A Joint Data Compression and Encryption Approach for Wireless Energy Auditing Networks

RUI TAN, Nanyang Technological University
SHENG-YUAN CHIU, National Tsing Hua University
HOANG HAI NGUYEN, University of Illinois at Urbana-Champaign
DAVID K. Y. YAU, Singapore University of Technology and Design
DEOKWOO JUNG, SK Telecom

Fine-grained real-time metering is a fundamental service of wireless energy auditing networks, where metering data is transmitted from embedded wireless power meters to gateways for centralized processing, storage, and forwarding. Due to limited meter capability and wireless bandwidth, the increasing sampling rates and network scales needed to support new energy auditing applications pose significant challenges to metering data *fidelity* and *secrecy*. This paper exploits the *compression* and *encryption* properties of compressive sensing (CS) to design a joint data compression and encryption (JICE) approach that addresses these two challenges simultaneously. Compared with a conventional signal processing pipeline that compresses and encrypts data sequentially, JICE reduces computation and space complexities due to its simple design. It thus leaves more processor time and available buffer space for handling lossy wireless transmissions. Moreover, JICE features an adaptive reconfiguration mechanism that selects the signal representation basis of CS at run time among several candidate bases to achieve the best fidelity of the recovered data at the gateways. This mechanism enables JICE to adapt to changing power consumption patterns. On a smart plug platform, we implemented JICE and several baseline approaches including downsampling, lossless compression, and the pipeline approach. Extensive testbed experiments show that JICE achieves higher data delivery ratios and lower recovery distortions under a range of realistic settings. In particular, at a meter sampling rate of 8 Hz, JICE increases the number of meters supported by a gateway by 50%, compared with the commonly used pipeline approach, while keeping a signal distortion rate lower than 5%.

CCS Concepts: •**Networks** → **Sensor networks;** •**Security and privacy** → *Software and application security;*

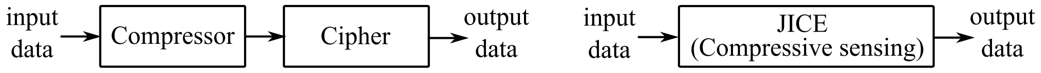
Additional Key Words and Phrases: Compressive sensing, signal processing, energy auditing

1. INTRODUCTION

In emerging smart grids, *wireless energy auditing network* (WEAN) [Jiang et al. 2009b; Jiang et al. 2009a; Dawson-Haggerty et al. 2012; Phillips et al. 2013], which consists of a network of wireless power meters, is a fundamental system to enable various new features such as demand response, usage disaggregation, power quality monitoring, and efficiency diagnosis. These wireless power meters, embedded in smart appliances, switches, and plugs, are designed to support continuous measurements of various physical quantities (e.g., voltage and current) at high frequencies [Jiang et al. 2009b]. However, due to their limited computation and storage capabilities as well as the need to simultaneously support multiple real-time energy auditing applications

Part of this work was published in IEEE SECON 2015.
Authors' addresses: R. Tan, N4-02C-85, 50 Nanyang Avenue, Singapore 639798; S.-Y. Chiu, No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan 30013, R.O.C.; H. H. Nguyen, Coordinated Science Laboratory, 1308 West Main St. Urbana, IL 61801; D. K. Y. Yau, 8 Somapah Road, Building 1, Level 5, Singapore 487372; D. Jung, SK T-Tower, Euljiro 65, Jung-gu, Seoul 100-999, Korea.

DOI: http://dx.doi.org/10.1145/0000000.0000000



(a) Pipeline approach

(b) JICE

Fig. 1. Pipeline approach and JICE.

and enable post-event investigations, it is desirable to transmit the raw metering data to central nodes such as backend servers for advanced processing and long-term storage. This data-collection-over-the-air architecture creates two fundamental requirements. First, *data fidelity* at the central nodes must be preserved to assure the quality of the aforementioned new functionality in smart grids. Second, *data secrecy* during the wireless transmissions must be ensured to protect customers' privacy from malicious eavesdroppers, since the power measurements can easily reveal customers' sensitive information such as their daily routines. Moreover, recent studies show that an adversary can infer much fine-grained information such as the TV channel being watched [Enev et al. 2011] and the web page being browsed [Clark et al. 2013] based on unencrypted power readings. Indeed, public concern for privacy has derailed planned mandatory deployments of smart meters [BBC 2013]. Thus, for WEANs to gain acceptance by consumers and utility, efficient technologies must be devised to address the two requirements.

Higher sampling rates for power signals capture more operating characteristics of appliances, which is desirable for energy auditing. For instance, to capture routine power activities in harmonics analysis, power meters with wireline communication often sample at 1.6 kHz or higher [Eaton 2006]. However, the highest sampling rate adopted by current WEAN prototypes is just 4 Hz [Jiang et al. 2009b]. Moreover, a recent study [Dawson-Haggerty et al. 2012] finds that a WEAN using one smart plug per occupant in a typical commercial office, resulting in 455 plugs deployed totally, can cover only 10% of the appliances. Thus, both the sampling rate and network size need to be increased to improve the spatiotemporal granularity of energy auditing beyond the state of the art. However, the goals impose significant challenges for the resource-constrained smart power meters to meet data fidelity and secrecy requirements. First, a large volume of measurements from individual meters will quickly congest wireless channels. The resulting poor end-to-end data delivery ratio will undermine data fidelity. The problem can be worsened if the wireless channels used by the meters are subject to interference with other wireless communications such as WiFi commonly found in buildings. Second, encryption must be used to achieve data secrecy, but it often incurs severe computational overhead at the meters. Thus, some off-the-shelf smart meters do not encrypt their measurements at all [Rouf et al. 2012]. Therefore, to support higher sampling rates and larger network sizes, new and efficient approaches that can reduce transmission volume and ensure secrecy are highly desirable. Applications in other domains often have such dual requirements of data fidelity and secrecy as well. A conventional approach of pipelining a compressor and a cipher, as illustrated in Fig. 1(a), has been widely employed in other domains, such as multimedia [Ou et al. 2006], storage systems [Grawinkel et al. 2015], and virtual private networks [McGregor and Lee 2000]. However, the long computation delays of the pipeline on the resource-constrained power meters may jeopardize the data fidelity, when some of the measurements have to be dropped to maintain real-time performance.

This paper proposes a joint compression and encryption (JICE) approach based on compressive sensing (CS) [Candès and Wakin 2008] to achieve data fidelity and secrecy simultaneously, as illustrated in Fig. 1(b). CS theory shows that a compressible signal can be represented by a small number of linear projections of the original signal and recovered with high probability. The much reduced transmission volume due to compression helps in turn to reduce channel contention and hence improve data fidelity. CS requires only a simple multiplication between a random matrix and a vector of the original signal, which can be efficiently implemented on capability-limited meters. Moreover, recent studies [Takhar et al. 2006; Rachlin and Baron 2008] show that the cryptanalysis of recovering the original signal from a CS-compressed signal is computationally hard without knowing the true random matrix. Thus, CS provides a form of encryption if the random matrix is established as a shared secret between the transmitter and the receiver. Because of these advantages, CS is a promising approach to achieving simultaneously the dual functions of data compression and encryption on capability-limited wireless meters. In particular, compared with the conventional pipeline approach, the CS-based JICE leaves more processor time and available buffer space for handling lossy wireless transmissions.

This paper presents the design of JICE in WEANs and quantifies the advantages of JICE over other plausible solutions through extensive benchmarking and testbed experiments. The technical contributions of this paper include:

- As CS design is application-specific, we address all the key design elements systematically for WEANs, which include signal sparsity, representation basis, and measurement matrix, based on real power data traces. The results show that, in contrast to the choice of the measurement matrix, the choice of the representation basis can significantly affect the recovery performance. This finding motivates the following adaptive reconfiguration approach for representation basis.
- We propose a novel adaptive reconfiguration approach that identifies the best representation basis autonomously and reconfigures a JICE system to maintain the best fidelity of the recovered data at a gateway in the presence of changing power consumption patterns. Identifying the best representation basis requires an exhaustive evaluation of all candidate bases using recent power data traces. Our approach offloads most computation to the resourceful gateway by leveraging a few key power features that indicate the best representation basis.
- Although an adversary cannot easily recover the measurement data without the random matrix used in the CS, we identify a vulnerability of CS that can leak important statistics information about the measurement data. To solve the vulnerability, we propose a lightweight perturbation approach that adds a random noise signal to the original power signal. We also study a variation of the *known-plaintext attack* against JICE, where the adversary can measure a noisy version of the original signal and estimate the random matrix used by JICE based on accumulated data. We analyze how frequently the random matrix needs to be updated, such that the adversary cannot accumulate enough data to estimate the random matrix.
- On a smart plug platform, we implement JICE and several baseline approaches. Benchmarking and testbed experiments show that JICE reduces the computational overhead and achieves higher data delivery ratios and lower recovery distortions. When the sampling rate is 8 Hz, the number of meters supported by a gateway is increased by 50%, compared with a pipeline approach that applies a compressor and a cipher sequentially, while keeping a data distortion rate lower than 5%.

In summary, as a major contribution of this paper, JICE pushes the Pareto-optimal frontier of the sensor-sampling-rate versus network-size trade-offs under the data fidelity and secrecy requirements, beyond the state of the art including commonly used compression-encryption signal processing pipelines.

The rest of this paper is organized as follows. Section 2 reviews the CS theory and related work. Section 3 states the objective and provides an overview of JICE. Sections 4 and 5 design JICE and discuss its secrecy properties, respectively. Section 6 presents implementation details and benchmarking results. Section 7 presents testbed experiment results. Section 8 extends JICE for voltage monitoring and discusses other application scenarios. Section 9 concludes.

2. PRELIMINARIES AND RELATED WORK

In this section, we first present the preliminaries of CS. Then, we review related work on WEAN and CS.

2.1. Preliminaries

This section briefly reviews CS theory [Candès and Wakin 2008]. Let N denote the length of the input signal. Suppose Ψ is an orthonormal *representation basis* $\Psi = [\psi_1 \psi_2 \cdots \psi_N] \in \mathbb{R}^{N \times N}$, where ψ_i is the ith column of Ψ. The coefficient sequence of a time-domain signal $\mathbf{x} \in \mathbb{R}^{N \times 1}$ on the basis Ψ is denoted as $\hat{\mathbf{x}}$, i.e., $\mathbf{x} = \Psi\hat{\mathbf{x}}$. In other words, $\hat{\mathbf{x}}$ is the Ψ-domain transform of $\mathbf{x}$. The signals $\mathbf{x}$ and $\hat{\mathbf{x}}$ are k-sparse if $\hat{\mathbf{x}}$ has k non-zeros. Let $\mathbf{y} \in \mathbb{R}^{M \times 1}$ denote the output signal and $\Phi \in \mathbb{R}^{M \times N}$ denote the *measurement matrix*, where $M < N$. The measurement process of CS is

$$\mathbf{y} = \Phi\mathbf{x}.$$

In this paper, the *compression ratio*, denoted by γ, is defined as

$$\gamma = \frac{N}{M}.$$

Let $\mathbf{x}'$ and $\hat{\mathbf{x}}'$ denote the recovered signal and its Ψ-domain transform, respectively. The recovery process is

$$\begin{aligned} \mathbf{x}' &= \Psi\hat{\mathbf{x}}', \\ \hat{\mathbf{x}}' &= \underset{\hat{\mathbf{z}} \in \mathbb{R}^{N \times 1}}{\arg\min} \|\hat{\mathbf{z}}\|_{\ell_1}, \text{ subject to } \Phi\Psi\hat{\mathbf{z}} = \mathbf{y}, \end{aligned}$$

where $\|\cdot\|_{\ell_1}$ represents the ℓ_1-norm.

It has been shown in [Candès and Wakin 2008; Candès and Romberg 2007] that, for any k-sparse signal $\mathbf{x}$, the recovery is exact (i.e., $\mathbf{x}' = \mathbf{x}$) if Φ is composed of M rows that are randomly selected from an orthonormal basis $\Phi^* \in \mathbb{R}^{N \times N}$ and

$$M \geq C \cdot \mu^2(\Phi^*, \Psi) \cdot k \cdot \log N, \tag{1}$$

where $\mu(\Phi^*, \Psi)$ is the *coherence* between the two orthonormal matrices Φ^* and Ψ. The coherence is formally given by

$$\mu(\Phi^*, \Psi) = \sqrt{N} \cdot \max_{1 \leq i,j \leq N} |\phi_i^* \cdot \psi_j| \in [1, \sqrt{N}],$$

where ϕ_i^* and ψ_j are the ith row and jth column of Φ^* and Ψ, respectively. In other words, the coherence measures the largest correlation between any row of Φ^* and any column of Ψ. In this paper, we adopt the normalized coherence defined as

$$\bar{\mu}(\Phi^*, \Psi) = \mu(\Phi^*, \Psi)/\sqrt{N} \in [1/\sqrt{N}, 1]. \tag{2}$$

Table I provides a summary of notations used throughout the paper, with their description and the sections where they are defined.

2.2. Related Work

Pervasive sensing is a key element of smart grid technologies. It is estimated that by 2019 more than 100 million wireless sensors will be installed in non-residential buildings, with wireless power meters taking up a major sector [Hatler et al. 2013]. Thus, WEAN has received much research interest in recent years. Early studies have focused on hardware design of the wireless meters [Jiang et al. 2009a], as well as

Table I. Summary of Notation

N	length of input signal to CS	§2.1	M	length of output signal of CS	§2.1
Ψ	representation basis	§2.1	Φ	measurement matrix	§2.1
$\mathbf{x}$	time-domain signal	§2.1	$\hat{\mathbf{x}}$	Ψ-domain transform of $\mathbf{x}$	§2.1
k	number of non-zeros in $\hat{\mathbf{x}}$	§2.1	$\mathbf{y}$	CS-compressed signal, $\mathbf{y} = \Phi\mathbf{x}$	§2.1
$\mathbf{x}'$	recovered time-domain signal	§2.1	$\hat{\mathbf{x}}'$	Ψ-domain transform of $\mathbf{x}'$	§2.1
γ	compression ratio, $\gamma = \frac{N}{M}$	§2.1	$\bar{\mu}$	normalized coherence	§2.1
ϵ	distortion	§4.1	ρ	sparsity of $\mathbf{x}$, $\rho = \frac{k}{N}$	§4.1
$\mathbf{u}$	structured component of $\mathbf{x}$	§4.1	$\mathbf{w}$	white Gaussian noise in $\mathbf{x}$	§4.1
$\mathbf{x}_{(k)}$	time-domain signal of $\hat{\mathbf{x}}_{(k)}$	§4.1	$\hat{\mathbf{x}}_{(k)}$	k largest coefficients of $\hat{\mathbf{x}}$	§4.1
$\rho_{\mathbf{u}}$	sparsity of $\mathbf{u}$	§4.1	Ψ_A	ADT basis	§4.2
Ψ_D	DCT basis	§4.2	Ψ_H	HWT basis	§4.2
Φ_G	Gaussian matrix	§4.3	Φ_R	Rademacher matrix	§4.3
Φ_B	binary matrix	§4.3	S	number of ones in each column of Φ_B	§4.3
$\mathbf{f}$	feature vector, $\mathbf{f} = [r_{\text{ac}}, r_{\text{sc}}, \sigma_x]$	§4.4	r_{ac}	rate of average crossings	§4.4
r_{sc}	rate of sharp changes	§4.4	σ_x	standard deviation of a block	§4.4
$T_1/T_2/T_3$	decision table thresholds	§4.4	$\bar{x}$	mean of a block	§5.2
k_p	shared secret key	§5.2	$\mathbf{p}$	time-domain perturbation vector	§5.2
$\mathbf{s}$	noise in noisy-plaintext attack	§5.3	σ_s^2	variance of $\mathbf{s}$	§5.3

networking issues in small-scale deployments [Jiang et al. 2009b]. A recent research project [Dawson-Haggerty et al. 2012] deployed 455 meters in a commercial building. Although it is recognized [Jiang et al. 2009a] that high sampling rates (multiple to tens of Hz) are important for load disaggregation, all the existing pilot deployments [Jiang et al. 2009b; Jiang et al. 2009a; Dawson-Haggerty et al. 2012] adopt low sampling rates such as one sample per minute [Jiang et al. 2009a]. Data compression to save wireless bandwidth is therefore not critical in these projects. Moreover, they do not address data encryption for privacy in their design.

CS-based data collection protocols that exploit the spatial sparsity of sensor readings have been developed for wireless sensor networks, in the media access control [Bajwa et al. 2006] and network [Luo et al. 2009] layers. These protocols leverage the compression nature of CS to reduce communication cost. In [Bajwa et al. 2006], multiple sensors simultaneously transmit their readings amplified by random factors to a sink node over an analog wireless channel, resulting in a projection of all the readings at the sink. From the multiple projections, the sink can recover the readings. In [Luo et al. 2009], sensors multiply their readings by random vectors and aggregate the vectors in the network, resulting in balanced energy consumption of the sensors. Recent studies have applied CS to various sensing systems including acoustic ranging [Misra et al. 2012], video background subtraction [Shen et al. 2012], and soil moisture monitoring [Wu and Liu 2012]. By exploiting the temporal sparsity of a single sensor's data, they apply CS to schedule the sleep of the sensors [Wu and Liu 2012], and reduce transmission volume [Misra et al. 2012] and computation overhead [Shen et al. 2012]. In contrast, this paper aims to exploit the low computation and space complexity of CS to jointly compress and encrypt temporally sparse power signals on embedded wireless power meters. Moreover, none of the earlier studies aim to reconfigure CS to adapt to changing signal patterns, but we do.

It is observed in [Takhar et al. 2006] that CS implements a form of encryption if the measurement matrix is a secret. As the mutual information between the input and output of CS is non-zero [Rachlin and Baron 2008], like many other ciphers, CS cannot achieve Shannon's *perfect secrecy*. Nonetheless, it is shown that recovering the original signal without the true measurement matrix is computationally difficult [Rachlin and Baron 2008; Orsdemir et al. 2008]. A signal recovered with a wrong measurement matrix is less sparse than the original signal [Rachlin and Baron 2008]. However, the cryptanalysis that exhaustively searches for a measurement matrix to minimize the

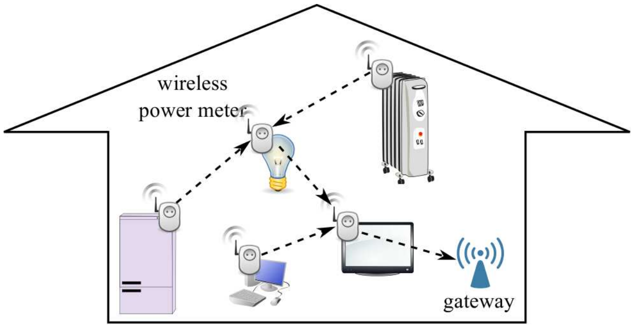


Fig. 2. An example WEAN in a home. A wireless power meter monitors the real-time power consumption of an appliance; the power meters can form a multi-hop network rooted at the gateway; the power meters transmit the metering data to the gateway.

sparsity of a candidate recovery is often intractable. Similarly, it is shown that brute force and structured attacks to estimate a Gaussian measurement matrix is practically infeasible [Orsdemir et al. 2008]. Such secrecy property of CS is used to establish secret communication in asymmetric channels through secure measurement matrices [Agrawal and Vishwanath 2011]. In addition, this secrecy property has also been exploited in [Wang et al. 2013] to develop an outsourcing technique that preserves the secrecy of the original data. Although previous studies [Rachlin and Baron 2008; Orsdemir et al. 2008] established the difficulty of signal recovery without the measurement matrix, this paper identifies a vulnerability of CS to leaking statistics of the original signal and proposes an approach to solving the vulnerability.

3. BACKGROUND, OBJECTIVE, AND OVERVIEW OF APPROACH

In this section, we first introduce the technical background of WEAN and state the objective of this paper. Then, we provide an overview of JICE. Lastly, we present an illustrating example that provides insights into the advantages of JICE.

3.1. Background and Objective Statement

A wireless power meter embedded in smart appliances, switches, and plugs typically consists of a power sensor, a microcontroller unit (MCU), a non-volatile memory, and a Zigbee radio. These meters may draw power from batteries or a power grid. To increase hardware reliability and minimize the impact on the power grid, these meters often use low-power hardware components with limited computation, storage, and communication capabilities. For instance, the TI MSP430F1xx, a widely adopted MCU family for these meters [Lawson 2010; Jiang et al. 2009a], has an 8MHz clock rate and 10KB RAM capacity only. As discussed in Section 1, to enable various advanced energy auditing applications, it is desirable to transmit the metering data to a *gateway* such as a smart meter or a Zigbee access point, while meeting the data fidelity and secrecy requirements. The wireless power meters can form a multi-hop network rooted at the gateway. This network is referred to as a WEAN. Fig. 2 shows an example WEAN in a household environment, where the wireless power meters monitor the real-time power consumption of home appliances.

As power signals are often temporally correlated and hence compressible, data compression can be applied to reduce transmission volume and alleviate the channel contention. Moreover, it allows more retransmission attempts for lost packets, given a transmission deadline that ensures real-time metering. Various ciphers [TinyOS 2010] can be employed to ensure data secrecy. However, system designers often face challenges in pipelining existing compressors and ciphers, as they incur significant computation and storage overhead at the capability-limited meters. Without careful design, the execution time of these algorithms could jeopardize the timeliness of metering easily. In this paper, we attempt to exploit the compression nature of CS and its encryption property studied recently [Takhar et al. 2006; Rachlin and Baron 2008; Orsdemir et al. 2008; Agrawal and Vishwanath 2011; Wang et al. 2013] to design a JICE approach that addresses the requirements of fidelity and secrecy simultaneously. Owing to the simplicity of CS, JICE can keep a simple system design of low computation complexity. As such, we aim to investigate whether and how the CS-based JICE can push the Pareto-optimal frontier of the sensor-sampling-rate versus network-size trade-offs under the data fidelity and secrecy requirements, beyond the commonly used compression-encryption signal processing pipelines.

To this end, in the rest of this paper, we aim to answer the following supporting research questions:

Q1. How to design the key elements of CS including representation basis and measurement matrix for WEANs to reduce transmission volume and preserve data fidelity without introducing significant computation overhead? This question is addressed in Section 4.

Q2. What are the secrecy vulnerabilities (if any) of CS as a form of encryption? And how to fix them? These two related questions are addressed in Section 5.

Q3. Can CS bring substantial benefits such as reduced packet losses and recovery errors, compared with other plausible approaches? This question is addressed in Sections 6 and 7.

3.2. Approach Overview

In answering the research questions *Q1*, *Q2*, and *Q3*, our designs and evaluations in the rest of this paper constitute an integrated JICE approach. Fig. 3 provides an overview of the work flow of JICE, which is described in this section. Note that our discussion on the design of JICE focuses on a pair of a wireless power meter and a gateway, while the performance evaluation of JICE will be based on a WEAN consisting of a network of wireless power meters and a gateway. Thus, the evaluation captures the overall system performance where each meter-gateway link adopts JICE. The wireless power meters can form a multi-hop network rooted at the gateway. Thus, the communications between a wireless power meter and the gateway as illustrated in Fig. 3 can be over multiple hops. To make sense the power consumption data collected by each wireless power meter, we assume that the clocks of the power meters are synchronized with that of the gateway with satisfactory accuracy, such that each power meter can timestamp the power consumption samples. CSMA-based media access control (MAC) protocols can be used to support JICE.

The basic work flow of JICE is as follows. The wireless power meter continuously samples the power consumption. JICE processes the real-time power consumption data block by block, where each block is a power signal x with a finite length. The length of x, i.e., N, is called *block size*. The meter compresses x using a CS algorithm and transmits the CS-compressed version of x (i.e., y in Fig. 3) over air to a gateway, which then recovers the x for electricity analytics. The threat is an eavesdropper that can capture the wireless communications between the power meter and the gateway. JICE makes the eavesdropper's cryptanalysis of recovering the x computationally hard.

Specifically, whenever a new block x is available, the power meter and the gateway perform the following steps that are illustrated in Fig. 3:

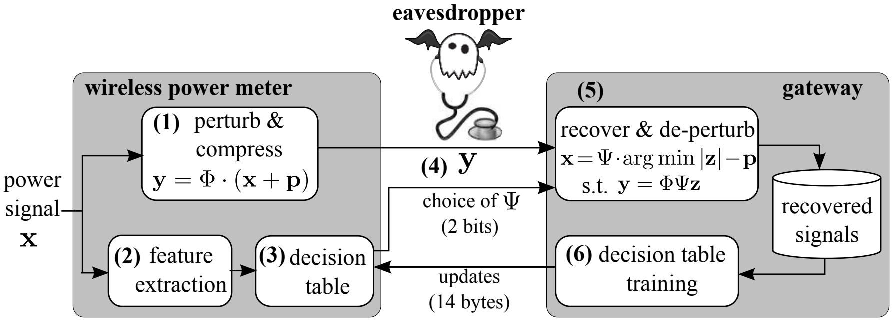

eavesdropper
wireless power meter
power signal $\mathbf{x}$
(1) perturb & compress
$\mathbf{y} = \Phi \cdot (\mathbf{x} + \mathbf{p})$
(2) feature extraction
(3) decision table
(4) $\mathbf{y}$
choice of Ψ (2 bits)
updates (14 bytes)
(5)
gateway
recover & de-perturb
$\mathbf{x} = \Psi \cdot \arg\min |\mathbf{z}| - \mathbf{p}$
s.t. $\mathbf{y} = \Phi\Psi\mathbf{z}$
recovered signals
(6) decision table training

Fig. 3. Overview of JICE work flow. The $\mathbf{y}$, the choice of Ψ, and the decision table can be transmitted by a multi-hop network.

(1) The meter first perturbs the original power signal $\mathbf{x}$ by adding a perturbation vector $\mathbf{p}$ to fix a secrecy vulnerability of CS identified when studying the research question *Q2*. It then compresses the perturbed signal by multiplying it by the random measurement matrix Φ.
(2) The meter also extracts a feature vector consisting of three simple statistics of $\mathbf{x}$.
(3) Based on the feature vector, the meter uses a decision table to choose the most efficient representation basis Ψ among several candidate bases. This design is motivated by a key observation when studying the research question *Q1*, i.e., the representation basis significantly affects recovery performance.
(4) The meter sends the CS-compressed signal $\mathbf{y}$ and the choice of Ψ to the gateway.
(5) Upon receiving $\mathbf{y}$ and choice of Ψ, the gateway recovers $\mathbf{x}$ using $\mathbf{p}$, Φ, and the chosen Ψ. The gateway may run advanced electricity analytics based on the recovered data and/or forward the data to backend servers through secure wireline links.

Moreover, as illustrated in Step (6) in Fig. 3, the gateway runs a decision table training algorithm periodically (e.g., every one hour) based on recently recovered data, and sends the updated decision table to the meter, such that the system can adapt well to changing power characteristics. The decision-table-based representation basis selection at the power meter, i.e., Step (3), and the decision table training at the gateway, i.e., Step (6), forms the *adaptive reconfiguration mechanism*.

Note that the perturbation vector $\mathbf{p}$ and the measurement matrix Φ are shared secrets between the meter and the gateway, to ensure the secrecy of the data transmissions over air. Moreover, they should be different across the meters, such that leak of the secret of a single meter (e.g., by extracting from the memory of a physically captured meter) will not reveal the secrets of other meters. The details for establishing these shared secrets will be discussed in Section 5.

Note that this paper focuses on the confidentiality threat from the eavesdropper shown in Fig. 3. A WEAN may face other threats such as integrity and availability attacks on the data packets exchanged between the power meter and the gateway. These attacks can adversely impact the performance of JICE. By further integrating JICE with cryptographic protection methods (e.g., cryptographic signature) and resilient data delivery approaches, the resilience of JICE can be improved against the integrity and availability threats. Due to channel contention and interference among wireless power meters, the gateway may receive a portion of the CS-compressed data (i.e., $\mathbf{y}$) only. Section 7 will present an approach for the gateway to recover the $\mathbf{x}$ from a portion of $\mathbf{y}$ and evaluate the impact of the data loss on the recovery distortion (cf. Fig. 22).

JICE is a novel application of CS to improve WEAN's data collection performance under the data fidelity and secrecy requirements. As discussed in Section 2.2, none

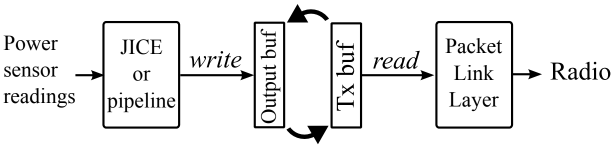


Fig. 4. Illustration of the double buffering between JICE (or the pipeline approach) and the reliable transmission protocol Packet Link Layer (PLL). In JICE, the whole output buffer is updated for each new power sensor reading. When the output buffer is ready after processing N new readings, the output buffer and the transmission buffer (Tx buf) are swapped. Then, the PLL starts sending new data.

of the earlier studies of applying CS in sensor networks reconfigure CS to adapt to changing signal patterns. Existing studies on the secrecy of CS focus on analyzing the difficulty of the cryptanalysis of recovering the original signal. The adaptive reconfiguration mechanism and the perturbation approach to fix specific information leakage issues discovered in this paper are two novel contributions of JICE.

3.3. An Illustrating Example

Before presenting the detailed design of each component of JICE, in this section, we present an illustrating example that shows the intricate interactions among the computation and space complexity of compression/encryption at the wireless power meter, the quality of wireless link, and the fidelity of signal recovery at the gateway. It also provides insights into understanding why JICE with low computation complexity can better address the challenges arising from the interactions.

We compare JICE with a pipeline approach that applies a wavelet-based compressor and the Advanced Encryption Standard (AES) cipher sequentially. We note that the implementations of JICE and the pipeline approach are well optimized. The implementation details can be found in Section 6. Both approaches process the metering data block by block. The block size N is a major factor for determining computation and RAM storage overhead. Thus, its setting must meet the meter's computation and memory constraints. For instance, on an MSP320-based smart plug platform [Sonnonet 2011], if the power sensor sampling rate is $64\,\text{Hz}$, the maximum block size for JICE is 2048 (i.e., each block x contains readings sampled for 32 seconds), since under this setting, JICE will fully utilize the RAM. To preserve data fidelity, we adopt a reliable transmission protocol called Packet Link Layer (PLL) [David M. 2007], which retransmits a packet if its acknowledgment is not received in time.

We adopt the widely used *double buffering* method to interface the output of the JICE/pipeline and the input of the PLL, which is illustrated in Fig. 4. PLL reads data sequentially from a *transmission buffer* for packeting and transmission. In JICE, the whole *output buffer* is updated whenever the meter obtains a new reading (cf. Section 6). Whenever the output buffer is ready for transmission after JICE/pipeline has processed N new readings (i.e., a block), the pointers to the two buffers held by the JICE/pipeline and the PLL are swapped. Then, the PLL starts sending new data. However, because of either channel contention or deteriorated link quality, the transmission buffer may not be fully processed when new data in the output buffer becomes ready. To prevent the new data from being overwritten and ensure real-time metering, the pointers to the two buffers are still swapped immediately and therefore, the meter stops sending from the partially processed transmission buffer and instead starts sending the new data. Intuitively, a large block size allows the meter to try more retransmissions per packet until a successful attempt under harsh channel conditions. In contrast, with small block sizes, the meter is more likely to stop retransmissions and discard old data pending transmission, which results in data loss. Thus, both JICE and the pipeline approach should adopt their maximum achievable block sizes given

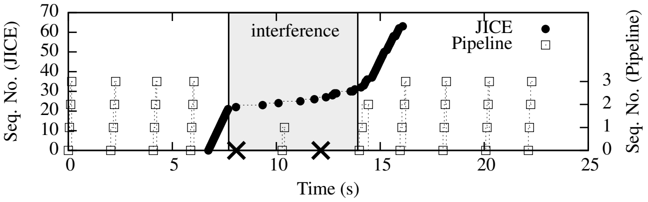


Fig. 5. Packet sequence number (starting from 0) vs. time. The JICE approach successfully transmits all the packets. The pipeline approach drops 4, 2, 4, and 1 packets around the 8th, 10th, 12th, and 14th seconds, respectively. A cross means that all the 4 packets are dropped by the pipeline approach.

certain sampling rate and RAM capacity, which will be benchmarked in Section 6. As the pipeline approach has higher computation and space complexities, its maximum achievable block size subject to real-time processing requirement and RAM capacity constraint is smaller than JICE's. Thus, compared with the baseline approach, JICE may better tolerate link quality deterioration due to its larger block size.

We conduct an experiment to validate the above discussion. We deploy the pipeline approach and JICE on two smart plugs sampling at $64\,\text{Hz}$, respectively. They use the same compression ratio ($\gamma = 4$) and their maximum achievable block sizes are 128 and 2048, respectively (cf. Table VII). As a result, the pipeline approach sends 4 packets every 2 seconds, while JICE sends 64 packets every 32 seconds. Fig. 5 shows the sequence numbers of packets received by the gateway versus time. We intentionally create wireless interference using another smart plug that continuously transmits packets. We can see that during the interference period, the packet delivery rate of JICE drops. Nevertheless, it successfully transmits all the 64 packets within 10 seconds, before the pointers to the two buffers are swapped. In contrast, the pipeline approach fails to transmit a total of 11 packets during the interference period, because it swaps the pointers every 2 seconds. The packet drops will inevitably affect the signal recovery quality. From this experiment, we can see that, although JICE and the pipeline approach achieve the same compression ratio with data secrecy, JICE maintains better data delivery ratio when the link quality deteriorates. We note that this experiment is specifically designed to provide insights into the advantages of JICE due to its low complexity. It also provides a basis for understanding the extensive experiments in Section 7 that compare JICE with the pipeline and other baseline approaches under realistic settings.

Having shown the above illustrating example, in the following two sections, we will present the detailed design of JICE.

4. COMPRESSIVE SENSING FOR POWER SIGNALS

This section analyzes the sparsity of power signals, which is essential to the design of JICE. We then design the representation basis Ψ and the measurement matrix Φ. There is no systematic way of selecting Ψ and Φ; it is usually done through a trial-and-error approach. This paper similarly adopts such an approach, with guidance by CS theory to make the process more scientific. Finally, we develop a novel adaptive reconfiguration approach for the power meters to choose the most efficient representation basis. Note that, in this paper, we focus primarily on the design of JICE for power signals, although to demonstrate generality, we also extend our design to address voltage signals in Section 8.

4.1. Approximate Sparsity

The sparsity of signal x or $\hat{\text{x}}$, denoted by ρ, is defined as $\rho = k/N$. CS theory assumes that $\hat{\text{x}}$ has $N - k$ zeros (i.e., k-sparse). However, in practice, $\hat{\text{x}}$ typically contains small values rather than zeros. A common approach is to approximate $\hat{\mathbf{x}}$ by $\hat{\mathbf{x}}_{(k)}$, which is obtained by keeping the k largest coefficients of $\hat{\mathbf{x}}$ and setting the others to zeros [Bajwa et al. 2006]. As a result, the setting of k (or ρ) affects the approximation accuracy and the design of CS (e.g., the choice of compression ratio). This section analyzes the impact of the setting for ρ on the approximation accuracy, which provides analytical guidelines for setting ρ. We assume that $\mathbf{x}$ is the sum of a *structured signal* (denoted by $\mathbf{u}$) and a zero-mean white Gaussian noise (denoted by $\mathbf{w}$). Formally, $\mathbf{x} = \mathbf{u} + \mathbf{w}$. The structured signal $\mathbf{u}$ captures the power consumption pattern of the monitored appliances. The white Gaussian noise assumption follows from the central limit theorem due to the fact that electrical circuits of appliances generate noises independently. The sparsities of $\mathbf{u}$ and $\mathbf{w}$ jointly determine ρ. The structured signal $\mathbf{u}$ is naturally sparse, while the white noise $\mathbf{w}$ is a limiting factor in determining ρ because it will not be sparsified given any orthonormal representation basis. In this section, we formally analyze the impact of the sparsity of $\mathbf{u}$ and noise strength on the setting of ρ.

We define the following notation. Define $\mathbf{x}_{(k)} = \Psi\hat{\mathbf{x}}_{(k)}$. The *distortion* of $\mathbf{x}_{(k)}$ is then given by

$$\epsilon(\mathbf{x}_{(k)}, \mathbf{x}) = \frac{\|\mathbf{x}_{(k)} - \mathbf{x}\|_{\ell_2}}{\|\mathbf{x}\|_{\ell_2}}. \tag{3}$$

Let $\sigma_{\mathbf{w}}^2$ denote the variance of $\mathbf{w}$, i.e., $\sigma_{\mathbf{w}}^2 = \lim_{N\to\infty} \|\mathbf{w}\|_{\ell_2}^2/N$. We define the signal-to-noise ratio as $\mathrm{SNR} = \lim_{N\to\infty} \|\mathbf{u}\|_{\ell_2}^2/\|\mathbf{w}\|_{\ell_2}^2$. Let $[\mathbf{z}]_i$ denote the ith element of a signal $\mathbf{z}$. We define $\hat{\mathbf{u}}_{(k)}$ by setting $[\hat{\mathbf{u}}_{(k)}]_i = 0$ if $[\hat{\mathbf{x}}_{(k)}]_i = 0$ or $[\hat{\mathbf{u}}_{(k)}]_i = [\hat{\mathbf{u}}]_i$ otherwise. We define $\hat{\mathbf{w}}_{(k)}$ according to $\hat{\mathbf{w}}$ and $\hat{\mathbf{x}}_{(k)}$ in a similar way. Let $\hat{\mathbf{u}}_{\min}$ and $\rho_{\mathbf{u}}$ denote the smallest non-zero element and the sparsity of $\mathbf{u}$, respectively. We have the following proposition, which gives an asymptotic lower bound for the distortion of $\mathbf{x}_{(k)}$. The proof can be found in Appendix A.

PROPOSITION 4.1. *For an orthonormal representation basis* Ψ,

$$\lim_{N\to\infty} \epsilon(\mathbf{x}_{(k)}, \mathbf{x}) \geq \begin{cases} \sqrt{\frac{\frac{\hat{\mathbf{u}}_{\min}^2}{\sigma_{\mathbf{w}}^2}(\rho_{\mathbf{u}} - \rho) + (1-\rho)}{1+\mathrm{SNR}}}, & \textit{if } \rho \in [0, \rho_{\mathbf{u}}]; \\ \sqrt{\frac{1-\rho}{1+\mathrm{SNR}}}, & \textit{otherwise}. \end{cases} \tag{4}$$

Fig. 6 illustrates the lower bound, which shows the trade-off between the distortion ϵ and the sparsity ρ. From Proposition 4.1, these bounds are largely affected by SNR and $\rho_{\mathbf{u}}$ (i.e., the sparsity of $\mathbf{u}$), where the SNR is an inherent property of $\mathbf{x}$ and $\rho_{\mathbf{u}}$ depends on Ψ. From Fig. 6, for a smaller $\rho_{\mathbf{u}}$, ρ have a larger range (i.e., $[\rho_{\mathbf{u}}, 1]$) where ϵ follows the lowest bound (i.e., $\sqrt{\frac{1-\rho}{1+\mathrm{SNR}}}$). Thus, it is desirable to choose Ψ to minimize $\rho_{\mathbf{u}}$ and achieve the lowest trade-off curve between ϵ and ρ. Moreover, from Proposition 4.1, in the presence of noise, there does not exist a setting for ρ within $(0, 1)$ such that the distortion is zero. Thus, we define approximate sparsity as follows.

Definition 4.2. A signal $\mathbf{x}$ is said approximately k-sparse, where k is the minimum of l subject to $\epsilon(\mathbf{x}_{(l)}, \mathbf{x}) \leq \epsilon_0$ and ϵ_0 is a small fixed threshold. The $\rho = \frac{k}{N}$ is the *approximate sparsity* of $\mathbf{x}$.

From Proposition 4.1, for a sufficiently small $\rho_{\mathbf{u}}$, we can solve ρ from the second inequality in Eq. (4) and obtain $\rho \lessapprox 1 - \epsilon_0^2(1 + \mathrm{SNR})$, which suggests the setting for ρ that ensures a distortion of ϵ_0 given an SNR. In the rest of this paper, we refer to *approximate sparsity* when we mention sparsity and ϵ_0 is set to 1% unless otherwise

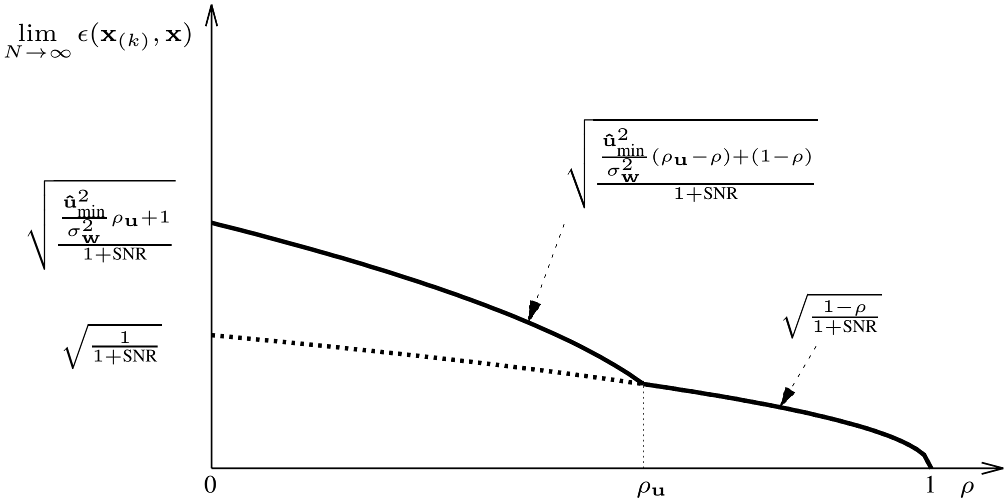


Fig. 6. Asymptotic lower bound of distortion versus setting of sparsity.

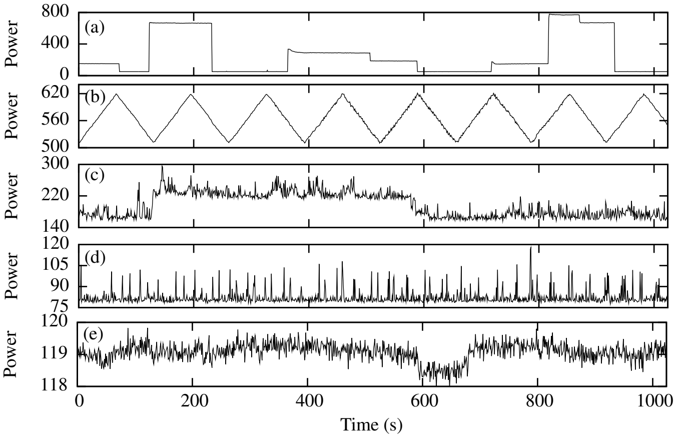


Fig. 7. Various types of power consumption (unit: watt): (a) Duty-cycled (pantry); (b) Periodic (boardroom); (c) Fluctuating (office); (d) Spiky (server room); (e) Silent (boardroom).

specified. In next subsection, we will apply the analytic result in Proposition 4.1 to explain the empirical results on the design of representation basis based on the real power consumption traces.

4.2. Design of Representation Basis Ψ

We drive the design of CS in JICE based on real power traces. We first present the details of a data set of real power traces collected on one floor of an office building for 18 hours. The floor has multiple rooms and open areas, which draw power from a total of 39 branches at a main power panel. We install power meters on the main power panel to measure the power consumption per branch at one Hz. Different appliances have different power consumption patterns. After a thorough inspection, the power signals can be classified into five categories: *duty-cycled*, *periodic*, *fluctuating*, *spiky*, and *silent*. Fig. 7 shows examples for the different types. Many heating/cooling appli-

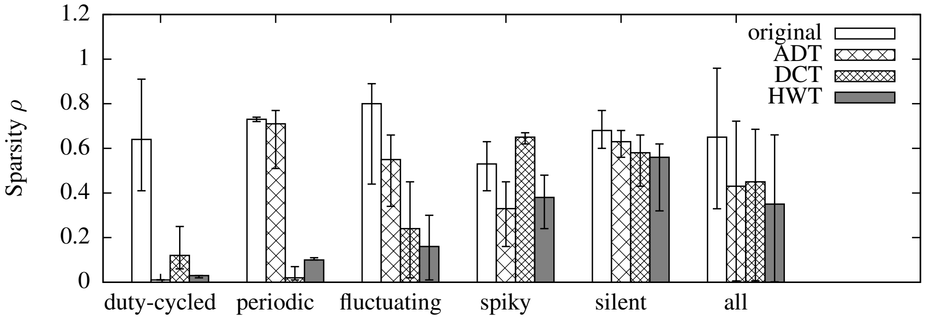


Fig. 8. Sparsity of various power signal categories under different representation bases. ($N = 1024$, the error bars represent min and max values.)

Table II. SNR and $\rho_{\mathbf{u}}$ for various combinations of power signal types and representation bases

	duty-cycled	periodic	fluctuating	spiky	silent
SNR	709.76	31.672	4.926	1.002	0.982
$\rho_{\mathbf{u}}$ (ADT)	0.0059	0.6406	0.0674	0.0186	0.1055
$\rho_{\mathbf{u}}$ (DCT)	0.0098	0.0020	0.0176	0.0020	0.3691
$\rho_{\mathbf{u}}$ (HWT)	0.0078	0.0420	0.0107	0.0029	0.0654

ances duty-cycle to achieve the desired temperatures. Fig. 7(a) shows the duty-cycled power consumption in a pantry with a refrigerator and a water dispenser. Fig. 7(b) shows the periodic power consumption of a projector in a boardroom. Computers can generate complex power consumption patterns. Fig. 7(c) shows the power trace of an office room, where the fluctuations of about 40 watts are caused by a desktop computer. Fig. 7(d) shows the power trace of a server room, with power spikes caused by a bursty workload of the servers. Fig. 7(e) shows the power consumption of the boardroom when only ceiling lights are on, where the fluctuations are within one watt.

From Eq. (1), it is desirable to choose a representation basis to efficiently sparsify the signal to achieve a better compression ratio. This paper considers three commonly adopted representation bases [Wu and Liu 2012; Luo et al. 2009], although JICE can easily encompass other bases. The *adjacent difference transform* (ADT) [Wu and Liu 2012] computes the difference between two adjacent samples and can sparsify a steady signal that occasionally has transient changes. The *discrete cosine transform* (DCT) expresses the signal by a weighted sum of cosine functions of different frequencies, and hence can efficiently sparsify signals with periodic components. The discrete wavelet transform can compactly represent the signal with both temporal correlation and periodicity. In this paper, we consider the *Haar wavelet transform* (HWT), which is commonly used with CS [Wu and Liu 2012]. Let Ψ_A, Ψ_D, and Ψ_H denote the representation bases for ADT, DCT, and HWT, respectively. Their definitions can be found in [Wu and Liu 2012; MathWorks 2015; MathWorks 2010].

We separately evaluate the sparsity of power signals for the different categories under different representation bases. The results are shown in Fig. 8. We can see that all the three transforms can reduce the signal sparsity with respect to the original signal. To explain the results, we estimate SNR and $\rho_{\mathbf{u}}$, which are the major factors of sparsity from Proposition 4.1. We also analyze the sparsity-distortion trade-off for the power signals under different representation bases. Table II summarizes these two parameters for various combinations of power signal types and representation bases. To estimate the two parameters, we decompose $\mathbf{x}$ to $\mathbf{u}$ and $\mathbf{w}$ as follows. With the representation basis that grants the smallest ρ, $\mathbf{x}_{(k)}$ is considered to be $\mathbf{u}$, where k is max-

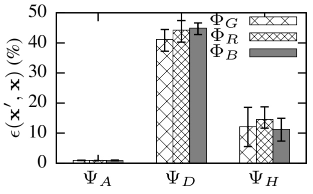


Fig. 9. Distortion of reconstruction for signals collected in pantry.

Table III. Coherence $\bar{\mu}(\Phi^*, \Psi)$

	Ψ_D	Ψ_H
Φ^*_G	0.154	0.158
Φ^*_R	0.148	0.144

† $N = 1024$

imized subject to that $\mathrm{w} = \mathrm{x} - \mathrm{x}_{(k)}$ passes the Durbin-Watson test (i.e., w is white). The SNRs in Table II explain the increasing ρ for the signal types from *duty-cycled* to *silent*. In particular, the small SNRs for *spiky* and *silent* signals result in limited sparsity gains over the original for all representation bases. Note that $\rho_{\mathbf{u}}$ cannot be accurately estimated for the signals with significantly low SNRs, as the $\mathbf{u}$ becomes hardly differentiable from the noise. Thus, the estimated $\rho_{\mathbf{u}}$ becomes no longer meaningful for low SNR signals. This explains that the results for *spiky* and *silent* signals in Table II are not perfectly consistent with Fig. 8. When the SNR is high enough, a correct choice of representation basis can greatly reduce the sparsity of a signal. From Table II, we can see that ADT and DCT obtain the minimum $\rho_{\mathbf{u}}$ for the *duty-cycled* and *periodic* signals, respectively. This is consistent with the results in Fig. 8. Fig. 8 also shows the overall results when all the types of signals are used. HWT outperforms the other two transforms in an average sense. Thus, Ψ_H is a preferable default basis when the prior knowledge of the power signal structures is not available.

4.3. Design of Measurement Matrix Φ

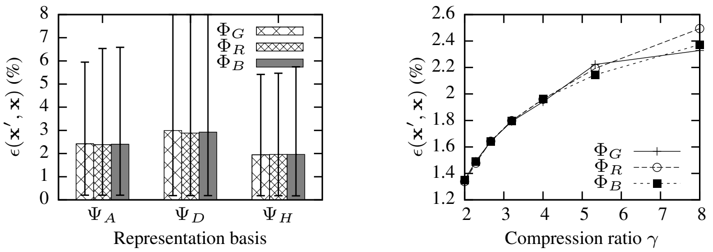
From Eq. (1), to achieve better compression ratios, Φ and Ψ should be chosen to jointly reduce the signal sparsity and coherence. In this paper, we consider three random measurement matrices that have been shown to satisfy the condition in Eq. (1) [Candès et al. 2006; Berinde et al. 2008] and employed in various applications [Wu and Liu 2012]:

Gaussian matrix Φ_G: Each element of Φ_G is drawn from the normal distribution $\mathcal{N}(0, \frac{1}{M})$, where M satisfies Eq. (1).

Rademacher matrix Φ_R: Each element of Φ_R is either $\frac{1}{\sqrt{M}}$ or $-\frac{1}{\sqrt{M}}$ with a probability of 0.5, where M satisfies Eq. (1).

Binary matrix Φ_B: Each column of Φ_B has S ones and $(M - S)$ zeros. The positions of ones are uniformly distributed. Φ_B satisfies a relaxation to the condition in Eq. (1) [Berinde et al. 2008].

We conduct extensive empirical studies on the performance of the combinations of the above three measurement matrices and the three representation bases described in Section 4.2. The compression ratio and distortion of the recovered signal are two important but competing performance metrics for signal compression. In this paper, the distortion is assessed by $\epsilon(\mathrm{x}', \mathrm{x})$, where x' is the recovered signal and $\epsilon(\cdot, \cdot)$ is defined in Eq. (3). We first evaluate the distortion for the *duty-cycled* signals collected in the pantry under different combinations of Φ and Ψ. The result in Fig. 9 shows that ADT obtains the smallest distortion across all the bases, which is consistent with Fig. 8. Table III lists the normalized coherence defined in Eq. (2) for different combinations of Φ and Ψ. Note that as Ψ_A and Φ_B are not orthonormal, the coherence involving them is undefined. Φ_R shows smaller coherence than Φ_G, but the difference is not significant. A similar trend can be found in Fig. 9, where the choice of Φ barely affects the distortion.



(a) $\gamma = 4$, $S = 8$

(b) $\Psi = \Psi_H$, $S = 8$

Fig. 10. Distortion of reconstruction (excluding pantry). Error bars represent min and max values.

Next, we evaluate the distortions for different combinations of Φ and Ψ over all the types of signal except for *duty-cycled*. We assume no prior knowledge of the structure of power readings. Fig. 10(a) shows the distortions based on different combinations of Φ and Ψ. From the figure, we can see that HWT yields a lower distortion than ADT and DCT, but the difference in distortion is not significant compared to *duty-cycled*. It also shows that the choice of measurement matrix has negligible impact on the distortion. This result applies for a range of compression ratios. Fig. 10(b) shows the distortion of HWT under different measurement matrices versus compression ratios. While the distortion increases with γ, it is similar among the measurement matrices. In summary, we have the following important insights on choosing Φ and Ψ. The Φ does not significantly affect the performance of JICE. If the signal structure is known, the optimal choice of Ψ can improve greatly the performance; otherwise, HWT yields a marginal performance gain over ADT and DCT on average.

4.4. Adaptive Reconfiguration of Ψ

Motivated by the observation in Section 4.3, we design a lightweight adaptive reconfiguration approach to select the most efficient Ψ for each signal block at run time.

4.4.1. Signal Feature Extraction. Our approach is based on a three-dimensional feature vector $\mathrm{f} = [r_{\mathrm{ac}}, r_{\mathrm{sc}}, \sigma]$ for each block x, where r_{ac} is the *rate of average crossings*, r_{sc} is the *rate of sharp changes*, and σ_x is the standard deviation. These metrics are related to the signal structure. Specifically, r_{ac} is the ratio of times when the signal crosses its average value to the block size, which is related to the periodicity of the signal; r_{sc} is the ratio of the number of sharp changes to the block size, which is related to the presence of duty-cycling appliances; σ_x is related to the magnitude of fluctuation of the signal. Formally, denoting by $\nabla x_i = (\bar{\mathrm{x}} - x_i)(\bar{\mathrm{x}} - x_{i-1})$ and $\triangle x_i = |x_i - x_{i-1}|$, these features are defined as $r_{\mathrm{ac}} = \frac{1}{N}\|\{i|\nabla x_i < 0, i = 2, \cdots, N\}\|, r_{\mathrm{sc}} = \frac{1}{N}\|\{i|\triangle x_i > \delta, i = 2, \cdots, N\}\|, \sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{\mathrm{x}})^2}$, where δ is a predefined threshold (e.g., $5\,\mathrm{W}$ in our implementation) to determine a sharp change, and $\|\cdot\|$ represents the cardinality of a set. As shown in Section 4.2, the most efficient representation basis Ψ (i.e., the one minimizes sparsity ρ) is different for the signals with different structures. Thus, the feature vector f can indicate the best configuration of Ψ. In Section 4.4.2, we will present a decision-table-based approach to identify the best Ψ from f.

From the implementation details described in Section 6, the meter does not store x to reduce RAM storage overhead. To follow this efficient paradigm, we use the average value of the previous block as $\bar{\mathrm{x}}$ to calculate σ_x, such that r_{ac}, r_{sc}, and σ_x can be computed in an incremental online manner.

Table IV. Threshold-based decision table.

$r_{\text{ac}}>T_1$? (T_1=0.095)	N	N	N	N	Y	Y	Y	Y
$r_{\text{sc}}>T_2$? (T_2=0.059)	N	N	Y	Y	N	N	Y	Y
$\sigma_x > T_3$? (T_3=51)	N	Y	N	Y	N	Y	N	Y
choice for Ψ	Ψ_D	Ψ_A	Ψ_H	Ψ_A	Ψ_H	Ψ_H	Ψ_H	Ψ_H

Table V. Structure features and chosen representation bases for the signals shown in Fig. 7.

	duty-cycled	**period**	**fluctuating**	**spiky**	**silent**
r_{ac}	0.00586	0.01758	0.04297	0.39844	0.00391
r_{sc}	0.02051	0.04004	0.62402	0.19629	0.03516
σ_x	247.5	31.4	30.3	5.1	0.3
Ψ	Ψ_A	Ψ_D	Ψ_H	Ψ_H	Ψ_D

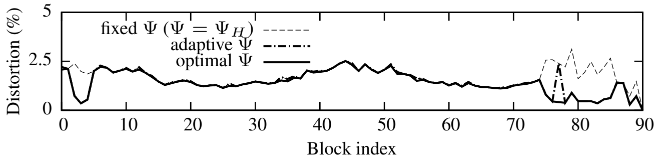


Fig. 11. Distortions of JICE with fixed Ψ, adaptive Ψ reconfigured by the decision table, and optimal Ψ.

4.4.2. Adaptive Reconfiguration of Ψ. Our approach aims to determine the most efficient Ψ based on f. As it is difficult to discover the statistical distributions for f, statistical classifiers (e.g., Bayesian) are not applicable. Moreover, the classification based on complex decision boundaries of these classifiers will impose substantial computational overhead for meters, potentially negating the benefit brought by reconfiguring Ψ at run time. In our approach, we adopt a threshold-based *decision table* to determine Ψ from f, which is a look-up table according to the results of comparing r_{ac}, r_{sc}, and σ_x with three thresholds (T_1, T_2, and T_3). Table IV shows such a table. The thresholds T_1, T_2, T_3, and the last row of the decision table are obtained by a training algorithm based on a training data set, which will be presented in Section 4.4.3. Table IV also includes the training results based on a half of the data set described in Section 4.2. Table V reports the feature vectors for the signals shown in Fig. 7 and the Ψ chosen according to the decision table in Table IV, which are the same as the optimal results. We note that the most efficient Ψ may not be consistent with the results shown in Fig. 8 (e.g., spiky and silent), which consider sparsity only. Fig. 11 plots the distortions with a fixed Ψ and an adaptive Ψ reconfigured using the decision table in Table IV, based on the other half of the data set. It also shows the distortions under the true optimal Ψ. Compared with a fixed Ψ, an adaptive Ψ can reduce the distortion effectively for 15 blocks out of totally 90 blocks. For one block only, our approach chooses a Ψ that is not necessarily the optimal one.

4.4.3. Decision Table Training Algorithm. This section presents an autonomous algorithm to train the threshold-based decision table, that is, to determine the thresholds T_1, T_2, T_3, and the choice of Ψ (i.e., the last row of Table IV). The basic idea of the algorithm is to minimize the average *extra* distortion in CS recovery due to wrong Ψ choices. Let $\mathbf{X}$ denote the training data set. For each block $\mathbf{x} \in \mathbf{X}$, we pre-compute the distortions under Ψ_A, Ψ_D, and Ψ_H, which are denoted by $\epsilon_{\Psi_A}(\mathbf{x})$, $\epsilon_{\Psi_D}(\mathbf{x})$, and $\epsilon_{\Psi_H}(\mathbf{x})$. Let $\Psi^*_{\mathbf{x}}$ denote the optimal representation basis for $\mathbf{x}$, i.e., the one with the smallest distortion.

With certain thresholds T_1, T_2, and T_3, we define $Z(\mathbf{x})$ for $\mathbf{x}$ as a three-bit integer, where the bits are given by the boolean results of the comparisons $r_{\text{ac}} > T_1$, $r_{\text{sc}} > T_2$, and $\sigma_x > T_3$, respectively. Formally, $Z(\mathbf{x}) = (\mathbf{1}_{r_{\text{ac}}>T_1}\mathbf{1}_{r_{\text{sc}}>T_2}\mathbf{1}_{\sigma_x>T_3})_2$, where $\mathbf{1}_c = 1$ if the

Algorithm 1 Decision table training algorithm

```
Input: training data set $\mathbf{X}$
Output: decision table consisting of the thresholds $T_1^*$, $T_2^*$, $T_3^*$, and the last row of $\Psi$
    choices denoted by $M^*(z), \forall z \in [0,7]$
min_total_increased_distortion = DBL_MAX    // initialize to a large number
for $T_1 \in [0,1]$, $T_2 \in [0,1]$, and $T_3 \in [T_3^{\min}, T_3^{\max}]$ do
    for $z \in [0,7]$ do
        $S_z = \emptyset$    // initialize to empty set
    end for
    for each block $\mathbf{x}$ in training data set $\mathbf{X}$ do
        $\Psi^*_{\mathbf{x}} = \arg\min_{\Psi \in \{\Psi_A, \Psi_D, \Psi_H\}} \epsilon_\Psi(\mathbf{x})$    // $\Psi^*_{\mathbf{x}}$ is optimal $\Psi$ for $\mathbf{x}$
        compute $r_{\text{ac}}$, $r_{\text{sc}}$, $\sigma_x$ of $\mathbf{x}$
        compute $Z(\mathbf{x}) = (\mathbf{1}_{r_{\text{ac}} > T_1} \mathbf{1}_{r_{\text{sc}} > T_2} \mathbf{1}_{\sigma_x > T_3})_2$    // classify $\mathbf{x}$ using $T_1$, $T_2$, $T_3$
        $S_{Z(\mathbf{x})} = S_{Z(\mathbf{x})} \bigcup \{\mathbf{x}\}$
    end for
    for $z \in [0,7]$ do
        $M(z) = \arg\min_{\Psi \in \{\Psi_A, \Psi_D, \Psi_H\}} \sum_{\mathbf{x} \in S_z} \epsilon_\Psi(\mathbf{x}) - \epsilon_{\Psi^*_{\mathbf{x}}}(\mathbf{x})$    // optimal $\Psi$ for $z^{\text{th}}$ class
        $\mathcal{E}(z) = \sum_{\mathbf{x} \in S_z} \epsilon_{M(z)}(\mathbf{x}) - \epsilon_{\Psi^*_{\mathbf{x}}}(\mathbf{x})$
    end for
    if min_total_increased_distortion $> \sum_{z=0}^{7} \mathcal{E}(z)$ then
        min_total_increased_distortion = $\sum_{z=0}^{7} \mathcal{E}(z)$
        $T_1^* = T_1$, $T_2^* = T_2$, $T_3^* = T_3$, $M^*(z) = M(z), \forall z \in [0,7]$
    end if
end for
```

condition c is true and $\mathbf{1}_c = 0$ otherwise. For instance, if for a block $\mathbf{x}$ we have $r_{\text{ac}} < T_1$, $r_{\text{sc}} > T_2$, and $\sigma_x > T_3$, then $Z(\mathbf{x}) = (011)_2 = 3$. Thus, an $\mathbf{x}$ is classified into any of the 2^3 classes in the decision table, where $Z(\mathbf{x})$ represents the class index. Define S_z as the subset of blocks classified into the z^{th} class. Formally, $S_z = \{\mathbf{x} | \forall \mathbf{x} \in \mathbf{X}, Z(\mathbf{x}) = z\}$, where $z = 0, 1, \ldots, 7$. Thus, any two subsets have no overlap and the training data set is the union of all the eight subsets. Formally, $S_z \cap S_y = \emptyset$ if $z \neq y$; and $\bigcup_{z=0}^{7} S_z = \mathbf{X}$. Define $M(z) = \arg\min_{\Psi \in \{\Psi_A, \Psi_D, \Psi_H\}} \sum_{\mathbf{x} \in S_z} \epsilon_\Psi(\mathbf{x}) - \epsilon_{\Psi^*_{\mathbf{x}}}(\mathbf{x})$. Note that $\epsilon_\Psi(\mathbf{x}) - \epsilon_{\Psi^*_{\mathbf{x}}}(\mathbf{x})$ is the *increased distortion* for block $\mathbf{x}$ if the representation basis Ψ rather than its optimal representation basis $\Psi^*_{\mathbf{x}}$ is used. Thus, this increased distortion is always non-negative. Therefore, $M(z) \in \{\Psi_A, \Psi_D, \Psi_H\}$ is the representation basis that minimizes the total increased distortion if $M(z)$ is used as the representation basis consistently for all blocks in S_z. For the z^{th} class, we define the corresponding total increased distortion as $\mathcal{E}(z) = \sum_{\mathbf{x} \in S_z} \epsilon_{M(z)}(\mathbf{x}) - \epsilon_{\Psi^*_{\mathbf{x}}}(\mathbf{x})$. We iterate T_1, T_2, and T_3 in their possible ranges to minimize $\sum_{z=0}^{7} \mathcal{E}(z)$ (i.e., the overall increased distortion for all eight classes), yielding the optimal settings for T_1, T_2, and T_3. Under these optimal settings, $M(z)$ gives the representation basis that should be chosen for any block in the z^{th} class (i.e., $Z(\mathbf{x}) = z$). Thus, $M(z)$ where $z = 0, 1, \ldots, 7$, gives the last row of Table IV.

Both the ranges of T_1 and T_2 are $[0, 1]$, which can be discretized to facilitate the search. The historical smallest and largest standard deviations with safeguard margins can be used to set the search range for T_3. Denote the search range for T_3 as $[T_3^{\min}, T_3^{\max}]$. Algorithm 1 provides the pseudocode of the decision table training algorithm. Denoting by n the search range size for the thresholds and by $\|\mathbf{X}\|$ the training data set size, the complexity of the algorithm is $O(n^3 \cdot \|\mathbf{X}\|)$.

4.4.4. Practical Issues. We now discuss a few practical issues for the adaptive Ψ reconfiguration approach. First, unlike many training-based approaches, our approach needs no ground-truth labels for the training data, since the most efficient Ψ can be identified autonomously by the training algorithm, simply by comparing the distortion under different representation bases (i.e., Line 7 in Algorithm 1). Thus, the training can be fully automated. Second, the decision table can be updated periodically (e.g., hourly) by the gateway using the training algorithm and the recently reconstructed signals as training data, such that the WEAN can adapt to changing characteristics or usage patterns of appliances. The updated decision table can be transmitted to the meter, which introduces little overhead (14-byte payload in our implementation). Alternatively, the decision table can be updated when there is a change of usage pattern. The change of usage pattern can be detected by tracking the distribution of the decision results over the eight classes shown in Table V. Compared with the periodic update approach, this on-demand approach can avoid the decision table training when the usage pattern remains unchanged. In JICE, the gateway computes different decision tables for different meters based on their own data.

5. DATA SECRECY OF COMPRESSIVE SENSING

This paper considers a wireless eavesdropping threat model, where an eavesdropper can capture the CS-compressed signal transmitted over air from a power meter to the gateway and aims to recover the original signal. In this section, we address two secrecy vulnerabilities of CS to wireless eavesdropping and develop countermeasures. The first vulnerability is a leak of several statistics of the original signal. To fix this, we propose a perturbation approach. The second vulnerability is the exposure of the random matrix, which is used as a secret key, in a variation of the known-plaintext attack. To fix this, we analyze how frequently the random matrix should be updated, such that the adversary cannot accumulate enough data to implement the attack.

5.1. Basic Secrecy Property Achieved by CS

With the CS-compressed signals only, recovering the original signals is computationally hard, as long as the random measurement matrix Φ is kept secret [Rachlin and Baron 2008; Orsdemir et al. 2008]. A symmetric secret key shared by the meter and the gateway can be used as the seed to generate Φ, where the symmetric key can be hard-coded or established using existing code libraries for key exchange (e.g., [Liu and Ning 2008]) that are often based on public-key cryptography. We note that, although establishment of the symmetric key may introduce some overhead due to the complexity of public-key cryptography, it is a one-time procedure during system initialization only. Different symmetric keys can be established for different meters. As such, the leak of a single meter's key (e.g., by extracting from the memory of a physically captured meter) will not reveal the keys of other meters.

5.2. Leak of Statistics under CS

5.2.1. Statistics of Compressed Signal. Although CS prevents the recovery of the original signal under wireless eavesdropping, we identify the following vulnerability of CS in leaking statistics of the original signal.

PROPOSITION 5.1. *When the Gaussian or the Rademacher matrix is adopted (i.e., $\Phi = \Phi_G$ or $\Phi = \Phi_R$), the eavesdropper can accurately estimate the ℓ_2-norm of the original signal $\|\mathbf{x}\|_{\ell_2}$ from the compressed signal $\mathbf{y} = \Phi\mathbf{x}$. They can also estimate the bounds for*

Table VI. An example of perturbation.

	Gaussian matrix Φ_G			binary matrix Φ_B
	ℓ_2-norm	$\bar{x}$	σ_x	$\bar{x}$
true value	13689	426.8	28.56	426.8
estimate (no perturbation)	14445	[14.1, 451.4]	≤451.1	426.8
estimate (k_p=5×10^3)	19773	[19.3, 617.9]	≤617.6	601.3
estimate (k_p=5×10^5)	544336	[531.5, 17010.5]	≤17002	16553.3
distortion (no perturbation)	2.47%			2.27%
distortion (k_p=5×10^3)	2.47%			2.31%
distortion (k_p=5×10^5)	2.49%			2.44%

the mean and standard deviation of $\mathbf{x}$ *(denoted by* $\bar{x}$ *and* σ_x*, respectively) as*

$$\frac{1}{N}\|\mathbf{x}\|_{\ell_2} \leq \bar{x} \leq \frac{1}{\sqrt{N}}\|\mathbf{x}\|_{\ell_2}, \tag{5}$$

$$\sigma_x \leq \frac{\sqrt{N-1}}{N} \cdot \|\mathbf{x}\|_{\ell_2}. \tag{6}$$

PROOF. By denoting y_i as the ith entry of $\mathbf{y}$, we have $y_i = \sum_{j=1}^{N} \phi_{i,j} x_j$, where x_j is the jth entry of $\mathbf{x}$ and $\phi_{i,j}$ is the (i,j)th element of Φ_G or Φ_R. For both Φ_G and Φ_R, the variance of $\phi_{i,j}$ is $\mathrm{Var}[\phi_{i,j}] = \frac{1}{M}$. As each $\phi_{i,j}$ is independent and identically distributed, we have $\mathrm{Var}[y_i] = \frac{1}{M}\sum_{j=1}^{N} x_j^2 = \frac{1}{M}\|\mathbf{x}\|_{\ell_2}^2$. Given $\mathbf{y}$, the unbiased sample variance of y_i for any i, denoted by s_y^2, is given by $s_y^2 = \frac{1}{M-1}\sum_{i=1}^{M}(y_i - \bar{y})^2$, where $\bar{y} = \frac{1}{M}\sum_{i=1}^{M} y_i$. As $\mathrm{Var}[y_i] \simeq s_y^2$, we can derive $\|\mathbf{x}\|_{\ell_2} \simeq \sqrt{\frac{m}{m-1}\sum_{i=1}^{m}(y_i - \bar{y})^2}$. In other words, the ℓ_2-norm of the original signal can be accurately estimated from the compressed signal. Based on the estimated $\|\mathbf{x}\|_{\ell_2}$, the adversary can further estimate bounds for the mean and standard deviation of $\mathbf{x}$. As $x_i \geq 0$, $\bar{x} = \frac{1}{N}\|\mathbf{x}\|_{\ell_1}$. As $\|\mathbf{x}\|_{\ell_2} \leq \|\mathbf{x}\|_{\ell_1} \leq \sqrt{N}\|\mathbf{x}\|_{\ell_2}$, we have Eq. (5). Moreover, as $\sigma_x = \sqrt{\frac{1}{N}\|\mathbf{x}\|_{\ell_2}^2 - \bar{x}^2}$, we have Eq. (6). □

PROPOSITION 5.2. *When the binary matrix is adopted (i.e.,* $\Phi = \Phi_B$*), the eavesdropper can exactly estimate* $\bar{x}$ *from the compressed signal* $\mathbf{y} = \Phi_B \mathbf{x}$.

PROOF. It is easy to verify that $\bar{x} = \frac{\sum_{i=1}^{M} y_i}{N \cdot S}$, since each column of Φ_B contains S ones. □

Note that $\bar{x}$ and σ_x represent important privacy information of the user. Table VI shows an example of the leak of statistics. We can see that, when Φ_G is used, the ℓ_2-norm of the original signal can be accurately estimated. The estimated upper bound of $\bar{x}$ is close to $\bar{x}$. When Φ_B is used, the $\bar{x}$ can be exactly estimated.

5.2.2. Perturbation. To solve the vulnerability, we propose to perturb the power signal. In JICE, the meter and gateway have a shared secret key denoted by $k_p \in \mathbb{R}$. Define a perturbation vector $\hat{\mathbf{p}} = [k_p, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^{N \times 1}$. Its time-domain counterpart is $\mathbf{p} = \Psi\hat{\mathbf{p}} = k_p\psi_1$, which can be pre-computed by the meter. The meter computes the sum of $\mathbf{x}$ and $\mathbf{p}$ to produce the perturbed signal denoted by $\widetilde{\mathbf{x}}$, i.e., $\widetilde{\mathbf{x}} = \mathbf{x} + \mathbf{p}$. The meter then applies CS to $\widetilde{\mathbf{x}}$ and transmits. As the sparsity of $\widetilde{\mathbf{x}}$, denoted by $\rho_{\widetilde{\mathbf{x}}}$, is at most $\rho + 1/N$, the extra distortion caused by the perturbation is almost negligible. Moreover, for DCT and HWT, the first transform coefficient corresponds to the lowest frequency component, which is typically non-zero. Therefore, the perturbation will not change the signal sparsity. As shown in Table VI that applies two settings for k_p, the perturbation does not lead to significant increases of distortion. As the gateway also knows $\mathbf{p}$, it can remove $\mathbf{p}$ from the reconstructed signal to obtain the original signal. When Φ_G or Φ_R

is used, the adversary can estimate $\|\mathbf{x} + \mathbf{p}\|_{\ell_2} = \|\hat{\mathbf{x}} + \hat{\mathbf{p}}\|_{\ell_2}$, but cannot estimate $\|\hat{\mathbf{x}}\|_{\ell_2}$ since $\hat{\mathbf{p}}$ contains an arbitrary number k_p. When Φ_B is used, the adversary can estimate $\bar{x} + \frac{k_p}{N} \sum_{j=1}^{N} \psi_{1,j}$. As $\sum_{j=1}^{N} \psi_{1,j} \neq 0$ and k_p is an arbitrary number, the adversary cannot estimate $\bar{x}$. From the example shown in Table VI, with perturbation, the adversary's estimates for $\|\mathbf{x}\|_{\ell_2}$ and the bounds for $\bar{x}$ and σ_x depend on k_p and they are wrong. Moreover, the perturbation causes little extra distortion in the signal recovery.

5.3. Noisy-Plaintext Attack

5.3.1. Attack Model. Because the CS-compressed data is a linear combination of the original signal, the random measurement matrix can be estimated in the known-plaintext attack. However, in WEANs, obtaining the original signal will need physical access to the power wires or retrofit to the meters, making it difficult and detectable. A more realistic threat model is that the wireless eavesdropper can measure a noisy version of the original signal, e.g., by a current transducer. As a variation of the known-plaintext attack, we call this attack model *noisy-plaintext attack*. Specifically, we assume that the adversary can measure $\mathbf{x}' = \mathbf{x} + \mathbf{s}$, where $\mathbf{s}$ is a Gaussian zero-mean noise with variance σ_s^2. Given the compressed signal $\mathbf{y}$ and the noisy signal $\mathbf{x}'$, the adversary aims to estimate the random measurement matrix Φ. Apparently, the adversary cannot solve Φ from $\mathbf{y} = \Phi(\mathbf{x}' - \mathbf{s})$ since $\mathbf{s}$ is unknown. However, if the Φ is fixed, the adversary can accumulate multiple blocks of $(\mathbf{y}, \mathbf{x}')$ and use existing approaches to estimate Φ. Specifically, let $(\mathbf{y}_i, \mathbf{x}'_i)$ denotes the ith block of the eavesdropped/measured data, we have $\mathbf{y}_{ij} = (\mathbf{x}'_i - \mathbf{s}_i)^T \cdot \Phi_j^T$ for $i = 1, 2, \ldots$, where $\mathbf{y}_{ij}$ is the jth element of $\mathbf{y}_i$ and Φ_j is the jth row of Φ. Methods such as *total least square* (TLS) can be applied to compute a good estimate of Φ_j from the above group of equations. Thus, the adversary can estimate all rows of Φ.

According to our numerical experiment, the accuracy of the TLS estimator heavily depends on the volume of the input data. This means that, with more data for a fixed Φ, the adversary can achieve a more accurate Φ estimation. To mitigate this, a countermeasure is to generate a Φ using a symmetric key every data block, so that the adversary cannot accumulate sufficient data. However, this approach would incur considerable overhead, as random matrix generation is a non-trivial computation task. A better countermeasure is to update Φ every few data blocks, to reduce the related overhead. In the following subsections, we analyze and evaluate a trade-off between the adversary's estimation accuracy of Φ and the update rate of Φ.

5.3.2. Analyzing Adversary's Φ Estimation Accuracy. Cramér-Rao bound (CRB) is a lower bound of the mean square error (MSE) for an estimated parameter. That is, the estimation of a parameter cannot be statistically accurate if its CRB is large. We use CRB as a metric to characterize how accurate the adversary can estimate Φ. Thus, we can assess the CRB under different settings for the update rate of Φ. CRB is defined as the inverse of the *Fisher information matrix* (FIM). The relationship between CRB and the MSE of an estimator is given by $MSE(\hat{\theta}) \geq CRB = I(\theta)^{-1}$, where $\hat{\theta}$ denotes the estimate of the parameter θ, and $I(\theta)$ is the FIM. A closed-form solution for $I(\theta)$ is given by $I(\theta) = \frac{(X^T X)}{(\sigma_s^2 \|\Phi_j\|_{l_2}^2)}$ [Wiesel et al. 2008], where $X = [\mathbf{x}_1 \mathbf{x}_2 ... \mathbf{x}_k]^T$ is a collection of the original signals. The CRB of the ith entry of Φ_j is given by the ith element in the diagonal of the matrix $I(\theta)^{-1} = (\sigma_s^2 \|\Phi_i\|_{l_2}^2)(X^T X)^{-1}$.

5.3.3. A Numerical Example. We present a set of numerical results to illustrate the effectiveness of the noisy-plaintext attack by evaluating the distortion of the signal recovered using the estimated Φ. In this example, we use the Gaussian matrix with variance of $\frac{1}{64}$ as Φ and TLS as the adversary's estimator. The original signal is taken from

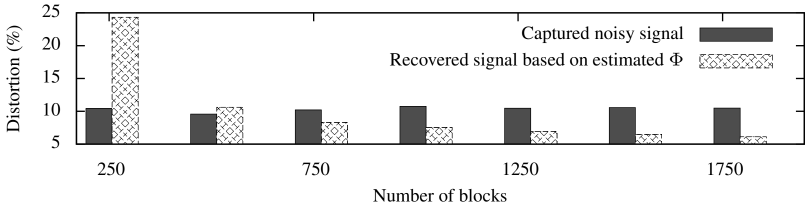


Fig. 12. Distortion of the captured noisy signal and the recovered signal using the estimated Φ in the noisy-plaintext attack.

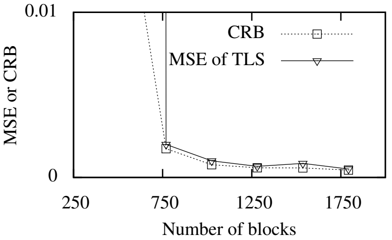


Fig. 13. The CRB and MSE of TLS estimation.

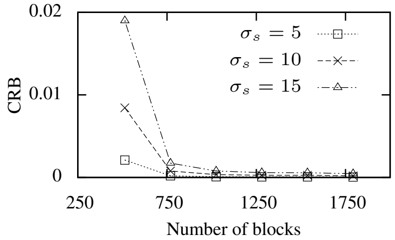


Fig. 14. CRBs under different noise levels.

the pantry (cf. Fig. 7(a)), and for the noisy signal acquired by the adversary, we add Gaussian noise to the original signal. The block size for CS is 256, and the compression ratio is 4. We gradually increase the volume of the adversary's eavesdropped data, and compare the distortion rates of the recovered signal and the captured noisy signal. Fig. 12 shows the result. Interestingly, we can see that, with sufficient data blocks, the adversary can use the estimated Φ to recover the original signal and achieve a distortion rate lower than that of the captured noisy signal. In particular, the adversary can achieve a distortion rate of 6% when they are provided with 1792 blocks of the noisy version of the original signal with a distortion rate of 10%.

We also evaluate the tightness of CRB. Fig. 13 shows the CRB and the MSE achieved by TLS versus the number of data blocks used by the adversary to estimate Φ. We can see that CRB, MSE, and the gap between them decrease with the number of blocks. As suggested by this result, when the number of blocks is small, the CRB may be loose. Thus, the CRB-based assessment of the adversary's Φ estimation accuracy may be conservative. The conservative assessment will not lead to a wrong setting for the Φ update rate, given a required level of MSE. We also assess the CRB under different noise levels as shown in Fig. 14. Consistent with intuition, the CRB increases with the noise level. Moreover, we can see that the CRB decreases drastically when the number of blocks increases from 512 to 768. This result suggests that, under the specific settings of this numerical experiment, the adversary's ability to estimate Φ is significantly weakened if we update Φ every 512 blocks instead of 768. This approach can also be applied to choose the Φ update rate under other settings.

6. IMPLEMENTATION AND BENCHMARKING

We implemented JICE on a smart plug platform called SPlug [Sonnonet 2011]. An SPlug consists of a Kmote and a power sensor (ADI ADE7763). The Kmote consists of a TI MSP430F1611 MCU (8MHz clock rate and 10KB RAM) and a Chipcon CC2420 Zigbee radio, and runs the TinyOS operating system. To evaluate JICE, we implemented the pipeline approach discussed in Section 3.3, a downsampling approach, and a lossless compression approach as baselines. We have released a code package [Chiu and Nguyen 2015] including all these implementations. This section presents the implementation details and measures the computation and RAM storage overhead of different approaches. The overhead measurement results validate that JICE has low computation and space complexities. The performance improvements by adopting the JICE implemented in this section will be evaluated through testbed experiments in Section 7.

6.1. Implementation Details

For all the approaches, data is represented as 4-byte integers or floating point numbers. To preserve data fidelity, we adopt a reliable transmission protocol called Packet Link Layer [David M. 2007], which retransmits a packet if its acknowledgment is not received in time. The default CSMA-based MAC protocol in TinyOS is used.

JICE: JICE uses the double buffering illustrated in Fig. 4 to interface the compression/encryption and the transmission. The sizes of the output and transmission buffers are M. When the meter obtains a power reading, it generates a random number, multiplies it with the reading, and adds the result to an entry of the output buffer. For the same reading, it repeats this process for each entry of the output buffer. We generate Gaussian and Rademacher random numbers by Box-Muller transform and thresholding based on uniform pseudo-random numbers, respectively. For the binary measurement matrix, we implement the approach described in [Berinde et al. 2008] to generate binary random numbers. For every N sensor readings, the meter stops sending from the current transmission buffer, switches the roles of the transmission and output buffers by swapping pointers to them (which avoids costly data copying), and starts transmitting from the new transmission buffer. In a packet to the gateway, the meter piggybacks the selection of the representation basis Ψ according to the decision table and a sequence number to synchronize the measurement matrix generations between the meter and gateway, which takes one byte of payload. In our current implementation, for each meter-gateway pair, we manually assign a shared secret key k_p that is used to generate the perturbation vector as described in Section 5.2.2. This manual approach can be improved by employing a key-agreement protocol such as the Diffie-Hellman key exchange protocol.

Pipeline: The pipeline approach employs a wavelet-based lossy compression algorithm that captures the main principle of most lossy compression schemes. It first computes the transform $\hat{\mathrm{x}}$ from the original signal x, then encodes the largest $\frac{M}{2}$ transform coefficients along with their positions in $\hat{\mathrm{x}}$, resulting in a total of M numbers. This approach employs the Advanced Encryption Standard (AES) algorithm, a representative symmetric-key cipher. Although the SPlug has built-in AES implementation in its CC2420 radio chip, this is not necessarily true of all meters. To preserve the generality of our results, we use a software implementation of AES [Pelissier 2010]. In fact, our tests show that the built-in AES is slower than the software implementation. The pipeline approach has two variants regarding the implementation of transforms. A few transforms such as ADT and HWT have efficient implementations without resorting to matrix-vector multiplication. We refer to the resultant variant as *native pipeline*. Other transforms may involve intensive floating-point computation (e.g., cosine in DCT) that incurs unacceptable delays. Instead, they can be implemented as a multiplication of x and a pre-computed Ψ, which incurs $O(N^2)$ RAM storage overhead, however. We refer to the resultant variant as *matrix pipeline*. We implement both the *native* and *matrix* versions of HWT, which allows us to understand the impact of RAM storage overhead on the overall performance. The pipeline approach also uses the double buffering to coordinate data processing and transmission.

Downsampling: This approach transmits raw data to the gateway without processing. For fair comparisons, we downsample the signal in this implementation to have the same transmission volume as those of the JICE and pipeline approaches. We use a

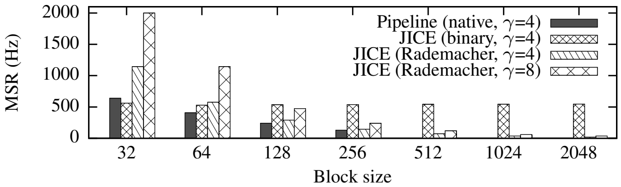


Fig. 15. