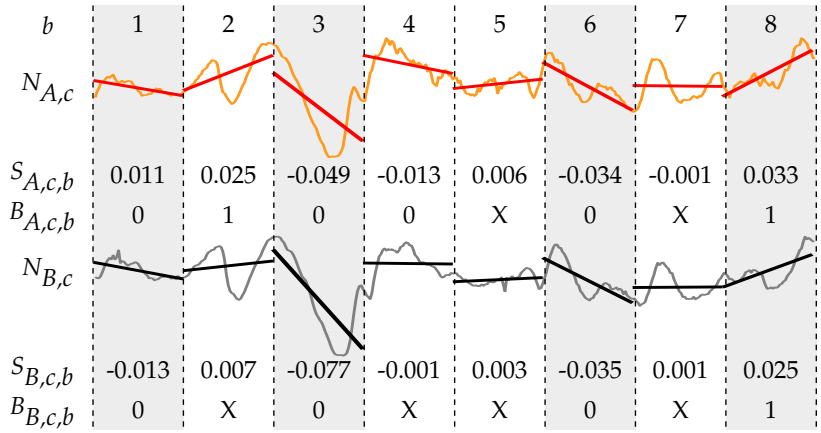


Figure 4.25: In QUANT stage with $n_b = 8$, the bits are only extracted from bins with slopes greater or less than the quantization threshold of 0.01: bins 1, 3, 6, and 8.

extract the evidence bit sequences.

Bit Quantization (QUANT)

In the QUANT stage, both devices translate the time-aligned noise signals ($N_{d,c}$) into a sequence of evidence bits, B_d, used to derive a symmetric key in the following RECON stage. AeroKey leverages the *gradient of the amplitudes of noise signals* to extract bit sequences. Each noise signal is segmented into n_b *bins* of the same length, with b representing the bin count ($b = 1, 2, 3, \ldots, n_b$). Since the length of the noise signal is l, the length of each bin is $\lfloor \frac{l}{n_b} \rfloor$. Next, this stage determines the slope of each bin, denoted by $S_{d,c,b}$, by curve-fitting its noise signal as a linear function, which is subsequently converted into a bit $B_{d,c,b}$: a bit 1 for a positive slope or a bit 0 for a negative slope. To minimize bit discrepancies due to small differences in the noise around zero, the stage only use the bins with the absolute value of the slope higher than a *quantization threshold*,

th; otherwise, the bins are discarded:

$$B_{d,c,b} = \begin{cases} 1 & \text{if } S_{d,c,b} > th \\ 0 & \text{if } S_{d,c,b} < -th \\ X & \text{discarded otherwise.} \end{cases} \tag{4.5}$$

Since this bit extraction is independently performed by two devices on their own noise signals, one device might discard bins (and corresponding bits) that are kept by the other. To use bits extracted from common bins, the two devices exchange the extracted bin numbers and agree to generate bits only from the bins that are kept by both devices. Note that exchanging the bin numbers is assumed to be eavesdropped by the adversary over the unencrypted channel, but it neither reveals information about the extracted bits nor makes it easier to guess them.

Figure 4.25 illustrates an example of the bit quantization process with $n_b = 8$ and $th = 0.01$. In this example, Device A extracts six bits from bins 1, 2, 3, 4, 6, and 8, and Device B extracts four bits from bins 1, 3, 6, and 8, where the absolute value of the slope is greater or less than positive and negative th, respectively. Device A may generate a bit 1 from bin 2 ($S_{A,c,2} > th$), but since Device B discards it ($-th < S_{B,c,2} < th$), bin 2 is discarded by both of them. A similar observation occurs for bin 4 as well. As a result, a bit sequence "0001" is extracted from bins 1, 3, 6, and 8 that both devices agree to use. In this process, both th and n_b impact the bit agreement rate and the number of extracted bits. Intuitively, increasing th increases the bit agreement rate at the cost of a reduced number of extracted bits per cycle, hence a longer authentication time. In the case of n_b, it should be chosen to estimate the gradient with fine granularity. I empirically choose the values of n_b and th later. Note that the number of extracted bits varies each cycle due to the random nature of the noise. Therefore, the cycle from the MEAS stage through the QUANT stage is

Table 4.2: Overview of RECON stage between Devices A and B (Fuzzy commitment).

	Device A		**Device** B
①	$K_A = \mathrm{PRNG}(k)$		
②	$H(K_A) = \mathit{HASH}(K_A)$		
③	$\lambda_A = \mathrm{ENCODE}(K_A)$		
④	$\sigma = B_A \ominus \lambda_A$	$\xrightarrow{\sigma, H(K_A)}$	$\lambda_B = B_B \ominus \sigma$
⑤			$K_B = \mathrm{DECODE}(\lambda_B)$
⑥			$H(K_A) \stackrel{?}{=} \mathit{HASH}(K_B)$

repeated until the total amount of extracted bit length m is reached by the generated evidence bit sequences B_A and B_B.

Key Reconciliation (RECON)

In the RECON stage, the two devices agree on an identical secret key, K, using the two evidence bit sequences, B_A and B_B. An underlying assumption in this stage is that two devices should authenticate if and only if their extracted evidence bit sequences have a small number of bit errors. AeroKey employs the *fuzzy commitment* scheme [33], where a key is turned into a commit/open pair using evidence bits, in such a way that only similar evidence bits can correctly open a key. This allows devices to transfer the secret data on the public channel as long as the adversary does not have similar evidence bits. The overview of the RECON stage between Device A and B is shown in Table 4.2. First, Device A generates a key K_A of length k, where $k < m$, using a pseudo-random number generator to be used as the symmetric key and calculates its hashed (e.g., SHA-256) value, $H(K_A)$. Then K_A is encoded into λ, using error correction code (e.g., Reed-Solomon coding), e.g., $\lambda = \mathrm{ENCODE}(K_A)$. The commitment σ is then calculated, which is a finite field subtraction between the generated evidence bits

and encoded codeword λ, i.e., $\sigma = B_A \ominus \lambda$. Then, σ and $H(K_A)$, which do not divulge any information about B_A or K_A, are sent to Device B. Upon receiving σ, Device B, derives K_B by decoding the subtraction result of σ and B_B, i.e., $K_B = \mathrm{DECODE}(\lambda_B)$ where $\lambda_B = B_B \ominus \sigma$. To verify the equality of the derived key, Device B compares the hashed values of K_A and K_B, i.e., $H(K_A) \stackrel{?}{=} H(K_B)$; if identical, two devices are successfully authenticated and can establish a secure channel with the derived key.

The fuzzy commitment scheme ensures identical derivation of the key as long as the error (Hamming distance) between the two evidence bit sequences does not exceed $\frac{m-k}{2}$ bits. The latter condition ensures that the difference operation $(B_B \ominus \sigma)$ when performed at Device B results in λ_B that is close to λ at A. The subsequent decoding of $B_B \ominus \sigma$ reveals K_B that should be the same as K_A.

4.2.3 Implementation and Evaluation

In this part, I evaluate the performance of AeroKey with varying parameter settings, deployed in various real-world environments. I also show the robustness of AeroKey against various strong adversaries.

Experimental Setup

AeroKey uses the on-chip ADC of commercial off-the-shelf MCU on the Arduino Due with a conductive wire to measure the ambient EMR. The Arduino Due is equipped with Atmel's 32-bit SAM3X8E ARM Cortex-M3 processor that can operate at 84 MHz. The ADC is set at a sampling rate of 24 kSPS (samples per second) with a 10-bit resolution. The conductive wire has a length of 6 cm unless stated otherwise and wires are oriented parallel to one another to ensure that they both experience the same oscillations in the EMR. The number of periods measured in each cycle (n_p) is fixed at 30 period/cycle (2 cycle/second), which requires only about

30 kB of storage space that can easily be supported even by typical low-cost MCUs without external memory. Finally, the final symmetric key length, k, is set to 128 bits, and I evaluate different error tolerance rates ranging from 5% up to 45% by varying the length of evidence bit sequences, m.

The evaluation considers two common use environments: a home and a lab. The home represents an uncontrolled use environment with high-wattage uncontrolled loads, such as ceiling lights, televisions, computers and a fridge, which are switched on or off anytime. On the other hand, the lab represents a controlled use environment. It has several low-amp uncontrolled loads, including lighting, computers, and monitors; and three high-wattage controlled loads, including a fan (50 W), a temperature chamber (240 W), and a smoke absorber (110 W). The controlled loads are not power cycled when the protocol is in progress. The total amount of measurement time is 1093 hours in the home and 1045 hours in the lab.

Evaluation Metrics

To demonstrate generating common evidence bit sequences using the proposed bit quantization, I use the following metric:

- *Bit agreement rate (BAR):* Rate of matching bits between two evidence bit sequences when compared bit-by-bit.

To evaluate the security of the authentication process, I use the following metrics:

- *True acceptance rate (TAR):* Rate of successful authentication between the legitimate devices within PAR. The false rejection rate (FRR) equals to 100 - TAR.
- *False acceptance rate (FAR):* Rate of successful authentication from the passive, replay, injection and ML attacks.

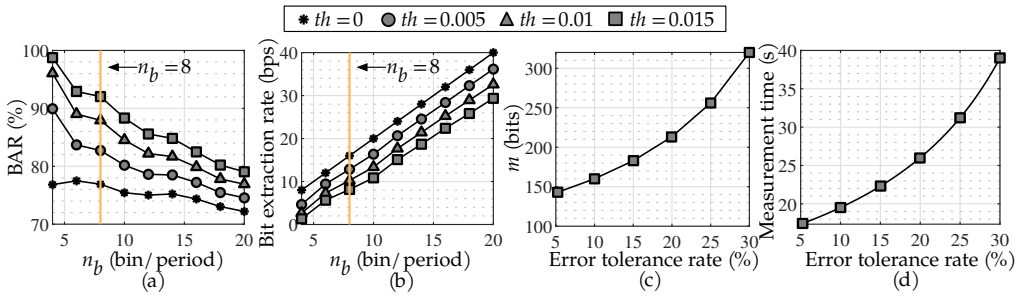


Figure 4.26: (a) BAR and (b) bit extraction rate between co-located devices for varying n_b and th. (c) Evidence bit length (m) and (d) measurement time required for varying error tolerance rate.

- *Equal error rate (EER):* Intersection point where FAR is equivalent to FRR. Low EER represents higher accuracy of distinguishing legitimate over adversarial devices.

Protocol Parameters

I first investigate the effects of the main protocol parameters to determine the values that dictate the performance of bit quantization. As previously discussed, th and n_b should be carefully chosen in the QUANT stage in order to achieve a high BAR and bit extraction rate. Figure 4.26(a) shows BAR for varying n_b and th between devices that are located 20 cm apart. Overall, the BAR consistently decreases as the n_b increases from 4 bins to 20 bins. This is because the gradient of the noise signals becomes less distinctive as the bin width decreases, leading to slight signal differences to extract erroneous bit pairs. When the quantization threshold is not applied ($th = 0$), BAR can be as low as 72.1%–77.4%. On the other hand, using a quantization threshold significantly increases the overall BAR because the bins with less distinctive slopes are more likely to be discarded. Specifically, when $th = 0.015$, high BAR reaches above 90% for n_b up to 9.

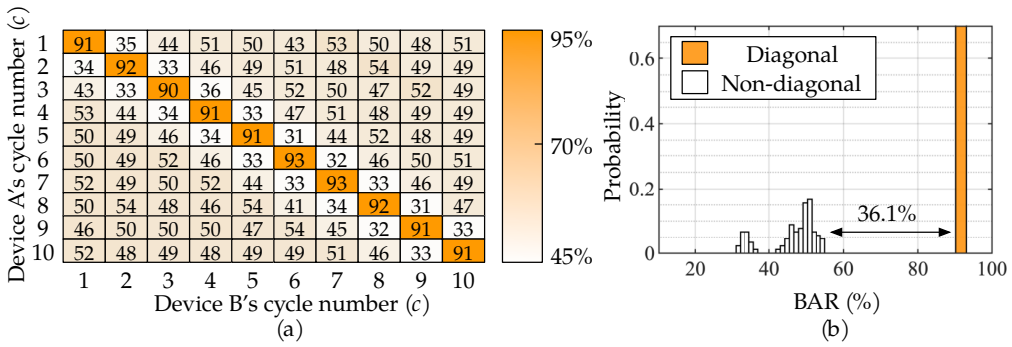
High BAR alone, however, is not enough for robust and usable (fast)

authentication. Increased BAR at low n_b and high th comes at the cost of reduced bit extraction rate, as shown in Figure 4.26(b). Specifically, when $n_b = 4$, the extraction rate ranges from 1.3 bps to 8 bps, while $n_b = 20$ exhibits 29.3–40.0 bps. As th increases, the bit extraction rate decreases due to greater number of discarded bins. To achieve the balance between the BAR and bit extraction rate, I set $n_b = 8$ for the rest of the evaluation; this value exhibits a relatively high BAR while maintaining a reasonable bit extraction rate ranging from 8.1–16 bps.

