VoltKey: Continuous Secret Key Generation Based on Power Line Noise for Zero–Involvement Pairing and Authentication

KYUIN LEE, University of Wisconsin–Madison, USA
NEIL KLINGENSMITH, Loyola University Chicago, USA
SUMAN BANERJEE, University of Wisconsin–Madison, USA
YOUNGHYUN KIM, University of Wisconsin–Madison, USA

The explosive proliferation of Internet-of-Things (IoT) ecosystem fuels the needs for a mechanism for the user to easily and securely interconnect multiple heterogeneous devices with minimal involvement. However, the current paradigm of context-unaware pairing and authentication methods (e.g., using a preset or user-defined password) poses severe challenges in the usability and security aspects due to the limited and siloed user interface that requires substantial effort on establishing or maintaining a secure network. In this paper, we present VoltKey, a method that transparently and continuously generates secret keys for colocated devices, leveraging spatiotemporally unique noise contexts observed in commercial power line infrastructure. We introduce a novel scheme to extract randomness from power line noise and securely convert it into the same key by a pair of devices. The unique noise pattern observed only by trusted devices connected to a local power line prevents malicious devices without physical access from obtaining unauthorized access to the network. VoltKey can be implemented on top of standard USB power supplies as a platform-agnostic bolt-on addition to any IoT devices or wireless access points that are constantly connected to the power outlet. Through extensive experiments under various realistic deployment environments, we demonstrate that VoltKey can successfully establish a secret key among colocated devices with over 90% success rate, while effectively rejecting malicious devices that do not have access to the local power line (but may have access to a spatially nearby line).

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Security and privacy** → *Authentication.*

Additional Key Words and Phrases: device pairing, device authentication, key generation, power line noise

ACM Reference Format:
Kyuin Lee, Neil Klingensmith, Suman Banerjee, and Younghyun Kim. 2019. VoltKey: Continuous Secret Key Generation Based on Power Line Noise for Zero–Involvement Pairing and Authentication . *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 93 (September 2019), 26 pages. https://doi.org/10.1145/3351251

1 INTRODUCTION

For the past several years, enthusiastic and ambitious projections have been made for the rapid growth of the Internet-of-Things (IoT) ecosystem. Intel, for instance, has predicted that by the year 2020, the number of connected IoT devices will grow to around 200 billion worldwide, which is more than 20 devices for every

Authors' addresses: Kyuin Lee, University of Wisconsin–Madison, 1415 Engineering Dr, Madison, Wisconsin, USA, kyuin.lee@wisc.edu; Neil Klingensmith, Loyola University Chicago, 1052 West Loyola Avenue, Chicago, Illinois, USA, neil@cs.luc.edu; Suman Banerjee, University of Wisconsin–Madison, 1210 West Dayton St, Madison, Wisconsin, USA, suman@cs.wisc.edu; Younghyun Kim, University of Wisconsin–Madison, 1415 Engineering Dr, Madison, Wisconsin, USA, younghyun.kim@wisc.edu.

2474-9567/2019/9-ART93 $15.00
https://doi.org/10.1145/3351251

person [7]. However, in domestic and personal sectors, this number seems to be far from reality. In fact, over 70% of currently deployed IoT devices are in business, manufacturing, and healthcare sectors, and domestic and personal IoT devices seem to be concentrated only in the hands of enthusiastic early adopters, but not the general public. One of the main hindrances to the adoption of IoT is the long-standing tension between security and usability. Usability is a key aspect of security mechanism for personal IoT systems that are deployed and maintained by non-professional users, as opposed to business, manufacturing, and healthcare domains where teams of professional staff are hired to deploy and maintain large-scale IoT systems. In particular, one of the paramount concerns that have continued to vex researchers is the question of how to quickly, securely, and effortlessly establish a common security key between a newly introduced device and an existing network and to subsequently manage the established connection securely.

Unfortunately, realizing secure and usable key generation is particularly challenging for personal IoT systems because most low-cost IoT devices delegate the user interface to web- or mobile-based apps rather than using their own on-board interfaces, mainly due to form factor or cost constraints [9]. They usually require the use of a third device, most commonly the user's mobile device, to configure devices and establish a secure network. In addition, the web- or mobile-based user interfaces tend to be siloed by the manufacturers—Nest thermostats, for instance, do not seamlessly plug-and-play with Apple HomeKit. Ultimately, users must be knowledgeable and determined to install, manage, and commission comprehensive set of IoT devices. Given this tension between security and usability, 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 [10]. Despite the federal government's consumer advisory [14], more than 15,000 private smart cameras are still unknowingly being streamed. Unfortunately, changing the default password does not adequately address this concern. Studies have shown that users often choose weak passwords or reuse passwords for different purposes [8]. 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. As the number of IoT devices that each user has to manage increases, combined with their heterogeneous and distributed nature, the conventional security mechanisms will fail to provide both security and usability.

Context-based pairing and authentication is a promising solution to this challenge. It exploits spatiotemporal randomness in the ambient environment (e.g., audio, luminosity, or received signal strength indicator), often called *contextual information* [9]. Devices that use context-based security take advantage of the fact that the common contextual information is shared only by a limited group of closely 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. The keys generated from contextual information can, therefore, be used to establish initial trust (as a pairing key) and to protect subsequent communication (as a cryptographic key). This eliminates the need for human involvement for making, entering, and managing a secret key, which dramatically improves the overall usability of IoT systems. In addition, the time-varying nature of contextual information also allows devices to use a new key for each pairing attempt or periodically update the cryptographic key, which significantly reduces the attack window for adversarial agents.

In this work, we introduce a key generation method named VoltKey, which can be used to realize zero-involvement context-based pairing and authentication, leveraging the plug-in power source of devices to extract a shared secret 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.

There are two key challenges that must be addressed in order to realize practical power line-based zero-involvement pairing and authentication. 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 [24]. Therefore, the key generation method should be capable of producing random bit sequences in the presence of strong predictable signals. Second, 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.

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 paired 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 paired across a circuit breaker, we claim this is an 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 pairing and authentication protocols will be required to augment security if such an attack is expected.

In summary, this paper makes the following contributions:

- We introduce a technique to extract randomness from power line noise measurements and convert it to random bit sequences that enable secure pairing and authentication without user involvement.
- We 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 pair or authenticate with each other.
- We implement a low-cost hardware prototype of VoltKey and evaluate it in a variety of environments: office, home, and lab settings. We demonstrate 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.

The remainder of the paper is organized as follows. Section 2 describes the principles behind electrical noise and the underlying assumptions of VoltKey. Next, Section 3 details VoltKey's hardware prototype, as well as overall key generation protocol including synchronization and key extraction methods. In Section 4, we evaluate the prototype of VoltKey under realistic deployment scenarios. Followed by a discussion in Section 5 and a survey of prior related work in Section 6, we conclude this paper in Section 7.

2 BACKGROUND

In this section, we describe the characteristics of the power line noise that VoltKey uses as a source of randomness for key generation. Next, we describe the overall system and threat model for VoltKey.

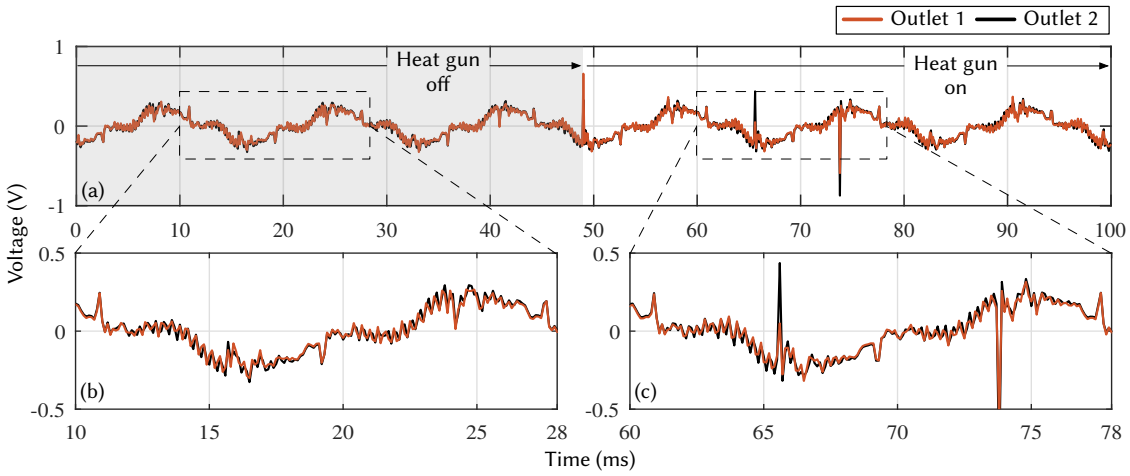

Fig. 1. (a) Measurement of voltage signal on two colocated outlets using a USB DAQ at a sampling rate of 10 kSPS. Single period of 60 Hz signal (b) when the heat gun is off and (c) when it is on.

2.1 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 [5, 24]. 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 [33].

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 [17]. This noise present 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 [6]. 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 [17]. 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 [33].

The combination of 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, we measure the voltage signals on two outlets (less than 20 cm apart) and power cycle a

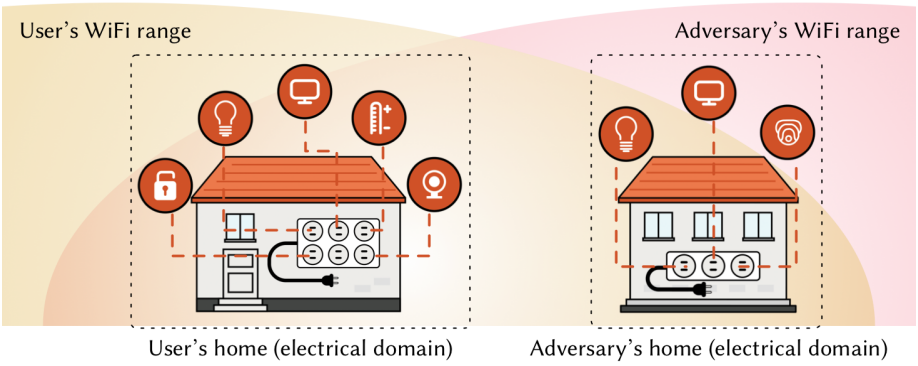

Fig. 2. System and threat models of VoltKey. A number of IoT devices are installed in each home. WiFi range of each home can reach neighboring homes, potentially the adversary's.

heat gun from an electrical outlet located 1.0 m away from the measuring point. Fig. 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). We 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 Fig. 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. 1(b) and 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, which is detailed in Section 3. 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.

2.2 System and Threat Models

We assume a scenario where a number of IoT devices are colocated in their owner's home, as shown in Fig. 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. We do not assume any strong timing properties—devices are not synchronized, and sampling rate and phase may vary among devices. 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, we assume 120 V and 60 Hz throughout this paper.

Our 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 can be a benign device that is accidentally trying to pair or authenticate with devices in the user's home within its WiFi range, or it can be a malicious device that is intentionally trying to do the same. The adversarial

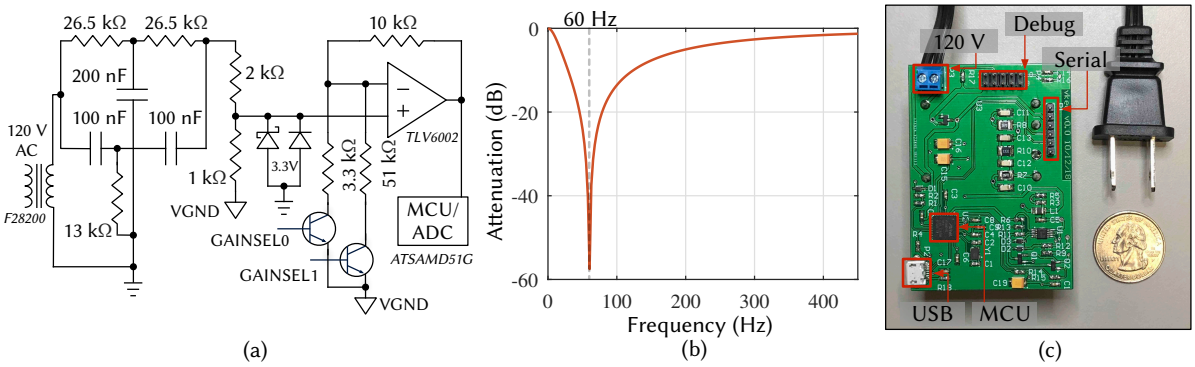

Fig. 3. (a) VoltKey's Analog front-end schematic. MCU's power regulation, debugger and serial communication circuitry is omitted for simplification purposes. (b) Frequency response of the twin-T notch filter used in our prototype. (c) Top-view of VoltKey prototype.

device can intercept unencrypted packets within its WiFi range and listen to public discussion. Also, we assume the adversary has physical access only to an adjacent electrical domain (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, we assume that 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.

3 VOLTKEY DESIGN

In this section, we describe the comprehensive design of VoltKey including its design of hardware prototype and details of overall communication protocol.

3.1 VoltKey Hardware Design

We design VoltKey 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 pairing and 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 Fig. 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. Our prototype uses a split-core transformer with two secondary coils: one to generate power for our circuitry and the host and another to measure noise. We do not want to measure noise on the same transformer tap that we use to generate power because noise from VoltKey 's digital components may corrupt the power line noise measurement. As illustrated in Fig. 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

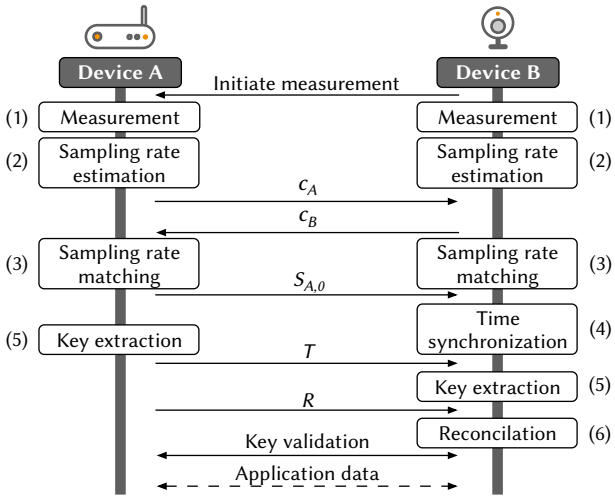

Fig. 4. Overview of VoltKey's key establishment protocol. Solid lines denote plaintext messages exchanged on a public channel and dotted lines represent encrypted messages.

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. We use 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 Fig. 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, we 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, we build 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. Our hardware prototype is equipped with the Microchip's ATSAMD51, a 32-bit ARM-Cortex M4 [12], 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 we chose is considerably more powerful than necessary—VoltKey's application uses very little memory and can run on a low-power processor.

3.2 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 Fig. 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

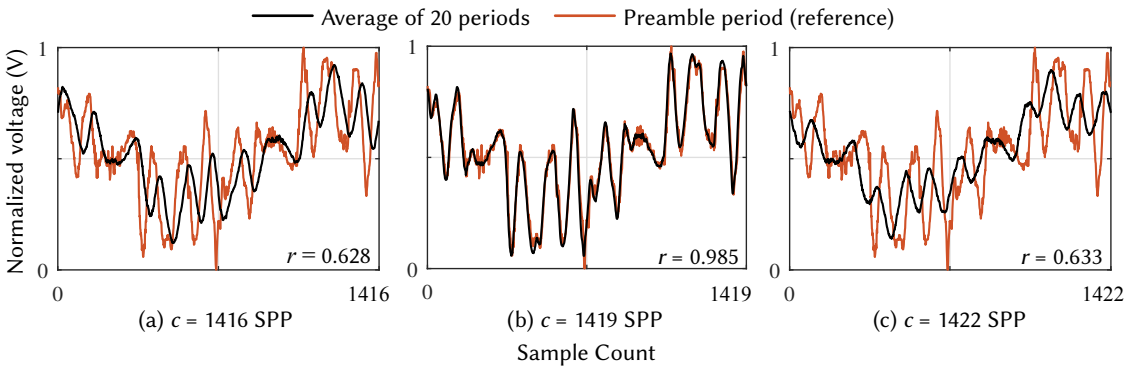

Fig. 5. Mean of uniformly sliced signal at (a) c =1416 SPP, (b) c =1419 SPP, and (c) c =1422 SPP. The correlation coefficient is highest when c is equivalent to the actual SPS divided by 60.

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 Fig. 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. (Section 3.3)
(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. (Section 3.3)
(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. (Section 3.4)
(5) *Bit sequence extraction:* Both devices independently execute bit sequence extraction procedure based on the timestamp T provided by A. (Section 3.5)
(6) *Reconciliation:* Differences in the extracted bit sequences are corrected by B with publicly exchanged data R through key reconciliation stage (Section 3.5).

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, we 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.

3.3 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

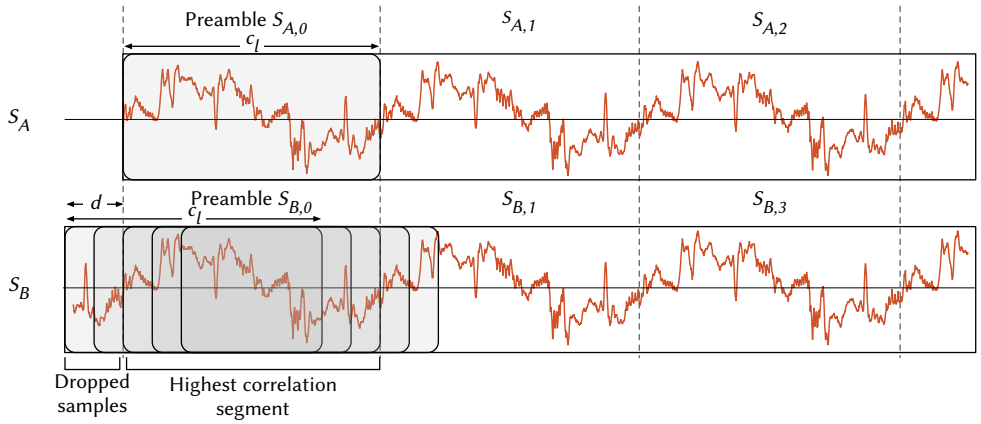

Fig. 6. VoltKey's time synchronization. Using the sliding window approach, Device B locates the most correlated segment between received preamble $S_{A,0}$ and discards the samples up to the offset d.

from timing variability, and moreover, the variability is time-varying. For example, the internal ultra-low-power oscillator frequency of the MCU in our prototype, ATSAMD51, can vary up to ±2% even at a constant room temperature [12]. Extreme temperature variation can cause more severe frequency variability ranging from −17% to +15%. Our 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. Fig. 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.

3.4 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).

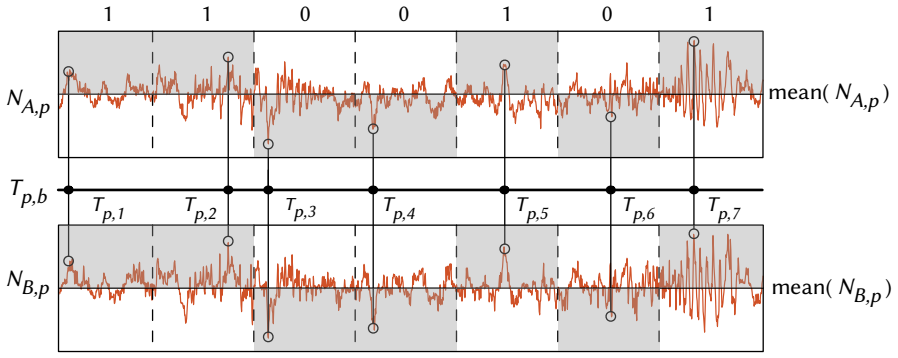

Fig. 7. Bit sequence extraction from the p-th noise period with $n_b = 7$. The largest absolute value of each bin is converted to a bit 1 if indexed value at $T_{p,b}$ is greater than the mean of the noise period, and a bit 0 otherwise.

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 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. Fig. 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 ($p = 0, 1, 2, \ldots, n_p$).

3.5 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 our goal is to remove the 60 Hz periodic portion of the signal before extracting entropy. We define *noise period*, $N_{u,p}$, 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:

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

We 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

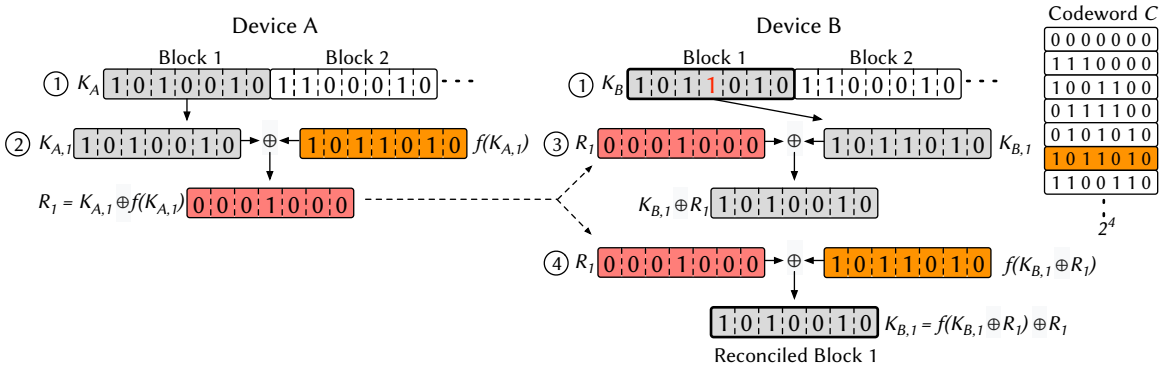

Fig. 8. Illustration of key reconciliation process of the first seven bits using Hamming(7,4) code. ① Bit sequences extracted from both devices are divided into linear blocks of seven bits. ② Difference (exclusive or) between bits in the block and its corresponding codeword, denoted as R_1, is transferred 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.

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:

$$K_{u,p,b} = \begin{cases} 1 & \text{if } N_{u,p}(T_{p,b}) \geq \text{mean}(N_{u,p}) \\ 0 & \text{if } N_{u,p}(T_{p,b}) < \text{mean}(N_{u,p}). \end{cases} \tag{2}$$

Fig. 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 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 [19]. However, we extract 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, our technique reduces bit-wise error in the resulting bit sequences.

3.6 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 our extracted bit sequences for authentication or encryption, we 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. We implement the *quantization-based construction* method presented in [19, 31].

Fig. 8 illustrates the process of quantization-based construction using Hamming(7,4) code. Both devices use a public set of codewords 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 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 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 C, the resulting entropy of each block is worth only 4 bits. Although any other ECC is suitable for quantization-based construction, for simplicity, we 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 [29].

Key reconciliation presents a unique problem when applied to VoltKey and other context-based key generation 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, we must limit the number of bit errors that are allowed to be corrected by key reconciliation. In the reconciliation scheme we presented above, this can be tuned by adjusting n and k, trading off security for reliability.

4 EVALUATION

In this section, we show that VoltKey reliably generates unique keys in a variety of electrical environments. We demonstrate that keys are single-use: they are unique in both place and time, so keys generated at one location at a particular time cannot be reused at a later time or at a different location. We first discuss our experimental setup, evaluation metrics and verify the effectiveness of VoltKey deployed in various environments using different parameters. Additionally, we show that the noise patterns observed by the malicious device with unauthenticated power line measurement do not contain enough information to authenticate a key with trusted devices. Finally, we evaluate VoltKey deployed under realistic scenarios in office and home with a single malicious device installed in the nearby power outlet.

4.1 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. We 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. We simulate bit sequence extraction and information exchange between devices using software written in Matlab. In order to evaluate the *bit agreement rate* in the generated bit sequences between two devices, we extract 128-bit long sequences with n_b=6, 8 and 10 bins, from n_p=22, 16 and 13 periods, respectively. Because pairing and

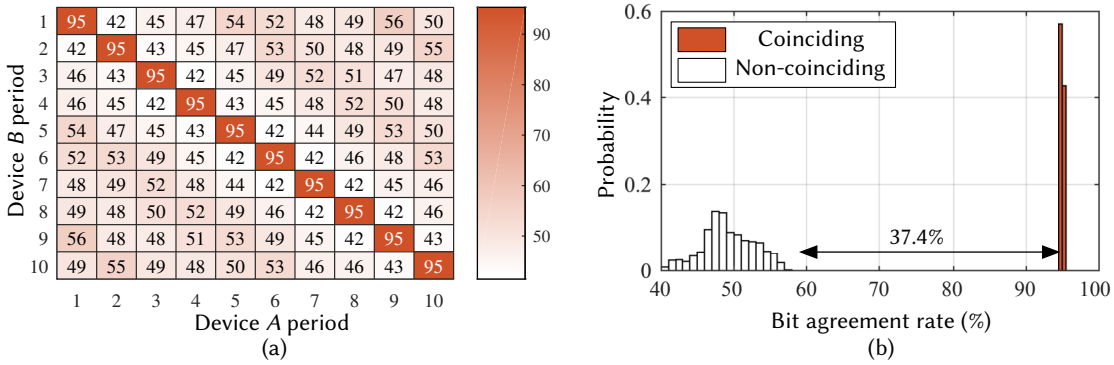

Fig. 9. (a) 10-by-10 confusion matrix of average bit agreement rate between bit sequences generated by noise periods obtained by Device A and B. (b) Distribution of bit agreement rate between diagonal and off-diagonal pairs of noise periods.

authentication by comparing two generated keys are considered successful only when bit-wise errors between two keys are zero after the reconciliation process, we define *pairing success rate* to be a percentage of key pairs with a bit agreement rate of 100% out of all pairs of generated keys.

4.2 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 $N_{u,p}$, we 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%.

Fig. 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). We can clearly see 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. Fig. 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 we 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 proposed in Sections 3.3 and 3.4. It also contributes to the strong security of VoltKey by allowing

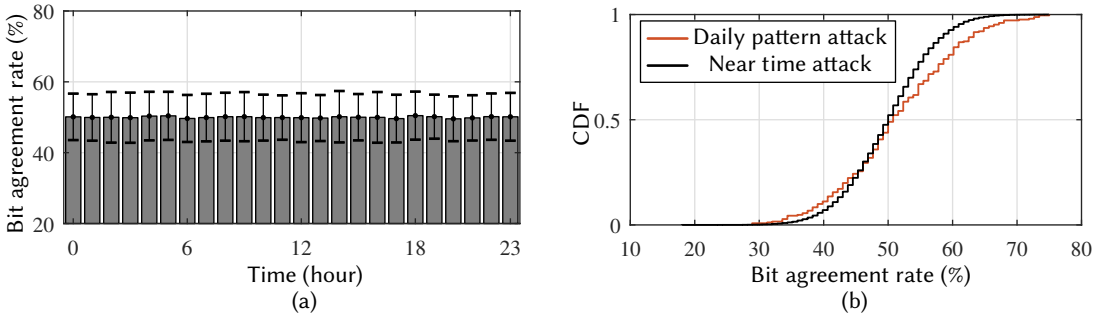

Fig. 10. (a) Bit agreement rate between all keys pairs generated within each hour over course of three consecutive days. (b) CDF of daily pattern and near time attack.

us to use each key only once. Even 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.

4.3 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. We define this attack scenario 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, we 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 Fig. 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 Fig. 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. We refer to this attack scenario as *daily pattern attack*. In order to simulate daily pattern attack, we 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 Fig. 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

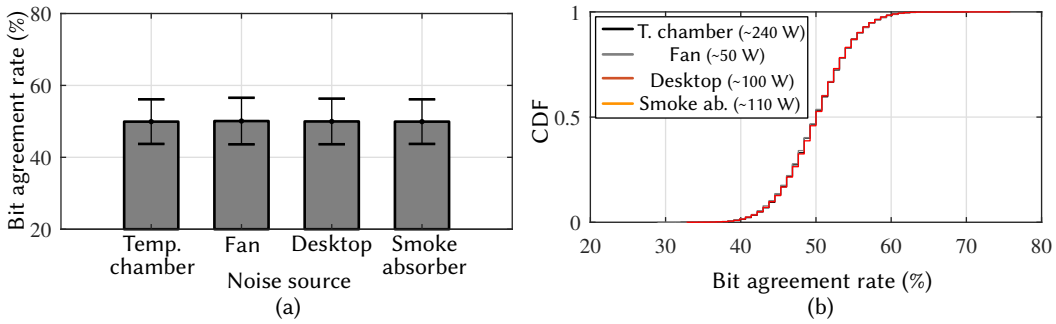

Fig. 11. (a) Bit agreement rate between all keys pairs generated with nearby inductive electrical loads. (b) CDF of dominant noise attack using different loads.

usage pattern, VoltKey harvests different enough bit sequences from the voltage noise to prevent adversaries from carrying out timing-based attack scenarios.

4.4 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 we do not want a malicious agent to 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, we 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. We compare the distribution of the bit agreement rates for each load. As illustrated in Fig. 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, we set another VoltKey device to generate 100 sets of keys with an identical nearby source of dominant noise at different times under different authenticated domain (nearby room at least 10 m apart). We then attempt to authenticate the malicious device to the legitimate network. The CDF of bit agreement rate of 10,000 key pairs are illustrated in Fig. 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).

4.5 Robustness against Temperature Variation

The frequency of the MCU's on-chip oscillator, used to time sample acquisitions from the ADC, is temperature-dependent [23]. Additionally, the ADC's internal voltage reference output also depends on the temperature. Because we are using low-cost MCU with an internal oscillator, the temperature of the device can heavily affect

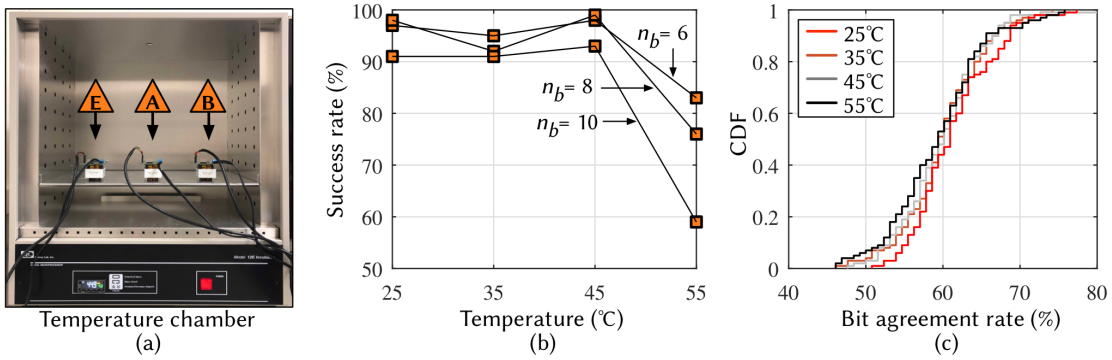

Fig. 12. (a) Experiment setup inside temperature chamber. (b) Success rate between legitimate devices with respect to different operating temperature. (c) CDF of passive attacks with different temperature.

the sampling rate, possibly resulting in the generated keys to be irreconcilable. To understand the effectiveness of VoltKey under different operating temperatures, we 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 Fig. 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 $^{\circ}C$. To ensure all devices to reach the specified temperature, devices are placed inside the temperature chamber for 30 minutes after each temperature adjustment.

Fig. 12(b) illustrates the success rate of legitimate devices with respect to different temperatures. At 25 to 45 $^{\circ}C$, 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 $^{\circ}C$, 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 Fig. 12(c). At 25 $^{\circ}C$, 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.

4.6 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, we set four VoltKey devices to authenticate with a single access point under varying distance within a realistic laboratory environment. As illustrated in Fig. 13(a), five devices (*A*, *B*, *C*, *D* and *E*) are drawing power

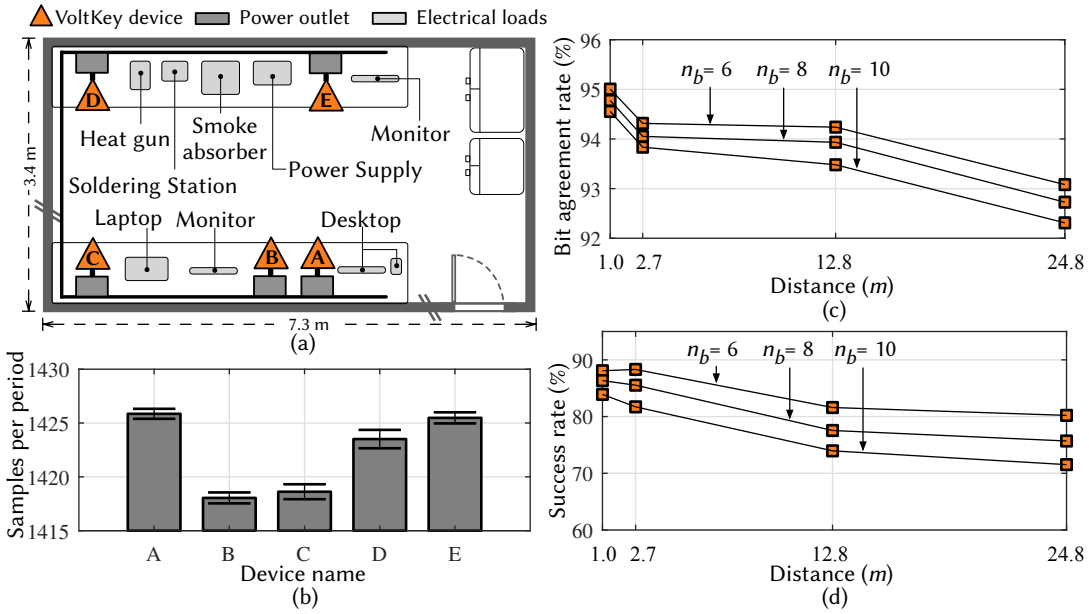

Fig. 13. (a) Location and distance between multiple VoltKey devices (not to scale). The power line is visible around the surrounding wall of the lab. The electrical distance from Device *A* to *B*, *C*, *D* and *E* is 1, 2.7, 12.8 and 24.8 m, respectively. (b) SPS of five different devices. (c) Bit agreement rate between devices with respect to the distance between authenticating devices. (d) Success rate of authentication attempts with respect to distance between authenticating devices.

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, we 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.

Fig. 13(c) illustrates the bit agreement rate against the distance 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 Fig. 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, we find that for each distance, there is a trade-off between the amount of entropy, n_b, and the amount of bit agreement

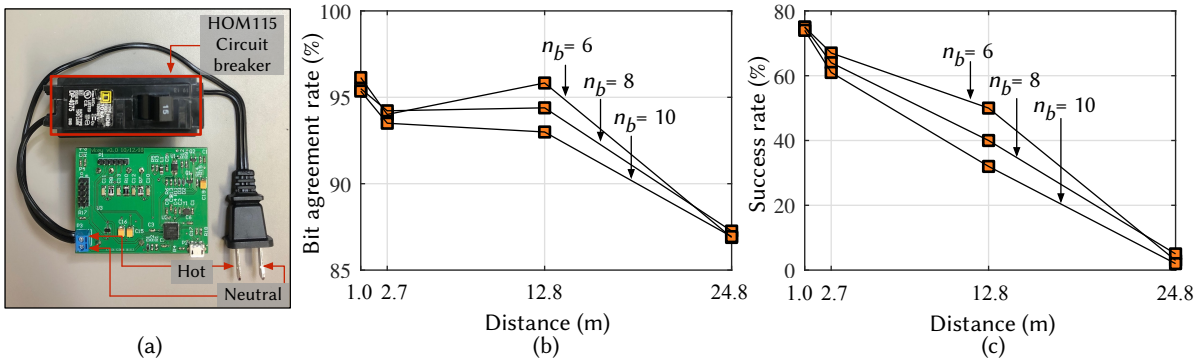

Fig. 14. (a) VoltKey prototype with circuit breaker attached on the hot line of power cable. (b) Bit agreement rate between devices with respect to the distance between authenticating devices. (c) Success rate of authentication attempts with respect to distance between authenticating devices.

we could achieve among nearby VoltKey devices in the authenticated electrical domain. The consequence of this observation is that if we want to extract more bits within a single noise period—which is good for encryption strength—we have 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 our key generation algorithm. Furthermore, to illustrate the effectiveness of VoltKey under varying sampling frequency, Fig. 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. Fig. 14(a) illustrates 120 V–15 A standard D type miniature circuit breaker (Schneider Electric HOM115) installed between hot lines of VoltKey's power cable [13]. For the experiment, Device *B* (with circuit breaker) is relocated to varying distance while Device *A* is fixed to stationary location as shown in Fig. 13(a). For each distance, 100 sets of keys are generated from each device under the duration of 3 hours. As illustrated in Fig. 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 Fig. 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. From this result, we can conclude that the contextual separation between authenticated

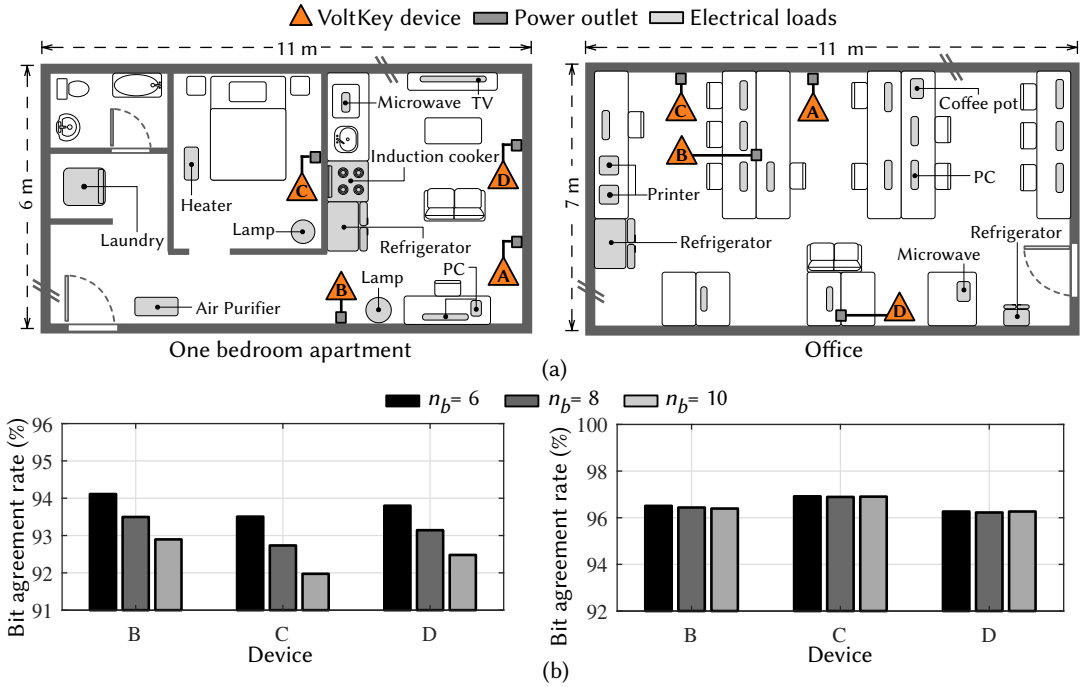

Fig. 15. (a) Floor plans of the one-bedroom apartment and office (not to scale). VoltKey devices are connected to different wall outlets to periodically authenticate themselves with Device *A*. (b) Bit agreement rate of devices with different n_b before key reconciliation.

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

4.7 Realistic Deployment

To verify the overall effectiveness of VoltKey under realistic deployment scenarios, we set four devices within a regular daily environment to measure the success rate and the bit agreement rate of each device. We 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. We conduct two separate experiments in a typical one-bedroom apartment and in an office environment. Fig. 15(a) illustrates our 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[1]. Significant loads that are constantly being power cycled in the course of daily usage are marked in addition to multiple VoltKey devices

[1]The adversary is two rooms over because we didn't have access to the adjacent room.

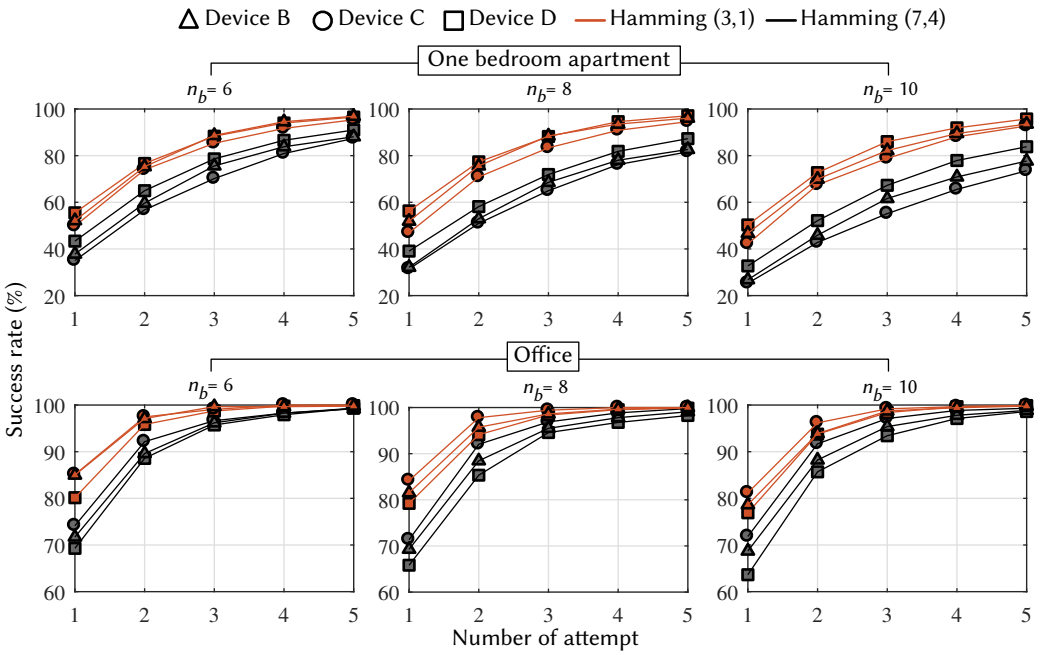

Fig. 16. Successful authentication rate with multiple trials of authentication in apartment and office environment.

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 Fig.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 to inaccurate bit agreement.

Fig.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 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.

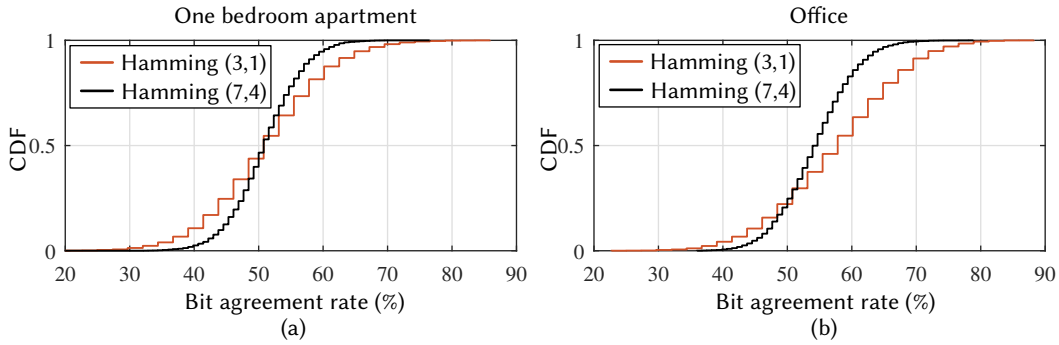

Fig. 17. (a) CDF of bit agreement rate for passive attack (n_b=6) on (a) one-bedroom apartment and (b) office.

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 Fig. 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.

5 DISCUSSION

We have demonstrated that VoltKey is practical in a variety of electrical environments. Key generation in all the environments we studied is reliable enough that heterogeneous IoT devices can use VoltKey to authenticate to an access point with no involvement from the user. We have also shown that it is possible to implement VoltKey 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 this section, we discuss practical challenges and concerns for deploying VoltKey en masse in more detail.

Table 1. NIST Randomness Test Results (p-value ≥ 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

5.1 Duration of Pairing

The duration of the pairing is directly proportional to the n_b and the type of Hamming(n,k) being used for reconciliation stage. Based on our 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 environments, respectively. Consequently, from an adversary's point of view, the resulting entropy of VoltKey is only 0.14 and 0.11 bits. According to [22], 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 we studied, 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 pair 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 pairing does not require a significant amount of time.

5.2 Key Randomness

In order to validate the randomness of the harvested bits, we applied a statistical test suite provided by National Institute of Standards and Technology (NIST) to the 850,000-bit long bitstream generated by VoltKey [1]. The test results are presented in Table.1. Out of 15 total randomness tests, our 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.

5.3 Pairing Radius

In our experiments, we found that the reliability of key reconciliation for devices inside the authenticated electrical domain varied by the environment (i.e. circuit breakers). We imagine that, 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

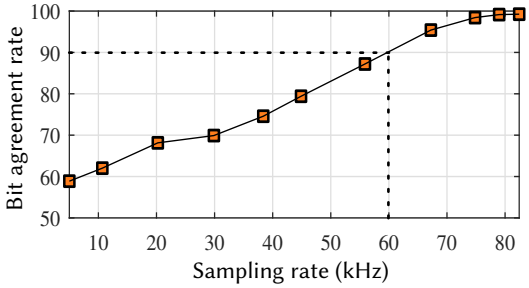

Fig. 18. Bit agreement rate of bit sequences generated by two colocated VoltKey devices with respect to different sampling rate.

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 any context-based pairing or authentication methods that do not exhibit strict context separation boundaries.

5.4 Minimum Sampling Rate

Because we are using 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 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, we downsample the measured signal from 80 kSPS to 5 kSPS to find out minimum sampling rate required to maintain high bit agreement rate. Fig. 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 microcontrollers.

6 RELATED WORK

To mitigate usability challenges, context-based pairing or authentication among mobile and IoT devices has actively been studied, leveraging different context information to establish a secure communication channel or keys using various on-board sensors. For body area networks of wearable devices, context information such as ECG (heartbeat data), EMG (produced by skeletal muscles) and skin vibration has been used to generate keys between low-cost wearable devices and implantable medical devices [2, 18, 25, 31, 32]. By harvesting random keys extracted from the variations in heartbeats measured by ECG, [18] is able to authenticate medical devices only when they are in direct physical contact with the human body using low-cost piezo sensors. In the mobile domain, several prior works have leveraged accelerometer readings to authenticate between user's trusted mobile devices [3, 4, 20]. The authors demonstrate that simultaneous shaking motion of two devices generates unique accelerometer readings that cannot easily be mimicked by an adversary at a close distance. However,

the shaking process significantly degrades user experience and poses impractical challenges when applied to the stationary devices in the IoT domain. This impracticality led other prior works to focus on using readily available contexts of stationary devices such as audio, humidity, luminosity, and visual channels [21, 22, 26–28]. Schürmann and Sigg [27] 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 zero-interaction pairing 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 [21].

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 [11, 15, 16, 30]. Moreover, ProxiMate utilizes any radio technology to extract a secret key based on small scale temporal variations in the perceived wireless signal [19]. However, limitations of these works include small pairing 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 [19]. In contrast, VoltKey is a first novel approach to leverage superimposed noise on the power line to generate keys with relatively low-cost hardware that can be attachable to any existing IoT device with a USB interface to solve usability and practical challenges of previous pairing or authentication schemes.

7 CONCLUSION

We 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. We devised techniques to address practical challenges in implementing VoltKey on low-cost IoT devices, and a hardware prototype was implemented to evaluate them. In our 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 pair with and authenticate each other securely and seamlessly.

ACKNOWLEDGMENTS

This work was supported by the Wisconsin Alumni Research Foundation and NSF under grants CNS-1719336 and CNS-1845469.

REFERENCES

[1] 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. 2010. *SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.* Technical Report. Gaithersburg, MD, United States.

[2] Taha Belkhouja, Xiaojiang Du, Amr Mohamed, Abdulla K. Al-Ali, and Mohsen Guizani. 2019. Biometric-based Authentication Scheme for Implantable Medical Devices during Emergency Situations. *Future Generation Computer Systems* 98 (September 2019), 109–119. https://doi.org/10.1016/j.future.2019.02.002

[3] Daniel Bichler, Guido Stromberg, and Mario Huemer. 2007. Innovative Key Generation Approach to Encrypt Wireless Communication in Personal Area Networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '07)*. 177–181. https://doi.org/10.1109/GLOCOM.2007.41

[4] Daniel Bichler, Guido Stromberg, Mario Huemer, and Manuel Löw. 2007. Key Generation Based on Acceleration Data of Shaking Processes. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp '07)*. 304–317. http://dl.acm.org/citation.cfm?id=1771592.1771610

[5] Morgan H. L. Chan and Robert W. Donaldson. 1989. Amplitude, Width, and Interarrival Distributions for Noise Impulses on Intrabuilding Power Line Communication Networks. *IEEE Transactions on Electromagnetic Compatibility* 31, 3 (August 1989), 320–323. https://doi.org/10.1109/15.30920

[6] Gabe Cohn, Erich Stuntebeck, Jagdish Pandey, Brian Otis, Gregory D. Abowd, and Shwetak N. Patel. 2010. SNUPI: Sensor Nodes Utilizing Powerline Infrastructure. In *Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp '10)*. 159–168. https://doi.org/10.1145/1864349.1864377

[7] Intel Corp. 2019. A Guide to the Internet of Things. https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html. (2019).

[8] Dinei Florencio and Cormac Herley. 2007. A Large-scale Study of Web Password Habits. In *Proceedings of the International Conference on World Wide Web (WWW '07)*. 657–666. https://doi.org/10.1145/1242572.1242661

[9] Mikhail Fomichev, Flor Álvarez, Daniel Steinmetzer, Paul Gardner-Stephen, and Matthias Hollick. 2018. Survey and Systematization of Secure Device Pairing. *IEEE Communications Surveys Tutorials* 20, 1 (2018), 517–550. https://doi.org/10.1109/COMST.2017.2748278

[10] Gizmodo. 2014. A Creepy Website is Streaming from 73,000 Private Security Cameras. (2014). https://gizmodo.com/a-creepy-website-is-streaming-from-73-000-private-secur-1655653510

[11] John E. Hershey, Amer A. Hassan, and Rao Yarlagadda. 1995. Unconventional Cryptographic Keying Variable Management. *IEEE Transactions on Communications* 43, 1 (January 1995), 3–6. https://doi.org/10.1109/26.385951

[12] Microchip Technology Inc. 2018. SAM D5x/E5x Family Data Sheet. (2018). http://ww1.microchip.com/downloads/en/DeviceDoc/60001507C.pdf

[13] Schneider Electric Inc. 2019. Product data sheet HOM115. (2019). https://www.schneider-electric.us/en/product/download-pdf/HOM115

[14] Federal Trade Commission Consumer Information. 2013. Using IP Cameras Safely. (2013). https://www.consumer.ftc.gov/articles/0382-using-ip-cameras-safely

[15] Suman Jana, Sriram Nandha Premnath, Mike Clark, Sneha K. Kasera, Neal Patwari, and Srikanth V. Krishnamurthy. 2009. 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 '09)*. 321–332. https://doi.org/10.1145/1614320.1614356

[16] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. 2010. Ensemble: Cooperative Proximity-based Authentication. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. 331–344. https://doi.org/10.1145/1814433.1814466

[17] Yang Li, Rui Tan, and David K. Y. Yau. 2017. Natural Timestamping Using Powerline Electromagnetic Radiation. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '17)*. 55–66. https://doi.org/10.1145/3055031.3055075

[18] Qi Lin, Weitao Xu, Jun Liu, Abdelwahed Khamis, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. 2019. H2B: Heartbeat-based Secret Key Generation Using Piezo Vibration Sensors. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN '19)*. 265–276. https://doi.org/10.1145/3302506.3310406

[19] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. 2011. ProxiMate: Proximity-based Secure Pairing Using Ambient Wireless Signals. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*. 211–224. https://doi.org/10.1145/1999995.2000016

[20] Rene Mayrhofer and Hans Gellersen. 2007. Shake Well Before Use: Authentication Based on Accelerometer Data. In *Proceedings of the International Conference on Pervasive Computing (Pervasive '07)*. 144–161. http://dl.acm.org/citation.cfm?id=1758156.1758168

[21] Markus Miettinen, N. Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. 2014. 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 '14)*. 880–891. https://doi.org/10.1145/2660267.2660334

[22] Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and N. Asokan. 2018. Revisiting Context-based Authentication in IoT. In *Proceedings of the Annual Design Automation Conference (DAC '18)*. Article 32, 6 pages. https://doi.org/10.1145/3195970.3196106

[23] A. Olmos. 2003. A Temperature Compensated Fully Trimmable On-Chip IC Oscillator. In *Proceedings of the Symposium on Integrated Circuits and Systems Design (SBCCI '03)*. IEEE Computer Society, Washington, DC, USA, 181–186. http://dl.acm.org/citation.cfm?id=942808.943948

[24] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. 2007. 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 '07)*. 271–288. http://dl.acm.org/citation.cfm?id=1771592.1771608

[25] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. 2013. Heart-to-heart (H2H): Authentication for Implanted Medical Devices. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*. 1099–1112. https://doi.org/10.1145/2508859.2516658

[26] Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiainen, and N. Asokan. 2006. Secure Device Pairing Based on a Visual Channel. In *2006 IEEE Symposium on Security and Privacy (S&P '06)*. 308–313. https://doi.org/10.1109/SP.2006.35

[27] Dominik Schürmann and Stephan Sigg. 2013. Secure Communication Based on Ambient Audio. *IEEE Transactions on Mobile Computing* 12, 2 (February 2013), 358–370. https://doi.org/10.1109/TMC.2011.271

[28] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N. Asokan. 2014. Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-sensing. In *Proceedings of Financial Cryptography and Data Security (FC '14)*. 349–364.

[29] A. M. Steane. 1996. Error Correcting Codes in Quantum Theory. *Physical Review Letters* 77 (July 1996), 793–797. Issue 5. https://doi.org/10.1103/PhysRevLett.77.793

[30] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal de Lara. 2007. Amigo: Proximity-based Authentication of Mobile Devices. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp '07)*. 253–270. https://doi.org/10.1007/978-3-540-74853-3_15

[31] Lin Yang, Wei Wang, and Qian Zhang. 2016. Secret from Muscle: Enabling Secure Pairing with Electromyography. In *Proceedings of the ACM Conference on Embedded Network Sensor Systems (SenSys '16)*. 28–41. https://doi.org/10.1145/2994551.2994556

[32] Zhaoyang Zhang, Honggang Wang, Athanasios V. Vasilakos, and Hua Fang. 2012. ECG-Cryptography and Authentication in Body Area Networks. *IEEE Transactions on Information Technology in Biomedicine* 16, 6 (November 2012), 1070–1078. https://doi.org/10.1109/TITB.2012.2206115

[33] Manfred Zimmermann and Klaus Dostert. 2002. Analysis and Modeling of Impulsive Noise in Broad-band Powerline Communications. *IEEE Transactions on Electromagnetic Compatibility* 44, 1 (February 2002), 249–258. https://doi.org/10.1109/15.990732