As previously mentioned, the fuzzy commitment scheme guarantees identical key ($k = 128$ bits) reconciliation and successful authentication as long as the number of bit errors between two evidence bit sequences does not exceed $\frac{m-k}{2}$ bits. Figure 4.26(c) shows the relationship between the error tolerance rate and the length of required evidence bit sequences, m. To tolerate up to 5% error between two evidence bit sequences, m needs to be at least 143 bits. To allow up to 30% error, at least 320 bits need to be extracted. While a higher tolerance rate increases the TAR between authenticating devices, it also allows adversaries to successfully authenticate to legitimate ones with fewer number of correct evidence bits. Furthermore, higher tolerance directly leads to longer measurement time due to higher number of required evidence bits. For instance, as shown in Figure 4.26(d), tolerating 30% of error requires devices to execute MEAS stage for at least 39.4 s (with $th = 0.015$ and $n_b = 8$) while 5% tolerance requires only around 17.4 s. Therefore, the error tolerance rate should be kept minimal to guarantee low authentication time for a seamless user experience. In the subsequent evaluations, I vary the error tolerance rate to find the optimal balance between authentication reliability and security.

Temporal Uniqueness of Evidence Bit Sequences

While it is important for two very closely located AeroKey devices to exhibit high BAR for reliable authentication, it is also imperative that the



	1	2	3	4	5	6	7	8	9	10
1	91	35	44	51	50	43	53	50	48	51
2	34	92	33	46	49	51	48	54	49	49
3	43	33	90	36	45	52	50	47	52	49
4	53	44	34	91	33	47	51	48	49	49
5	50	49	46	34	91	31	44	52	48	49
6	50	49	52	46	33	93	32	46	50	51
7	52	49	50	52	44	33	93	33	46	49
8	50	54	48	46	54	41	34	92	31	47
9	46	50	50	50	47	54	45	32	91	33
10	52	48	49	48	49	49	51	46	33	91

Figure 4.27: (a) 10-by-10 confusion matrix of average BAR between evidence bits generated at different cycles. (b) Distribution of BAR between diagonal and non-diagonal element pairs.

extracted evidence bit sequences are temporally unique for secure authentication. In other words, the evidence bits obtained at different times (cycle) within the same location should be different enough to prevent adversaries from reusing the previously extracted bits. To investigate this temporal uniqueness, I deploy two devices (A and B located 20 cm apart) to periodically obtain 10 consecutive noise signals (from 10 cycles) under a daily home environment with typical usage of various surrounding electronic loads for three days. Then, all the pairs of the noise signals on the two devices are used to extract the evidence bits using previously obtained parameters of $th = 0.015$ and $n_b = 8$. Figure 4.27(a) presents a confusion matrix of average BAR achieved between evidence bits generated at varying cycles between devices, A and B. The diagonal elements, representing BAR between bits generated at the same cycles, consistently exhibit over 90% while the non-diagonal elements, representing BAR from bits generated at different cycles, show close to 50%. Note that the elements neighboring the diagonal ones exhibit mean BAR of 33.1%. While this can imply that the adversary with the noise signal from one cycle can reuse it to derive 66.9% (100%–33.1%) of the bits derived in the next single cycle by flipping the bits, only about 4 bits are extracted per cycle which is negli-

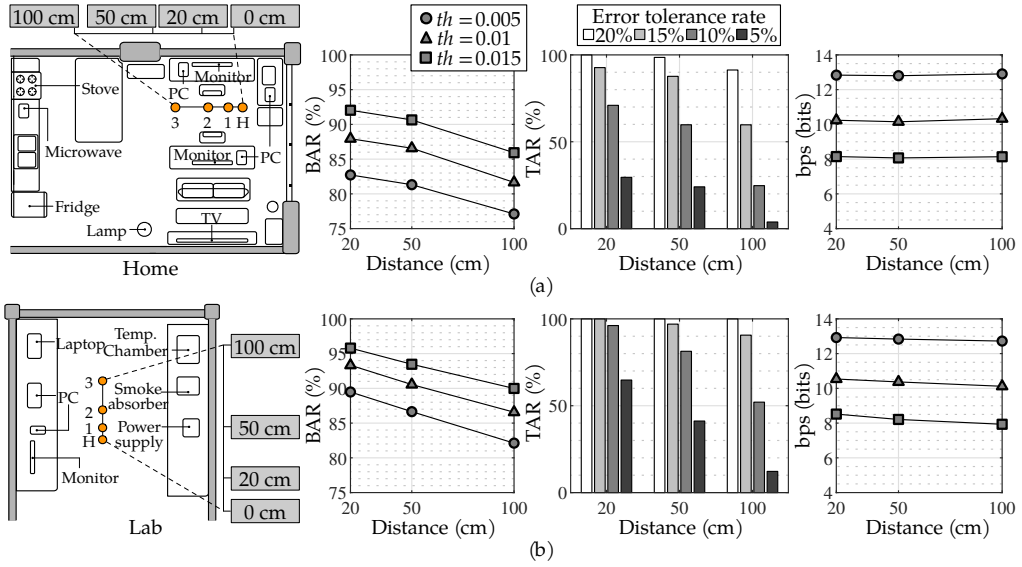


Figure 4.28: BAR, TAR, and bit extraction rate with respect to varying distance between authenticating devices within (a) home and (b) lab environment.

gible considering the total length of the evidence bit sequence ($m > 128$). Furthermore, the BAR exhibits close to a random guess beyond one subsequent cycle, with mean of 49.4%, which significantly makes it difficult for the adversary to reuse the noise signal. The distribution of the rates shown in Figure 4.27(b) illustrates the clear distinctions between diagonal against non-diagonal elements by at least 36.1%. This result successfully demonstrates that the evidence bits (and noise signal) extracted from one cycle does not closely correlate to the ones generated at different cycle. Moreover, this temporally unique characteristic implies that the extracted bits are high in entropy, making it difficult for the adversary to reuse the previously obtained bits within the same location.

Authentication Distance

To demonstrate AeroKey's reliable authentication performance among devices within PAR, I attempt to simultaneously authenticate a set of multiple client devices to a single host device under varying distances within 100 cm. Figure 4.28(a) and (b) illustrate the floorplan of the deployment environments, locations and distances between each device pairs. Additionally, I present BAR ($n_b = 8$), bit extraction rates and TAR between each device pairs. In each scenario, over 500 authentication attempts are made over the span of three consecutive days. When evaluating TAR, the error tolerance rate is varied from 0% up to 20% while setting $th = 0.015$.

In both environments, the BAR decreases as the distance increases and, higher th leads to higher BAR. The decrease in the BAR is mainly due to the device's location variation that leads to slightly different observed noise signals between authenticating devices. In the home, with $th = 0.015$, BAR is maintained above 90% between devices located within 50 cm apart. Specifically, it achieves 92.1% and 90.6% as the distance increases to 20 and 50 cm, respectively. The high BAR at a close distance ultimately leads to high TAR as shown on the right. Specifically, when 20 cm apart, 92.6% of the attempts are successful when the error tolerance threshold is set to 15%. Even with 10% error tolerance, the success rate is as high as 70.9%. When the devices are 50 cm apart, over 98% TAR is achieved with an error tolerance of 20%. If the distance increases to 100 cm, relatively high BAR is still maintained at 85.9% with $th = 0.015$. This leads to high TAR of 91.3% with 20% of error tolerance. Authentication attempts in the lab are more reliable than the attempts made in the home due to a fewer number of active electric loads and human activities. Within a 20 cm to 50 cm range, the devices exhibit a relatively higher BAR of over 93.4% with $th = 0.015$. This leads to high TAR of nearly 100% for devices within a distance of 20 cm, and 50 cm, respectively, under a 15% error tolerance rate. When the distance increases to 100 cm, a relatively high TAR is

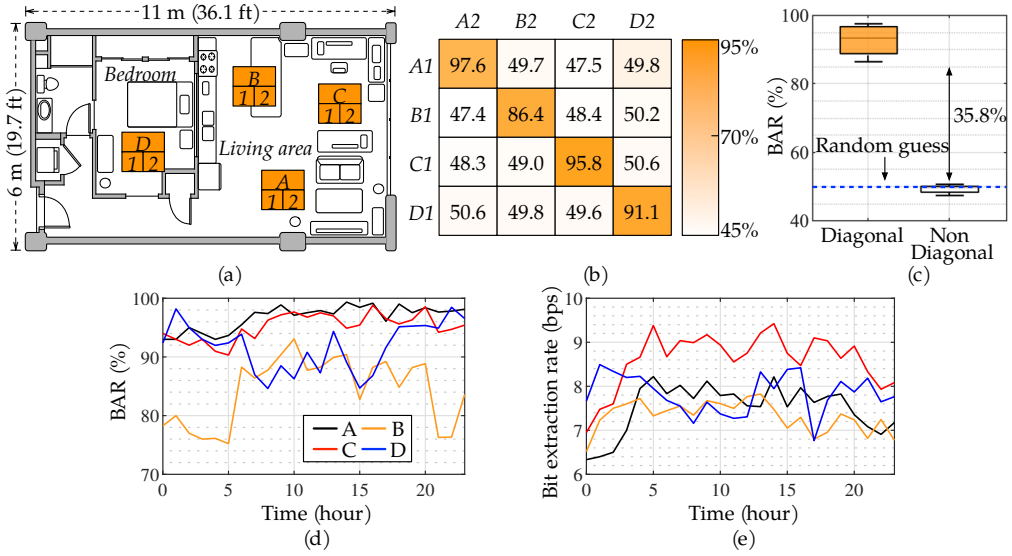


Figure 4.29: (a) Four different regions of deployed device pairs. (b) Confusion matrix and (c) distribution of BAR between all evidence bit sequence pairs. (d) BAR and (e) bit extraction rate of devices within four locations with respect to different hours of the day.

maintained with 90.6%. While the home environment achieves slightly less BAR and TAR than the lab, the bit extraction rate remains relatively equivalent. Additionally, the distance increase between the devices has no significant effect on the extraction rate since the noise signal remains relatively correlated. More specifically, in the home and lab environment, an average extraction rate of 8.2 bps and 8.3 bps is expected by the devices located inside PAR with $\mathrm{th} = 0.015$. Overall, under both environments with 20% error tolerance and identical parameter settings ($\mathrm{th} = 0.015$), AeroKey reliably authenticates all proximate devices with mean TAR of 98.3% within 100 cm, while maintaining similar bit extraction rates.

Spatial Variation

The size of PAR should be limited so that relatively distant devices, whether accidental or malicious, can be effectively rejected from authentication attempts. To validate this, I set a total of eight AeroKey devices within the home to attempt to authenticate with each other while deployed in four different regions that are at least 3 m apart. Over 1600 authentication attempts are made over the span of two consecutive days under typical electric load usages. Figure 4.29(a) depicts four different regions: A, B, C, and D. In each region, I place a pair of devices (1 and 2) within 20 cm of each other to evaluate BAR between different regions. Figure 4.29(b) and (c) illustrate the confusion matrix and the distribution of the BAR between evidence bit sequences generated by all pairs of deployed devices with $n_b = 8$ and $th = 0.015$. The diagonal elements in the matrix equate to the BAR between devices located in the same region whereas off-diagonal elements represent the BAR obtained between different regions. Clearly, devices within the same region, which are considered legitimate device pairs within PAR, show a high BAR ranging from 86.4% up to 97.6%. On the contrary, the BAR achieved between devices in different regions exhibits a mean of 49%, which differs from the legitimate pairs by at least 35.8%. This successfully indicates the spatially unique property of evidence bit sequences that can effectively allow AeroKey to distinguish proximate (legitimate) devices from the distant (adversarial) ones.

Because AeroKey strongly leverages the radiated EMR as the main source of entropy, the local environment including human activities and operating state of surrounding loads may impact the bit extraction rate, as well as BAR, achieved from devices within different locations. Figure 4.29(d) and (e), present the BAR and bit extraction rate ($th = 0.015$) of devices within four locations with respect to varying hours of the day. Different locations exhibit slightly varying performance throughout the day depending on their surrounding load usage characteristics. For in-

stance, devices within the bedroom (location D) suffer a slight loss in BAR and extraction rate during the daytime (6 a.m to 8 p.m) as the neighboring loads such as air purifier and cellphone charging are not active during work hours. On the contrary, devices in location B exhibit higher BAR and extraction rates during the daytime due to operating high wattage ceiling lighting nearby. The two locations A and C exhibit a relatively high and stable BAR throughout the day but the extraction rate is slightly higher during the daytime as they are surrounded by constantly running PCs, monitors and lighting. Nevertheless, regardless of time and varying human activities, devices in all four locations can experience a high BAR of above 85% and an extraction rate of above 6 bps.

Adversarial Scenarios

I next evaluate the robustness of AeroKey against the previously mentioned attack scenarios. The attacker aims to derive the legitimate device's symmetrical key by extracting similar evidence bit sequences (within the error tolerance level) using a combination of its previously observed and predicted signals. The attacker is aware of the parameter settings ($th = 0.015$ and $n_b = 8$) and is equipped with identical hardware and ADC settings as legitimate devices to further increase the attack success rate. I implement the five types of attacks (two types of ML attacks) as follows; for each attack type, over 100 authentication attempts are made:

- *Passive attack:* An adversary, just outside the boundary of a PAR (1.7 m and 2 m), attempts to authenticate in the home and lab environments in the presence of regular loads. I also investigate the effects of active high-wattage loads by activating a fan, a temperature chamber, or a smoke absorber placed within 10 cm of the victim device.

- *Replay attack:* The adversary with the knowledge of a future PAR and exact authenticating timestamp directly uses pre-measured EMR signals in the exact place at the same hour and minute of the day to authenticate itself in the home and lab environments.

- *Replay injection attack:* Replay injection attack is similar to replay attack, but the adversary activates high-wattage loads (fan, temperature chamber, and smoke absorber) within 10 cm of victim device in the lab environment.

- *Active injection attack:* An adversary activates high-wattage signal generator (with load) just outside the boundary of a PAR (1.7 m) to force a common environmental bits as the adversarial device located close (within 10 cm) to the generator. Unlike replay injection attack, the victim and the adversarial devices are time synchronized.

- *ML-raw attack:* An adversary with knowledge of the future PAR collects series of raw EMR signals over the course of one entire day and trains an ML model to use one raw 60 Hz period (input) to predict following subsequent 60 Hz (output) period. (i.e., $R_{A,1,1}$ to predict $R_{A,1,2}$). When legitimate authentication takes place, the adversary uses eavesdropped $R_{A,1,1}$ and b (in SYNC and QUANT stage) to sequentially predict subsequent periods and ultimately extract evidence bits in both home and lab environment. Specifically, I use an artificial neural network (NN) composed of a one hidden layer with 100 nodes, ReLU activation function, and a softmax layer to predict subsequent raw EMR signals.

- *ML-key attack:* An adversary with the knowledge of the future PAR trains an ML model to predict the first two bits of evidence bits using $R_{A,1,1}$. The training occurs for one day and uses the trained model and eavesdropped $R_{A,1,1}$ to predict the first two bits of legitimate

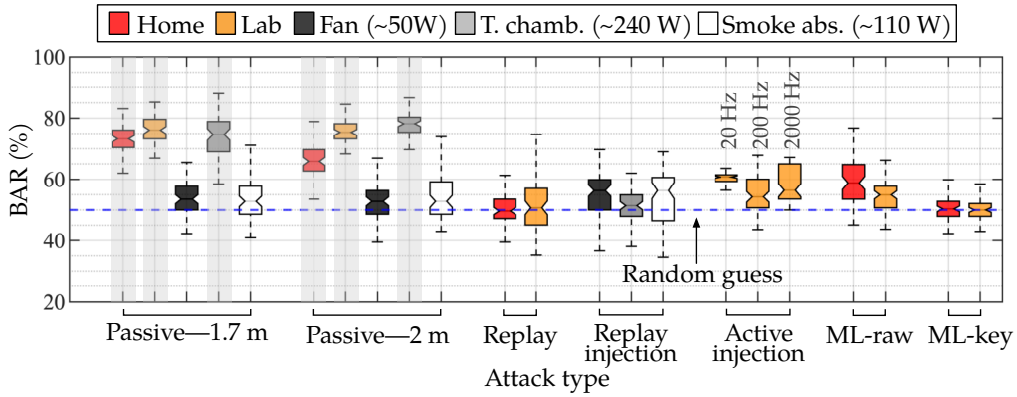


Figure 4.30: BAR achieved from passive, replay, replay injection, active injection and ML attacks. Six most effective attacks (high BAR) are highlighted in gray columns.

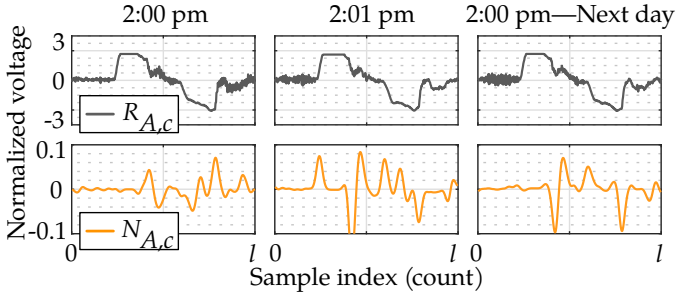


Figure 4.31: Raw ($R_{A,c}$) and noise ($N_{A,c}$) signals extracted from the same location at different time instances with an operating temperature chamber nearby (replay injection attack).

evidence bits the next day in the home and lab environment. I use a NN with one hidden layer, consisting of 28 nodes and an output layer with two nodes to predict the first two bits.

The distribution of the BAR achieved from all the attacks is shown in Figure 4.30. In the passive attack at 1.7 m, mean of 72.8% and 76.0% of evidence bits can be derived by the adversary in the home and lab, respectively, under regular load usage. When the temperature chamber is turned

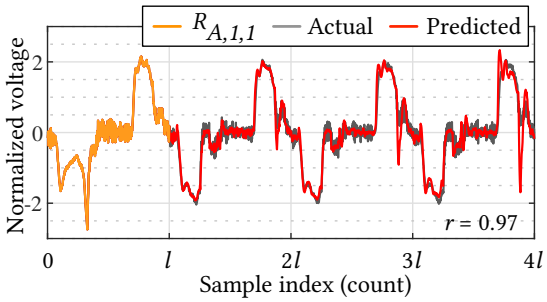


Figure 4.32: Measured and predicted raw signal using trained ML-model with ($R_{A,1,1}$) as an input (ML-raw attack). Two signals exhibit high correlation of $r = 0.97$.

on, the BAR remains relatively similar at 72.8%. On the other hand, when the fan or smoke absorber is turned on, they produce a unique EMR signal that does not propagate such a far distance, leading the attacks to achieve significantly lower BAR of 54.0% and 54.9%, respectively. Compared to the attacks from 1.7 m, attacks from 2 m achieve lower mean BAR of 66.0% and 75.9% in the home and lab, respectively, due to greater spatial variability of ambient EMR. Replay attack and replay injection attack result in a low BAR (close to random guess), ranging from 49.4% to 55.4%, either with or without active loads. This is because the raw EMR signal, $R_{A,c}$, even with its unique dominant structure, has temporal variability as illustrated in Figure 4.31. The PPFE stage of AeroKey essentially removes this dominant structure, leaving the noise signal, $N_{A,c}$, with just temporally varying noise signals. Even when the high-power consuming temperature chamber is operating nearby, $N_{A,c}$, extracted from different time instant of one minute apart within the same day (2:00 p.m. and 2:01 p.m.) differs significantly. Further, the same time on the next day exhibits different noise signal even under identical surrounding appliances. This effectively prevents an adversary from reusing the previously derived (at same hour and minute under the same load on a prior day) evidence bit sequences.

The active injection attack with varying frequency range is also not very

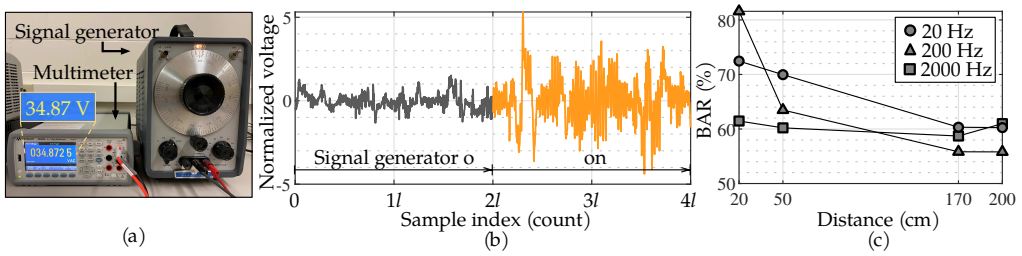


Figure 4.33: (a) Signal generator (Hewlett-Packard audio oscillator 200AB) outputting 34.87 V AC signal. (b) Noise signal, N_A, observed from the victim device as the signal generator is turned off and on (˜2000 Hz). (c) BAR achieved from the devices located in different distances from the generator signaling different frequency components.

effective in forcing enough environmental bits to the victim device, achieving mean BAR of 60.3%, 55.7% and 58.9% at output frequency of 20 Hz, 200 Hz and 2000 Hz, respectively at distance of 170 cm. Figure 4.33(a) illustrates the attack setup that outputs 35.87 V sine wave at different frequency. As shown in Figure 4.33(b), when the signal generator is turned on, the victim device located at 1.7 m away observes significantly varying noise signal with high amplitude. However, the generated EMR signal does not identically propagate through environment or gets measured by the distant devices due to EMR signal rapidly dissipating with the increasing distance. Even for a pair of devices located close to the generator within 20 cm, the mean BAR decreases to 72.4%, 61.4% and 81.5%, respectively for increasing frequency, which exhibits less BAR than passive attack without signal generator in the same environment. Similarly, ML-based attacks are not favorable to the adversary. ML-raw attack performs close to a random guess, achieving only 58.3% and 54.2% BAR in home and lab, respectively. As illustrated in Figure 4.32, although the ML model is able to closely predict the general structure of the subsequent raw signals with a correlation coefficient of 0.97, it was not effective in accurately predicting high frequency noise portion of the signal that AeroKey uses to

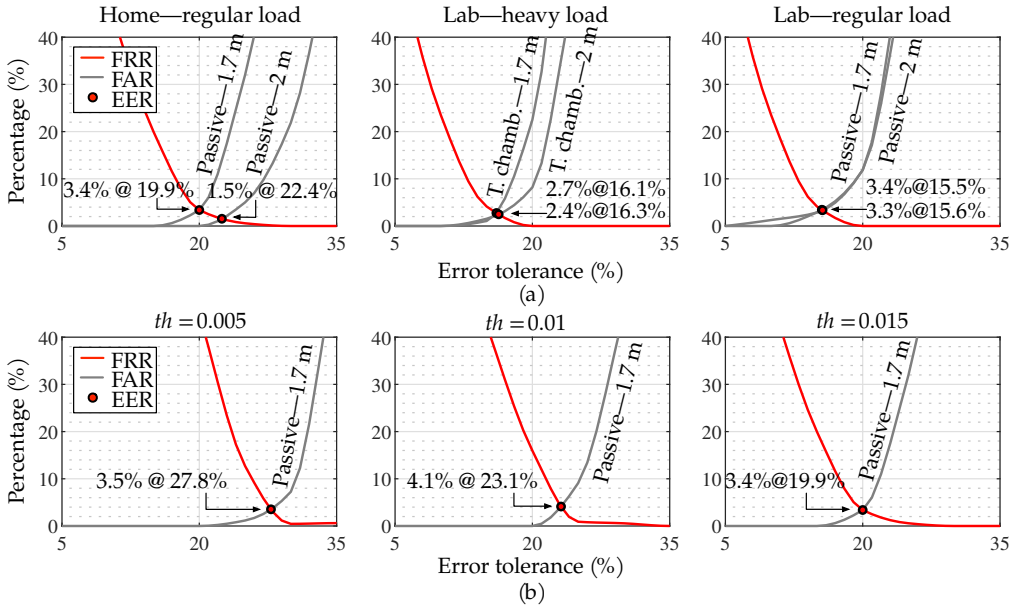


Figure 4.34: (a) Resulting EER from six most effective passive attacks. (b) EER from varying th by passive attack within home environment.

extract bit sequences. Additionally, the small error from the early period number (i.e., $p = 2$) propagates to prevent reliable prediction of the later periods (i.e., $p = 10$). Lastly, the ML-key attack was also not sufficient enough to predict enough legitimate bits with its mean BAR of 49.9% and 49.8% in home and lab, respectively. While earlier period number can be assumed to have more impact on the first few bits of the evidence bit sequences, the model fails to capture enough information about the noise signal. This demonstrates that the adversary, even with the eavesdropped $R_{A,1,1}$, cannot acquire much information about the evidence bit sequence itself.

Overall, the passive attacks are the most successful in obtaining relatively higher portions of legitimate evidence bits, because the timestamp of the attacks is synchronized to the legitimate authenticating devices. Figure 4.34(a) shows the EER of the six most effective passive attacks: at

1.7 m and 2 m range in the home and lab (with and without an operating temperature chamber) environments; all other attacks exhibit EER of 0% due to low BAR. The home and lab environment accomplish similar EER of at most 3.4% under regular load usage. Compared to the attacks at 2 m distance, attacks from 1.7 m exhibit higher EER at lower error tolerance rates. Because home is less permissive in authenticating devices compared to the lab, EER of 3.4% occurs at 19.9% error tolerance rate, which is comparatively higher than that of the lab, which occurs at 15.5%. In the lab, when the temperature chamber is operating nearby a legitimate device, a relatively lower EER of 2.7% and 2.4% is achieved by the same attacks. The above observation suggests that depending on the context of the environment, the user may manually (one-time configuration) vary the error tolerance rate (19.9% for home and 15.5% for lab) to strike an optimal balance between usability and security. Nevertheless, selecting the lower threshold of 15.5% can effectively reject 96.6% of the passive attacks while maintaining over 80% TAR in both environments.

I next evaluate the optimal error tolerance rate for varying th within the home environment against passive attack from 1.7 m. As illustrated in Figure 4.34(b), varying th results in different FRR and FAR due to lower th exhibiting lower BAR. Ultimately, this leads to different security and usability balancing point (EER) at a different error tolerance rate; higher th results in a lower rate. Specifically, the tolerance rate of the EER is achieved at 27.8%, 23.1% and 19.9% for increasing th of $0.005, 0.01$ and 0.015, respectively, while maintaining a low EER of less than 4.1%. This sets the guideline error tolerance for varying th, which is particularly related to the usability aspect of AeroKey. While lower th results in a higher bit extraction rate, the requirement of a higher error tolerance rate increases the number of extracted bits to derive identical final keys on both devices.

Table 4.3: Measurement time required for varying th in home and lab.

	Home			**Lab**		
th	0.005	0.01	0.015	0.005	0.01	0.015
Error tolerance rate (%)	27.8	23.1	19.9	23.3	19.1	15.5
m (bits)	287	238	213	240	207	186
bit extraction rate (bps)	13.0	10.4	8.2	12.9	10.3	8.3
Measurement time (s)	22.0	22.8	25.9	18.6	20.0	22.4

Authentication Time and Energy Consumption

We next investigate the authentication time required for AeroKey to extract a 128-bit final key, K. As previously presented, each th requires a different optimal error tolerance rate that leads to varying m. Table 4.3 presents the required measurement time (time required to run MEAS stage) as well as the length of the evidence bit sequence, m, for corresponding th based on its bit extraction rate. As th increases from 0.005 to 0.015, an increase in the BAR and lower required error tolerance rate leads to shorter m in both environments. Specifically, in home, m reduces from 287 to 213, respectively. However, contrary to the shorter length of m, the overall time spent on the MEAS stage increases due to lower bit extraction rate on higher th; approximately 22.0 s up to 25.9 s of measurement time is necessary. In lab, a relatively lower number of bits are required that ultimately results in a measurement time of 18.6 s to 22.4 s as th increases.

To further accurately estimate realistic overall authentication time accounting for computation, I implement AeroKey on four different devices: Google Pixel 2 (2.35-GHz), Google Pixel 3 (2.5-GHz), Raspberry Pi 4 (1.5-GHz) and Arduino Due (84-MHz). On each device, I measure the authentication time 50 times with $m = 287$ and $th = 0.005$ which exhibits the lowest authentication time among all th in home environment. As Figure 4.35(a) shows, Pixel 3 achieves the fastest mean authentication

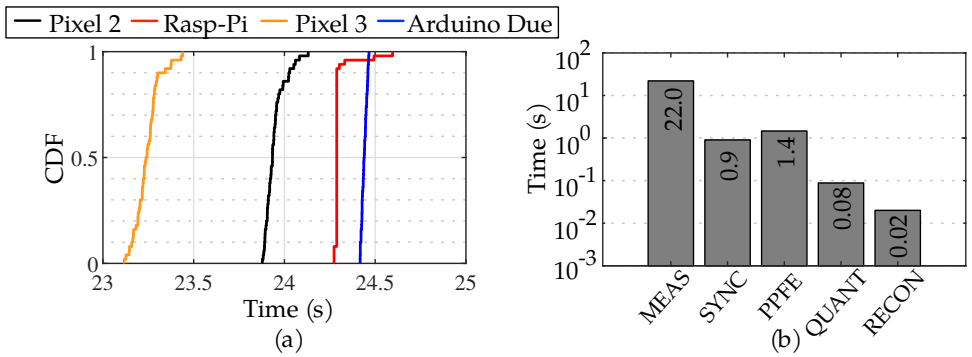


Figure 4.35: (a) Total authentication time measured on four different devices. (b) Execution time of each AeroKey stages on Arduino Due (log scale).

time of 23.2 s. While Pixel 2 and Raspberry Pi 4 exhibit a similar mean time of 23.9 s and 24.2 s, respectively, Arduino due exhibits the slowest mean authentication time of 24.4 s due to the relatively slower processor speed. The execution time of each AeroKey stage running on Arduino Due is illustrated in Figure **??**(b). The MEAS stage accounts for a majority of the authentication time (22.0 s), which is identical across all devices. The SYNC and PPFE stage takes 0.9 s and 1.4 s, respectively, due to the intensive computation (DTW) and filtering algorithms. The remaining stages (QUANT, and RECON) make up a negligible portion of less than 0.1 s in total. Overall, the results suggest that the authentication can take place within 24.5 s on various processors.

Next, to show the feasibility of running AeroKey on various battery-operated mobile devices, I investigate the energy consumption of four major stages (SYNC, PFFE, QUANT and RECON) on Arduino Due that exhibits the slowest authentication time. Because the MEAS stage merely utilizes the on board ADC, the energy consumption is negligible compared to running other stages (0.5 mJ, assuming 10-bit resolution at 24 kSPS). In order to measure accurate power usage, I subtract the idle power from

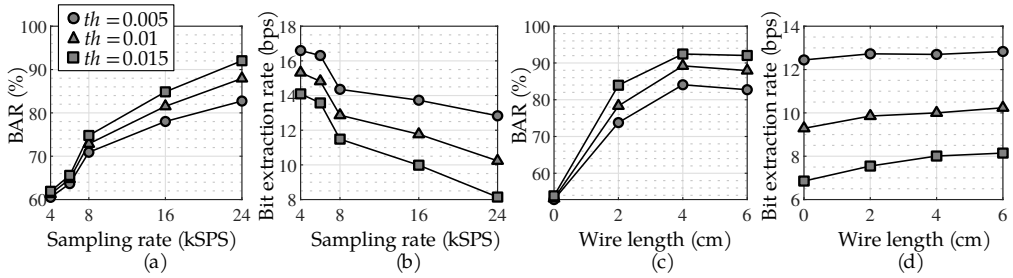


Figure 4.36: (a) BAR and (b) bit extraction rate under varying sampling rate between two closely located devices. (c) BAR and (d) bit extraction rate under varying wire length between two closely located devices.

the power readings when running four stages. On average, SYNC, PFFE, QUANT and RECON stages consume 13 mW of power. This leads to a total energy consumption of approximately 30 mJ, which makes AeroKey suitable for running under a modern device's battery capacity (i.e., a typical smartphone battery can hold 40 kJ).

Device Configurations

We next investigate the effect of the device configuration, including the ADC's sampling rate and length of the conductor wire connected to the input pin. We downsample the measured raw signals of two closely located devices (20 cm apart) from 24 kSPS down to 4 kSPS and compare its BAR and bit extraction rates. As Figure 4.36(a) illustrates, at 24 kSPS and 16 kSPS, BAR is over 80% for $\mathrm{th} \geqslant 0.01$, and falls below for 8 kSPS and 4 kSPS. To maintain BAR above 80% for reliable key reconciliation, the sampling rate should be at least 16 kSPS, which can easily be achieved by low-cost MCUs. In Figure 4.36(b), I illustrate the affects on the bit extraction rate. As the accuracy of the measured signal increases with a higher ADC sampling rate, it tends to pick up more noise and high frequency components that decrease the bit extraction rate. Nevertheless, 16 kSPS and 24 kSPS range that exhibits reasonable BAR can expect bit

extraction rate above 8.1 bps regardless of th.

Next, I examine the impact of the conductor wire's length. This is particularly important when AeroKey is to be implemented in various mobile devices with small form factor constraints. As shown in Figure 4.36(c), BAR is above 80% when the wire length is 4 cm or 6 cm for all $th \geqslant 0.005$. If the length decreases to 2 cm, BAR slightly drops, and when there is no wire connected to the ADC, significant degradation in BAR occurs to less than 60%. Therefore, to maintain high BAR, the conductor wire should be at least 4-cm long. As shown in Figure 4.36(d), contrarily to the sampling rate variation which is more sensitive to pick up high frequency signal, the wire length does not significantly impact the bit extraction rate. The length of 4 cm up to 6 cm can experience above 8 bps regardless of th.

Discussion

In the following, I discuss some constraints of the current design of AeroKey and the ways to improve it.

Authentication time: Using a commercial off-the-shelf MCU with no hardware modifications, a pair of AeroKey-enabled devices can successfully authenticate each other within 24.5 s. The majority of this time is spent on the MEAS stage, and the bottleneck that constrains the overall authentication time of the system is mainly attributed to two reasons. First, because AeroKey targets low-cost IoT devices, typically with an on-chip ADC (low sampling frequency and resolution) with no hardware modifications (amplifier and analog front-end circuitry), the measured raw signal is very noisy and does not highly correlate between two devices. To solve this, MEAS stage has to collect an enough number of 60 Hz periods within each cycle for the PPFE stage to extract a highly correlating noise signal between two close devices as shown in Figure 4.24(a). Secondly, to guarantee a robust security of 128-bit final key, the length of the environmental bit sequences needs to be significantly longer (over 200 bits) due

Table 4.4: NIST test results of AeroKey (p-value $\geqq 0.05$)

Test	p**-value**	**Test**	p**-value**
Frequency	0.3504	FFT	0.2133
Block frequency	0.5341	Nonperiodic temp. match	0.7399
Cumulative sums	0.0668	Overlapping temp. match	0.3504
Runs	0.3504	Serial	0.9114
Longest runs of ones	0.1223	Linear complexity	0.3504
Rank	0.3504		

to RECON stage that incurs some entropy loss. Nevertheless, if AeroKey is used for simple device pairing purposes (e.g., Bluetooth pairing mechanism requires two devices to agree on a 32-bit pin), users can expect to pair the devices with a relatively fast pairing time of under 5 s, regardless of th. This is significantly faster compared to the usability study that presented average typical pairing time of 27 s to copy and confirm 8-digit pin in IoT devices with limited user interface [67].

Randomness of generated bits: Just like every other cryptographic bit sequence or key, it is important for the AeroKey-generated evidence bits to be random and unpredictable. To examine the randomness, I conduct the NIST statistical suites test [4] on the generated bit sequences with a length of over one million bits. The NIST test consists of 15 randomness tests, each with an output p-value. Table 4.4 presents AeroKey's NIST test results. AeroKey passes 11 out of 15 tests with p-value greater than 0.05. However, to be considered as a true random number generating source, AeroKey should pass all 15 tests with p $\geqq 0.05$. Therefore, further random bit corrections are necessary to produce higher quality evidence bit sequences [70, 32].

Radius of PAR: The evaluation suggests that the typical range of the PAR can be as large as one meter. The concept of Transitivity of Trust (ToT)

protocol [24] can be applied to increase the radius and authenticate devices within the entire household. In such case, the chain of authentication takes place through bridging devices deployed within different rooms. On the other hand, the user may wish to further decrease the range to be resilient against dedicated adversary. In these cases, AeroKey can be configured with a higher th and/or with a lower error tolerance rate to the point where the user feels comfortable with the tuned authentication range.

Outdoor applicability: The underlying assumption of the AeroKey protocol is that authenticating devices are located indoors where EMR is observable. Therefore, traditional password-based authentication should be used to authenticate devices in outdoor scenarios (e.g., in the vehicle or on the street).

Orientation of antenna: Our experiments are conducted in the setting where the conductive wires connected to the authenticating MCUs are oriented parallel to one another. However, because the EMR pattern propagates in three orthogonal planes (x, y and z) [66], the authenticating devices may experience signal correlation issues when one device is not properly oriented with respect to another authenticating device. To potentially deal with this issue, multiple orthogonally aligned antenna (with PCB trace or in the form of patch antenna) can be designed to receive spherical radiation pattern (isotropic measurement) while preserving the form-factor of the devices.

4.2.4 Conclusion

The ever-growing IoT ecosystem is demanding secure yet usable device authentication methods to ensure trustworthy wireless connectivity between a large number of small, low-cost and low-power devices. ZIA is a promising solution to achieve both security and usability by generating random bit sequences to be used as a security key from an ambient source of ran-

domness. This Section presented AeroKey, a secure and usable device authentication scheme leveraging spatiotemporally unique ambient EMR signature. This is the first work that realizes ZIA without any dedicated sensor, which makes it favorable for application in virtually any IoT device. With novel pre-processing and feature extraction techniques, AeroKey allows co-located devices to autonomously generate near-identical and high-entropy evidence bit sequences from noisy measurement signals on off-the-shelf MCUs with zero hardware modifications. The evaluation successfully demonstrated high authentication success rates between devices within a small PAR deployed in home and lab environments with relatively low authentication time of as low as 24 s. Also, AeroKey shows its robustness against various adversaries leveraging time, space, electrical appliances and ML models with a low EER of 3.4% or less under different environments. To summarize, AeroKey enables low-cost IoT and mobile devices to periodically (re-)authenticate themselves to facilitate secure, usable, and practical device authentication with no human involvement.

5 BALANCING BETWEEN SECURITY AND USABILITY IN ZIA

While various ZIA techniques have been proposed in many different works, the focus has mainly been placed on the modalities of sensing, i.e., the sources of entropy to create the keys. However, the signal processing pipeline that generates the keys from the measurement has received less attention and has been rather ad hoc. The pipeline typically consists of stages for measurement, bit quantization, and key reconciliation, which are highly tailored and designed toward a specific sensing modality. A signal processing stage designed for one technique does not work for others, and the lack of a "systematic" pipeline design is a large missing piece of ZIA techniques. In the following Section, I introduce a framework designed for ZIA users and developers to help them quickly and efficiently determine an optimal parameter that can be used to balance security and usability of any ZIA techniques.

In this work, as a first step towards exploring a systematic approach for signal processing in ZIA techniques, I investigate and focus on the key reconciliation stage among other stages that is most sensitive to parameter tuning and most computationally heavy. More specifically, I propose a generic framework that automatically determines the reconciliation parameter that balances security and usability, given a user-defined authentication range. The proposed framework can serve as a guideline for ZIA developers when coming up with new techniques or be implemented in existing ZIA works to seamlessly determine proper parameter values. Additionally, I analyze the two most commonly used key reconciliation schemes (leveraging error-correcting codes and compressed sensing) in terms of their reconciliation rates, entropy loss, and computation costs based on varying parameters to determine a standard scheme to be used

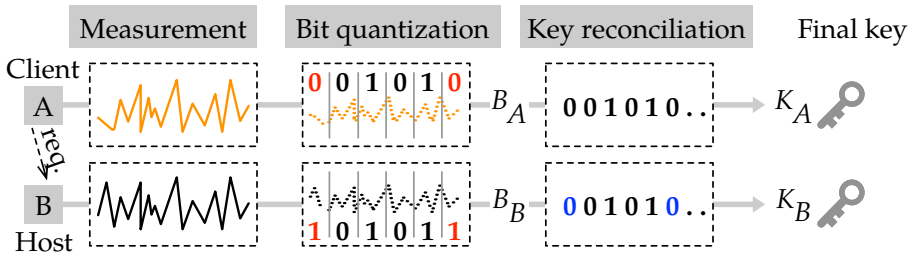


Figure 5.1: General pipeline of ZIA techniques between two Devices: A (Client) and B (Host).

in the framework.

5.1 SECURITY AND USABILITY OF ZIA

Signal Processing Pipeline

The security and usability of a ZIA technique are sensitive to the design of its signal processing pipeline. A typical pipeline of ZIA techniques is illustrated in Figure 5.1. The pipeline consists of three general stages ("Measurement", "Bit quantization", and "Key reconciliation") to generate the final authentication key. We assume that Device A (Client) initiates the process by sending an authentication request to Device B (Host).

Measurement: Two devices independently measure a surrounding noise context using their embedded sensors.

Bit quantization: The noise measurements are quantized into a sequence of 1s and 0s using statistical features in the time or frequency domain. We refer to the quantized bit sequences as *environmental bit sequences*, B_A and B_B, and further denote their length with b. Even the bit sequences from two very closely located devices will likely exhibit occasional bit errors due to various factors including sensor variations, random errors, etc.

Key reconciliation: The bit errors between B_A and B_B are reconciled

using a reconciliation scheme to produce identical *final keys*, K_A and K_B, of bit length k. During the reconciliation, partial information about B_A and B_B are exchanged over the public channel that incurs some entropy loss which results in K exhibiting less entropy than B. It is important to ensure that the exchanged information does not leak enough information to fully derive B nor K because the public channel is always assumed to be eavesdropped by malicious adversaries trying to infer K.

Balancing Security and Usability

Authentication is deemed successful only when the two resulting final keys are identical with no bit errors. I define all devices within the authentication range from each other as legitimate devices and further refer to devices located outside the range as adversarial devices. Note that the only difference between the legitimate and adversarial devices is whether they are located in or outside the authentication range. As a metric to compare the similarity of two bit sequences (i.e., B or K), I use *bit agreement rate* (BAR), which refers to the rate of matching bits. To quantify usability, I use *true acceptance rate* (TAR), referring to the rate of successful authentication between the legitimate devices. The security of the system is quantified with *false acceptance rate* (FAR), which is the rate of successful authentication of adversarial devices. The overall usability and security balance is measured with *equal-error rate* (EER), which refers to the intersection point where FAR is equivalent to false rejection rate (FRR=100−TAR). A low ERR represents better accuracy of distinguishing legitimate devices over adversarial ones.

Because a device's physical location is a proxy for its legitimacy in ZIA, it is important to be able to strictly control the authentication range at which devices should be allowed to authenticate with one another. The parameters of the key reconciliation scheme determine its error tolerance threshold, which in turn depicts the boundary of the authentication range

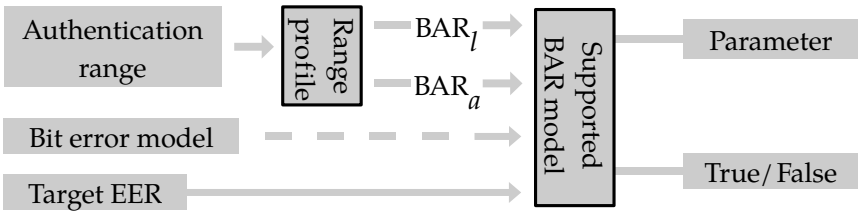


Figure 5.2: Framework to determine reconciliation parameter given three user inputs: authentication range, bit error model (optional), and target EER.

after two authenticating devices translate its measured environmental noise into bit sequences; if and only if the two bit sequences exhibit fewer errors than the threshold for the given parameters, identical keys can be derived to authenticate two devices. As such, the reconciliation parameters should be carefully chosen because the parameters that correspond to a high error tolerance threshold will allow distant adversarial devices to be able to authenticate (high FAR), while a too low threshold will suffer from low usability (high TRR). To strike this sensible balance between security and usability, it is crucial to understand and determine proper reconciliation parameters for different existing reconciliation schemes.

Different ZIA techniques use different forms of environmental noise with different spatial extents. Generally, as the distance between the two devices increases, BAR between their environmental bit sequences decreases due to spatially unique nature of the noises. For instance, in VoltKey, which utilizes powerline noises as a source of entropy, BAR decreases from 95% to 91% as the distance between the devices increases from 1 m to 24 m [38]. To balance security and usability for a given authentication range, the key reconciliation parameters must be judiciously fine-tuned to successfully authenticate only when BAR is higher than or equal to the BAR of the desired range. While most previous works presented new methods and different noise sources that ZIA can leverage, the problem of automatically determining optimal reconciliation parameters for varying

authentication range is yet to exist.

I propose a systematic framework to analyze and determine the parameters given an *authentication range* and a *target EER* defined by either the manufacturer or the user. The overall diagram of the framework is illustrated in Figure 5.2. As inputs, it takes in the desired authentication range and a target EER of the system. The bit-error model between B_A and B_B (e.g., burst error model) is an optional input. With the desired range, the "Range profile" block determines the minimum BAR that is required between the legitimate devices, BAR_l, and the maximum BAR that is exhibited by devices outside the range, BAR_a. The Range profile block is pre-profiled with the varying distance and its corresponding BAR between B_A and B_B, which may differ from one noise source to another. Once profiled, BAR_l is selected as the corresponding BAR under the given range, and BAR_a is selected as the highest BAR achieved beyond the range. With the selected BAR_l and BAR_a, the "Supported BAR model" block, which is also pre-profiled with varying reconciliation parameters and its reconciliation performance, outputs the optimal parameter. Additionally, it confirms if the user-given target EER can be met. Note that this block can be based on different existing key reconciliation schemes leveraging error-correcting codes or compressed sensing, which will be described in the next section. This framework can particularly benefit ZIA developers to quickly determine proper reconciliation parameter (regardless of noise sources) that balances usability and security (EER) without manually computing through different parameter values and validating the EER point.

5.2 KEY RECONCILIATION PROTOCOLS

In this part, I describe two commonly used key reconciliation schemes in ZIA research to understand how their parameters affect the performance

Table 5.1: Error-correcting code based reconciliation (Fuzzy commitment) [33, 24, 43, 50, 51, 58]

	Device A		**Device** B
①	$K_A = \mathsf{PRNG}(k)$		
②	$H(K_A) = \mathsf{SHA256}(K_A)$		
③	$\lambda_A = \mathsf{ENCODE}(K_A)$		
④	$\sigma = B_A \oplus \lambda_A$	$\xrightarrow{\sigma, H(K_A)}$	$\lambda_B = B_B \oplus \sigma$
⑤			$K_B = \mathsf{DECODE}(\lambda_B)$
⑥			$H(K_A) \stackrel{?}{=} H(K_B)$

of ZIA techniques. One scheme is based mainly on *error-correcting code* (*ECC*) and the other leverages the notion of *compressed sensing* (*CS*) to reconcile differences in bit sequences.

ECC-based Reconciliation

ECC-based reconciliation, also commonly known as *fuzzy commitment* scheme, leverages linear ECC to allow two very similar bit sequences as evidence to lock and unlock a "secret" so that it can be transferred over public wireless channel [33]. In the case of ZIA, the "secret" is the final key, K, and the two environmental bit sequences from two devices are used to lock and unlock K.

The overall reconciliation steps are illustrated in Table 5.1. First, Device A uses a pseudo-random number generator to generate a random bit sequence of length k to be used as a final key, K_A. It then performs one-way hashing on the result (e.g., SHA256) to generate $H(K_A)$. Afterward, K_A is encoded into a codeword, λ_A, of length b using an ECC (e.g., Reed-Solomon(b,k)) to add redundancy based on the polynomials over Galois fields to support error correction. Then, it performs exclusive OR between λ_A and extracted B_A to obtain σ, which is transferred to Device B

Table 5.2: Compressed sensing based reconciliation [73, 46, 45, 71]

	Device A		**Device** B
①	$K_A = B_A$		
②	$H(K_A) = SHA256(K_A)$		$C_B = \Phi \cdot K_B$
③	$C_A = \Phi \cdot K_A$	$\xrightarrow{C_A, H(K_A)}$	$\Delta C = C_A - C_B$
④			$\Delta B = l1.min(\Delta C)$
⑤			$K_B = B_B \oplus \Delta B$
⑥			$H(K_A) \stackrel{?}{=} H(K_B)$

through a public channel along with $H(K_A)$. Note that σ does not divulge any information about B_A nor λ_A to the malicious eavesdropper on the public channel. Once Device B receives σ, it performs exclusive OR with its own extracted B_B to produce λ_B which is very close to originally encoded λ_A. Afterward, λ_B is decoded using the same ECC into K_B. To verify the equality of the two keys obtained on two devices, K_A and K_B, Device B compares the two hashed results; if equal, two devices are successfully authenticated. The fuzzy commitment scheme assumes identical derivation of the key as long as the number of bit errors (Hamming distance) between B_A and B_B does not exceed $\frac{b-k}{2}$ bits, and I denote this number of bit error as T. Note that the added redundancy to correct the error between the B_A and B_B results in final key length k to be shorter than b ($k < b$). In this work, I use T as a main parameter to benchmark the ECC-based reconciliation.

CS-based Reconciliation

CS is an information processing technique to reconstruct a high-dimensional sparse signal from a low number of measurements. A CS-based reconciliation scheme mainly uses this property to reconstruct the sparse error between B_A and B_B as they are very close to each other in terms of their

Hamming distance. The overall steps are illustrated in Table 5.2. Unlike ECC-based reconciliation, Devices A and B directly use the extracted B_A and B_B to be used as K_A and K_B, respectively, which results in $k = b$. Device A further obtains the hashed key to later verify the equality with K_B. Then, K_A and K_B are compressed into C_A and C_B, respectively, with Φ representing the random Bernoulli matrix with ± 1 with equal probability. The Φ represents a randomly generated sensing matrix that is assumed to be identical on both devices (can be pre-established) and the dimension of Φ specifies the compression rate; $k \times M$ matrix results in the length of C_A to be M bits. Afterward, Device A transfers C_A, along with $H(K_A)$ to Device B. Upon receiving, Device B obtains ΔC by subtracting the two C which essentially represents the difference between B_A and B_B in a lower dimension. Afterward, the difference, ΔB, is recovered to a higher dimension signal by solving the l_1 minimization problem of ΔC, which regularizes the problems by using the sparsity of the solution. Then, K_B can be derived by performing exclusive OR between ΔB and B_B. Later in this Section, I use M as a main parameter to benchmark the CS-based reconciliation.

5.3 ANALYSIS OF KEY RECONCILIATION SCHEMES

In this section, I benchmark the two above-mentioned key reconciliation schemes in terms of their reconciliation performance, entropy losses, and computation overheads. For all evaluations, I assume $b = 128$ bits and vary the parameters T and M for ECC- and CS-based reconciliation, respectively.

Reconciliation Success Rate

I first compare how the key reconciliation parameter affects the performance of the two reconciliation schemes. As an evaluation metric, I use

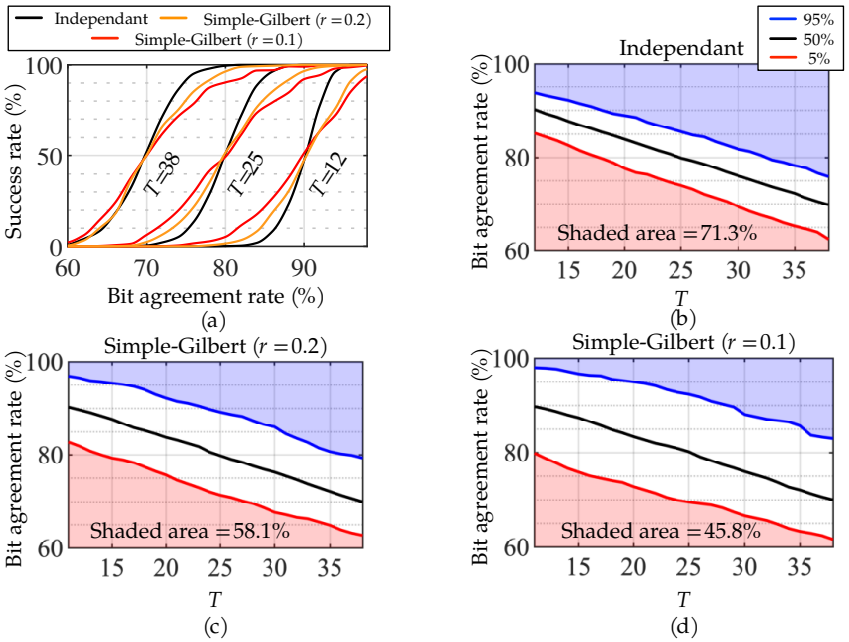


Figure 5.3: (a) Success rate for ECC-based scheme varying T and required BAR between devices to achieve 5%, 50% and 95% success rate in (b) independent and (c) simple Gilbert model with r=0.2 and (d) r=0.1.

success rate, which refers to the rate of successful authentication (100% BAR between K_A and K_B) to all authentication attempts. To simulate two environmental bit sequences, B_A and B_B, with varying BAR, I first use a pseudo-random number generator to obtain B_A and consider three error characteristics to obtain B_B with two different bit error models: the *independent* model and the *simple Gilbert* model [19]. In the independent error model, every bit has an equal bit error rate equivalent to the given bit error rate (i.e., 100%-BAR). In the simple Gilbert model, the bit errors are based on the two-state (good and bad states) Markov approach, where each of them may generate errors based on its state (good: 0%, bad: 50%). I use r (the probability of transitioning from bad to good state) of 0.1 (more bursty) and 0.2 (less bursty) to simulate different error burstiness. Using the three different error characteristics, I vary the BAR from 60% to 98%

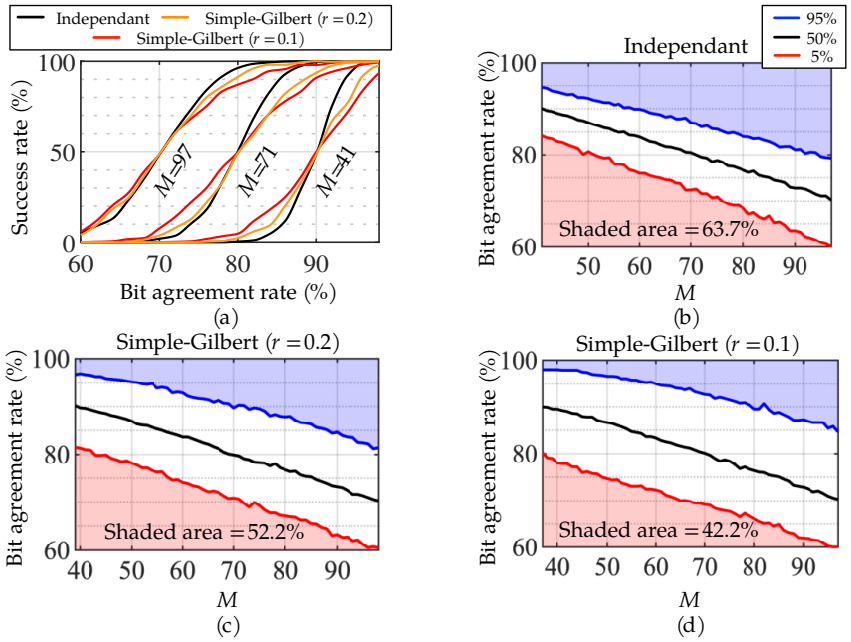


Figure 5.4: (a) Success rate for CS-based scheme varying M and required BAR between devices to achieve 5%, 50% and 95% success rate in (b) independent and (c) simple Gilbert model with r=0.2 and (d) r=0.1.

on a sequence of 100,000 bits.

Figures 5.3(a) and 5.4(a) illustrate the success rate of ECC- and CS-based reconciliation, respectively, for varying parameters. Since BAR monotonically decreases as the inter-device distance increases, an ideal success rate curve should have a steep cut-off slope at the BAR that corresponds to the target distance. This would allow devices with a BAR above the threshold to always succeed in authenticating and reject devices with a BAR below the threshold. However, in reality, the success rate curve show a gradually sloping curve because the BAR distribution of B within the specified mean BAR exhibit a normal distribution. In all cases, a burstier error results in a flatter slope. This is because the BAR variances of each B with $b = 128$ are higher for burstier error models even though all three error characteristics exhibit an identical mean BAR. In both schemes, as

the error tolerance parameter, T or M, increases, the success rates increase for given BAR. This means that an increase in T and M can tolerate more number of bit errors between devices. Specifically, on ECC-based reconciliation, $T = 12$ achieves a 50% success rate when the BAR between the devices is at 90% under different error characteristics. As T increases to 38, the same 50% success rate is achieved when the two devices only exhibit a 70% BAR. Similar performance is achieved on CS-based reconciliation with $M = 41$ and 97 that can support 90% and 70% BAR, respectively.

Figures 5.3(b), (c) and (d) illustrate the required BAR between devices to achieve 5%, 50% and 95% success rate for varying parameters (equivalently supporting 70% to 90% BAR) for each error characteristics with the EEC-based reconciliation; and Figures 5.4(b), (c) and (d) illustrate the same for the CS-based reconciliation. For each parameter, T or M, the shaded area above 95% line (blue) indicates the supported BAR between legitimate devices (BAR_l) to achieve the minimum TAR of 95%. On the other hand, the shaded area below 5% line (red) indicates the supported BAR from the adversarial devices (BAR_a) to achieve a maximum FAR of 5%. This indicates that as long as BAR_l is above the blue line and BAR_a is below the red line for a specific parameter, the overall system can expect 5% EER or less (over 95% TAR and under 5% FAR). We quantize the rate of the shaded area over the plotted region for comparison. A larger area indicates better reconciliation performance with a more range of BAR to potentially exhibit 5% or less EER. In both schemes, the burstier error results in a less shaded area due to a flatter slope of the success rate as previously mentioned. Compared to the CS-based scheme, the ECC-based reconciliation has a higher rate of shaded area with 71.3%, 58.1% and 45.8% for independent, and Simple Gilbert models with $r = 0.2$ and 0.1, respectively.

These plots can be implemented as a Supported BAR model in the envisioned framework to output optimal reconciliation parameters. The

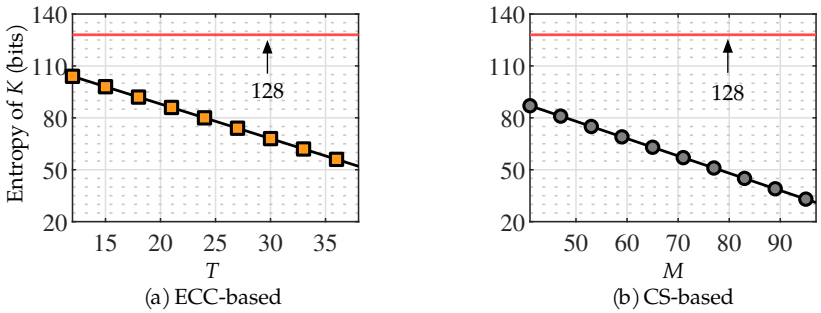


Figure 5.5: Entropy of the final key, K, using (a) ECC-based and (b) CS-based reconciliation schemes.

black line, which represents the 50% success rate for varying parameters, is the guideline for selecting the optimal parameter. The parameter is chosen by taking the mean BAR between BAR_l and BAR_a, and selecting the corresponding BAR in the black line. If the given BAR_l and BAR_a are within (above and below) the shaded region, the target EER can be met. Otherwise, the target EER cannot be met and the authentication range must be decreased. While these plots only represent 5% EER (95% and 5% success rate line), further plots representing varying success rates can be pre-profiled within the Supported BAR model to verify the validity of the target EER.

Entropy Loss

We next evaluate the quality of the final key, K, generated from B in terms of the resulting entropy. Note that the length of B is fixed to 128 bits ($b = 128$). In ECC-based reconciliation, the bit length of the final key, k is obtained by taking $b - 2T$ (assuming Reed-Solomon code). In CS-based scheme, k is equivalent to b because B is directly utilized as K. If only considering the key length, CS-based scheme retains more entropy. However, in CS-based scheme, when C_A is transferred over to Device B over a public channel, partial information is leaked to the potential adversary. Assuming a strong

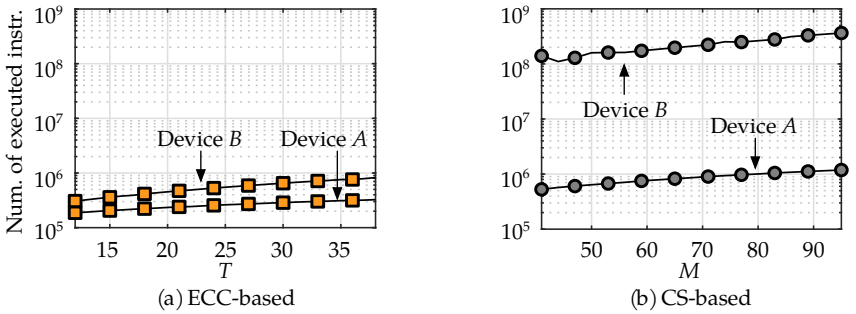


Figure 5.6: Number of executed instructions under (a) ECC-based and (b) CS-based reconciliation schemes.

attack model where the adversary has access to a pre-established sensing matrix Φ, I have to assume that M bits out of k bits have been leaked. Considering this information leakage, I present the resulting entropy of K from the two reconciliation schemes in Figure 5.5. The parameters T and M are set to 12–38 and 41–96, respectively, representing the equivalent reconciliation performance derived from the previous analysis. In both schemes, as T or M increases to tolerate higher error, the resulting entropy decreases linearly. Specifically, in the ECC-based scheme with $T = 12$, I can expect the final key to retain 104 bits of entropy, whereas with $T = 38$ that can tolerate as low as 70% BAR retains 52 bits of the entropy. Overall, the CS-based reconciliation experiences higher entropy loss within the same reconciliation performance range. When $M = 41$ to support 90% BAR, the resulting entropy of the final key is 87 bits. An increase in M up to 96 will result in K retaining only 32 bits. Considering the two equal reconciliation performance, ECC-based scheme results in retaining more bits of entropy in the final key.

Computation Overhead

Finally, I compare the two schemes in terms of their computation costs. We implement the two schemes in C language on both client (Device

A) and host (Device B) sides. As the main metrics, I use the *number of executed instructions* and the *execution time* measured using Linux `perf` command on the Raspberry Pi 4 equipped with an ARM Cortex-A72 1.4 GHz processor. The benchmark results of the schemes with varying T and M are illustrated in Figure 5.6(a) and (b), respectively (note the log scale on the y-axis). In both schemes, an increase in both T and M require a higher number of executed instructions due to the higher number of bit errors it can tolerate. In the ECC-based scheme, Device A merely performs an encoding function (involving linear operations to add redundancies) that results in execution of 180k to 320k (1.4 to 1.5 ms of execution time) instructions as T increases from 12 to 38. On the other hand, Device B performs more number of instructions, ranging from 300k to 800k (1.5 to 3.2 ms of execution time), due to the higher computation costs of the decoding function. Nevertheless, both sides can execute under 3.2 ms regardless of the increase in T. The CS-based scheme, on the other hand, is orders-of-magnitude more computation-intensive. In Device A, $k \times M$ matrix multiplication using Φ results in a relatively higher number of executed instructions, ranging from 500k to 1,000k as M increases from 41 to 97, respectively. This results in an execution time of 1.5 to 2.7 ms. On the other hand, Device B requires orders of magnitude more number of instructions ranging from 128M to 363M (48 to 100 ms of execution time) due to the heavy computation of the l1 minimization algorithm to recover sparse vector.

Overall, ECC-based reconciliation is computationally lighter and faster than CS-based scheme for the same reconciliation performance. Also, in both schemes, the computation overhead of the host and the client is significantly different, especially in the CS-based scheme. This asymmetry must be considered when implementing a ZIA technique on resource-constrained devices so that the device with more resource play the host role.

5.4 CONCLUSION

While many prior works successfully propose new sensing modalities, they do not explicitly state the definition of co-location or provide the parameter values for different authentication ranges, which often leave the users to determine the right values based on different deployment environments. Further, the lack of comprehensive comparison of different key reconciliation schemes leads to different groups of work using different schemes, which hinders the coherence and fair comparison of many ZIA techniques. As a first step towards developing a systematic ZIA pipeline, I proposed a novel framework to determine the proper reconciliation parameter for the desired authentication range. We analyzed the two commonly used reconciliation schemes in terms of their security, usability, and computation overhead to find a more suitable scheme for optimizing ZIA operation. The proposed framework can be embedded in many resource-constrained devices to dynamically adjust its authentication range and adapt based on its deployed environment.

6 FUTURE WORKS

The difficulty of authenticating heterogeneous IoT devices will continue to present security and usability challenges in the future as the number of devices that the user must manage continue to increase. While there has been an increasing number of research activities in ZIA field in the last several years, there still remain significant challenges for ZIA to be widely adopted in commercial devices. In this Section, I discuss future research agenda addressing remaining challenges.

6.1 SENSING HARDWARE VARIATION

One of the main drawbacks that hinders ZIA from being implemented in commercial devices is the sensing hardware variation that leads to varying measurement results across devices. For instance, in the case of using ambient sound as environmental context to generate key, different device manufacturers may embed different microphone hardware with diverse settings or calibration methods that leads to incomparable sound signal across authenticating devices. Towards the goal of expanding the ZIA to commercial applications, stronger understanding of the various environmental contexts are necessary to successfully capture the invariant property that can be sensed with different device and sensor configurations.

6.2 USABILITY AND PRIVACY OF ZIA

Currently there is no profound, real-world usability study of different ZIA mechanisms to identify the most usable context for the users. For instance, in SyncVibe and ivPair, the users are required to put devices in direct contact while the authentication takes place. Comparatively,

authentication using VoltKey and AeroKey require no action from the user with enhanced usability. For coherent and fair comparison of using different contexts, solid usability metrics need to be defined and developed and through user study needs to be conducted. Similarly, privacy implications of ZIA need to be studied. For instance, some users might not be comfortable with devices gathering environmental contexts such as audio or luminosity for autonomous device authentication purposes as it may contain information that can lead to activity recognition. Therefore, comprehensive user study is deemed necessary addressing usability and privacy of ZIA.

6.3 THOROUGH EVALUATION OF THE SECURITY PROPERTIES OF ZIA

Another remaining challenge in ZIA is the lack of understanding of security properties of ZIA against dedicated adversary. For instance, injection attacks on the authentication system using ambient sounds can particularly be vulnerable to a distant adversary with high-powered directional speakers. To address this challenge, machine learning based defense model based on previously measured contextual information can specifically be advantageous. Leveraging this, the model can detect environmental anomalies caused by the external attacks, which can be used to effectively reject authentication from malicious signals and significantly improve the overall security of the ZIA process. Also sensor fusion approach leveraging different sensing modalities can also be effective in detecting maliciously injected signals.

7 CONCLUSION

In this dissertation, I have presented series of four ZIA methods towards the goal of improving security, usability, and practicality over currently employed password-based authentication method. Specifically targeting mobile devices, I introduced SyncVibe and ivPair, leveraging induced or inherent vibration, measured through direct contact of devices, to quickly and effectively authenticate two devices with no prior knowledge. Due to the ubiquitous nature of the accelerometer in mobile devices, the proposed works can easily be implemented in commercial devices without any hardware level modifications to facilitate more spontaneous and secure association of devices in various mobile scenarios. Furthermore, I presented two truly zero-interaction device authentication method named VoltKey and AeroKey, designed to authenticate indoor IoT devices. The proposed works effectively eliminates the "human-assisted" aspect during the authentication process by using omnipresent indoor contexts (power line noise and ambient EMR), ultimately resulting in higher overall security and usability: the randomly generated keys provide higher security than user-created passwords do, and it does not depend on the user to create, remember, and enter the password, which allow devices to autonomously re-establish fresh keys and authenticate more frequently. Finally, addressing the limitations that I came across during development of multiple ZIA techniques, I proposed a generic key reconciliation framework that can be used in any ZIA work to determine the proper reconciliation parameter based on a user-given authentication range, which is a first work that targets to develop systematic design optimization of ZIA's signal processing pipeline. Overall, the proposed works effectively address the current usability problems of the device authentication in IoT systems so that people who have limited to no skills to operate computers can easily keep a secure internet connected environment.

DISCARD THIS PAGE

BIBLIOGRAPHY

[1] Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, May 2005.

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

[3] S Abhishek Anand and Nitesh Saxena. Coresident evil: Noisy vibrational pairing in the face of co-located acoustic eavesdropping. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 173–183, 2017.

[4] Lawrence E. Bassham, III, Andrew L. Rukhin, Juan Soto, James R. Nechvatal, Miles E. Smid, Elaine B. Barker, Stefan D. Leigh, Mark Levenson, Mark Vangel, David L. Banks, Nathanael Alan Heckert, James F. Dray, and San Vo. Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Gaithersburg, MD, United States, 2010.

[5] Taha Belkhouja, Xiaojiang Du, Amr Mohamed, Abdulla K. Al-Ali, and Mohsen Guizani. Biometric-based authentication scheme for implantable medical devices during emergency situations. *Future Generation Computer Systems*, 98:109–119, 2019.

[6] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the International Conference on Knowledge Discovery*, pages 359–370, 1994.

[7] Chongguang Bi and Guoliang Xing. Real-time attitude and motion tracking for mobile device in moving vehicle. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 357–358, 2018.

[8] Daniel Bichler, Guido Stromberg, and Mario Huemer. Innovative key generation approach to encrypt wireless communication in personal area networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOCOM)*, pages 177–181, 2007.

[9] Daniel Bichler, Guido Stromberg, Mario Huemer, and Manuel Löw. Key generation based on acceleration data of shaking processes. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, pages 304–317, 2007.

[10] Morgan H. L. Chan and Robert W. Donaldson. Amplitude, width, and interarrival distributions for noise impulses on intrabuilding power line communication networks. *IEEE Transactions on Electromagnetic Compatibility*, 31(3):320–323, August 1989.

[11] Gabe Cohn, Erich Stuntebeck, Jagdish Pandey, Brian Otis, Gregory D. Abowd, and Shwetak N. Patel. Snupi: Sensor nodes utilizing powerline infrastructure. In *Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp)*, pages 159–168, 2010.

[12] John Dunning. Taming the blue beast: A survey of bluetooth based threats. *IEEE Security and Privacy (S&P)*, 8(2):20–27, March 2010.

[13] Sead Fadilpašić. Nearly all iot traffic is unencrypted, 2020.

[14] Mohammed Ferdjallah and Ronald E. Barr. Adaptive digital notch filter design on the unit circle for the removal of powerline noise from biomedical signals. *IEEE Transactions on Biomedical Engineering*, 41(6):529–536, June 1994.

[15] Mikhail Fomichev, Flor Álvarez, Daniel Steinmetzer, Paul Gardner-Stephen, and Matthias Hollick. Survey and systematization of secure device pairing. *IEEE Communications Surveys Tutorials*, 20(1), 2018.

[16] Mikhail Fomichev, Julia Hesse, Lars Almon, Timm Lippert, Jun Han, and Matthias Hollick. Fastzip: Faster and more secure zero-interaction pairing. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (*MobiSys*), pages 440–452, New York, NY, USA, 2021. Association for Computing Machinery.

[17] Mikhail Fomichev, Max Maass, Lars Almon, Alejandro Molina, and Matthias Hollick. Perils of zero-interaction security in the internet of things. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (*IMWUT*), 3(1), March 2019.

[18] Carlos Garrido, Antonio F. Otero, and Jose Cidras. Low-frequency magnetic fields from electrical appliances and power lines. *IEEE Transactions on Power Delivery*, 18(4):1310–1319, October 2003.

[19] E. N. Gilbert. Capacity of a burst-noise channel. *The Bell System Technical Journal*, 39(5), 1960.

[20] Gizmodo. A creepy website is streaming from 73,000 private security cameras, 2014.

[21] Prosanta Gope and Biplab Sikdar. Lightweight and privacy-preserving two-factor authentication scheme for iot devices. *IEEE Internet of Things Journal*, 6(1):580–589, 2019.

[22] Core Specification Working Group. *Bluetooth Core Specification*, 12 2019. Rev. 5.2.

[23] Bogdan Groza, Adriana Berdich, Camil Jichici, and Rene Mayrhofer. Secure accelerometer-based pairing of mobile devices in multi-modal transport. *IEEE Access*, 8:9246–9259, 2020.

[24] Jun Han, Albert Jin Chung, Manal Kumar Sinha, Madhumitha Harishankar, Shijia Pan, Hae Young Noh, Pei Zhang, and Patrick Tague. Do you feel what i hear? enabling autonomous iot device pairing using different sensor types (s&p). In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 836–852, 2018.

[25] Jun Han, Madhumitha Harishankar, Xiao Wang, Albert Jin Chung, and Patrick Tague. Convoy: Physical context verification for vehicle platoon admission. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications* (*HotMobile*), pages 73–78, 2017.

[26] John E. Hershey, Amer A. Hassan, and Rao Yarlagadda. Unconventional cryptographic keying variable management. *IEEE Transactions on Communications*, 43(1):3–6, January 1995.

[27] Microchip Technology Inc. Sam d5x/e5x family data sheet, 2018.

[28] Schneider Electric Inc. Product data sheet hom115, 2019.

[29] Federal Trade Commission Consumer Information. Using ip cameras safely, 2013.

[30] Suman Jana, Sriram Nandha Premnath, Mike Clark, Sneha K. Kasera, Neal Patwari, and Srikanth V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the Annual International Conference on Mobile Computing and Networking* (*MobiCom*), pages 321–332, 2009.

[31] Wenqiang Jin, Ming Li, Srinivasan Murali, and Linke Guo. Harnessing the ambient radio frequency noise for wearable device pairing.

In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1135–1148, 2020.

[32] Nick Jones and LluÍs Masanes. Key distillation and the secret-bit fraction. *IEEE Transactions on Information Theory*, 54(2):680–691, 2008.

[33] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 28–36, 1999.

[34] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. Ensemble: Cooperative proximity-based authentication. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys*, pages 331–344, 2010.

[35] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. Sound-proof: Usable two-factor authentication based on ambient sound. In *Proceedings of the USENIX Security Symposium (USENIX Security '15)*, pages 483–498, August 2015.

[36] Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K. Jha, and Anand Raghunathan. Vibration-based secure side channel for medical devices. In *Proceedings of the 52nd Annual Design Automation Conference (DAC)*, pages 32:1–32:6, 2015.

[37] Kyuin Lee and Younghyun Kim. Balancing security and usability of zero-interaction pairing and authentication for the internet-of-things. In *Proceedings of the 2nd Workshop on CPS&IoT Security and Privacy*, pages 29–34, 2021.

[38] Kyuin Lee, Neil Klingensmith, Suman Banerjee, and Younghyun Kim. Voltkey: Continuous secret key generation based on power line noise for zero-involvement pairing and authentication. *Proceedings of the*

ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 3(3), September 2019.

[39] Kyuin Lee, Neil Klingensmith, Dong He, Suman Banerjee, and Younghyun Kim. ivPair: Context-based fast intra-vehicle device pairing for secure wireless connectivity. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, pages 25–30, 2020.

[40] Kyuin Lee, Vijay Raghunathan, Anand Raghunathan, and Younghyun Kim. Syncvibe: Fast and secure device pairing through physical vibration on commodity smartphones. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pages 234–241, 2018.

[41] Kyuin Lee, Yucheng Yang, Omkar Prabhune, Aishwarya Lekshmi Chithra, Jack West, Kassem Fawaz, Neil Klingensmith, Suman Banerjee, and Younghyun Kim. Aerokey: Using ambient electromagnetic radiation for secure and usable wireless device authentication. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1), March 2022.

[42] Jonathan Lester, Blake Hannaford, and Gaetano Borriello. "are you with me?" – using accelerometers to determine if two devices are carried by the same person. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, pages 33–50, 2004.

[43] Xiaopeng Li, Qiang Zeng, Lannan Luo, and Tongbo Luo. T2pair: Secure and usable pairing for heterogeneous iot devices. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 309–323, 2020.

[44] Yang Li, Rui Tan, and David K. Y. Yau. Natural timestamping using powerline electromagnetic radiation. In *Proceedings of the ACM/IEEE*

International Conference on Information Processing in Sensor Networks (*IPSN*), pages 55–66, 2017.

[45] Qi Lin, Weitao Xu, Guohao Lan, Yesheng Cui, Hong Jia, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. Kehkey: Kinetic energy harvester-based authentication and key generation for body area network. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1), 2020.

[46] Qi Lin, Weitao Xu, Jun Liu, Abdelwahed Khamis, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. H2b: Heartbeat-based secret key generation using piezo vibration sensors. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks* (*IPSN*), pages 265–276, 2019.

[47] Mark Loveless. Understanding bluetooth security. `https://duo.com/decipher/understanding-bluetooth-security`, 2018.

[48] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services* (*MobiSys*), pages 211–224, 2011.

[49] Rene Mayrhofer and Hans Gellersen. Shake well before use: Authentication based on accelerometer data. In *Proceedings of the International Conference on Pervasive Computing* (*Pervasive*), pages 144–161, 2007.

[50] Markus Miettinen, N. Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), pages 880–891, 2014.

[51] Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and N. Asokan. Revisiting context-based authentication in iot. In *Proceedings of the Annual Design Automation Conference* (*DAC*), pages 32:1–32:6, 2018.

[52] A. Olmos. A temperature compensated fully trimmable on-chip ic oscillator. In *Proceedings of the Symposium on Integrated Circuits and Systems Design* (*SBCCI*), pages 181–186, Washington, DC, USA, 2003. IEEE Computer Society.

[53] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *Proceedings of the International Conference on Ubiquitous Computing* (*UbiComp*), pages 271–288, 2007.

[54] L. Perneel, H. Fayyad-Kazan, and M. Timmerman. Can Android be used for real-time purposes? In *Proceedings of the International Conference on Computer Systems and Industrial Informatics* (*CIICS*), pages 1–6, December 2012.

[55] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. Heart-to-heart (h2h): Authentication for implanted medical devices. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), pages 1099–1112, 2013.

[56] Anthony Rowe, Vikram Gupta, and Ragunathan (Raj) Rajkumar. Low-power clock synchronization using electromagnetic energy radiating from ac power lines. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems* (*SenSys*), pages 211–224, 2009.

[57] Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *2006 IEEE Symposium on Security and Privacy* (*S&P*), pages 308–313, 2006.

[58] Dominik Schürmann and Stephan Sigg. Secure communication based on ambient audio. *IEEE Transactions on Mobile Computing*, 12(2):358–370, February 2013.

[59] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N. Asokan. Drone to the rescue: Relay-resilient authentication using ambient multi-sensing. In *Proceedings of Financial Cryptography and Data Security (FC)*, 2014.

[60] Bluetooth SIG. Bluetooth sig statement regarding the exploiting cross-transport key derivation in bluetooth classic and bluetooth low energy (blurtooth) and the security implications of key conversion between br/edr and ble vulnerabilities. `https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/bluetooth-security/blurtooth/`, 2020.

[61] Mike Silva, Norm P. Hummon, D. Rutter, and C. Hooper. Power frequency magnetic fields in the home. *IEEE Transactions on Power Delivery*, 4(1):465–478, January 1989.

[62] Smith. 127 devices added to the internet each second, but congress is clueless about iot, 2015.

[63] StackCommerce. Projections are pointing to 83 billion iot connections by 2024, 2021.

[64] A. M. Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77:793–797, July 1996.

[65] Kaixin Sui, Mengyu Zhou, Dapeng Liu, Minghua Ma, Dan Pei, Youjian Zhao, Zimu Li, and Thomas Moscibroda. Characterizing and improving wifi latency in large-scale operational networks. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 347–360, 2016.

[66] Santi Tofani, Piero Ossola, Giovanni d'Amore, L Anglesio, Motohisa Kanda, and David R Novotny. A three-loop antenna system for performing near-field measurements of electric and magnetic fields from video display terminals. *IEEE Transactions on electromagnetic compatibility*, 38(3):341–347, 1996.

[67] Ersin Uzun, Kristiina Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography and Data Security*, pages 307–324, 2007.

[68] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal de Lara. Amigo: Proximity-based authentication of mobile devices. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, pages 253–270, 2007.

[69] Verizon. 2016 data breach investigations report. Technical report, 2016.

[70] Jack West, Kyuin Lee, Suman Banerjee, Younghyun Kim, George K. Thiruvathukal, and Neil Klingensmith. Moonshine: An online randomness distiller for zero-involvement authentication. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 93–105, 2021.

[71] Yuezhong Wu et al. Auto-key: Using autoencoder to speed up gait-based key generation in body area networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1), 2020.

[72] Wei Xi, Chen Qian, Jinsong Han, Kun Zhao, Sheng Zhong, Xiang-Yang Li, and Jizhong Zhao. Instant and robust authentication and key agreement among mobile devices. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 616–627, New York, NY, USA, 2016. Association for Computing Machinery.

[73] Weitao Xu et al. Lora-key: Secure key generation system for lora-based network. *IEEE Internet of Things Journal*, 6(4), 2019.

[74] Weitao Xu, Guohao Lan, Qi Lin, Sara Khalifa, Neil Bergmann, Mahbub Hassan, and Wen Hu. Keh-gait: Towards a mobile healthcare user authentication system by kinetic energy harvesting. In *Network and Distributed Systems Security Symposium (NDSS)*, pages 1–15, 2017.

[75] Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks (IPSN)*, pages 3:1–3:12, 2016.

[76] Weitao Xu, Junqing Zhang, Shunqi Huang, Chengwen Luo, and Wei Li. Key generation for internet of things: A contemporary survey. *ACM Computing Surveys*, 54(1), January 2021.

[77] Zhenyu Yan, Qun Song, Rui Tan, Yang Li, and Adams Wai Kin Kong. Towards touch-to-access device authentication using induced body electric potentials. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–16, 2019.

[78] Lin Yang, Wei Wang, and Qian Zhang. Secret from muscle: Enabling secure pairing with electromyography. In *Proceedings of the ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 28–41, 2016.

[79] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. Proximity based iot device authentication. In *2017 IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9, 2017.

[80] Zhaoyang Zhang, Honggang Wang, Athanasios V. Vasilakos, and Hua Fang. Ecg-cryptography and authentication in body area net-

works. *IEEE Transactions on Information Technology in Biomedicine*, 16(6):1070–1078, November 2012.

[81] Manfred Zimmermann and Klaus Dostert. Analysis and modeling of impulsive noise in broad-band powerline communications. *IEEE Transactions on Electromagnetic Compatibility*, 44(1):249–258, February 2002.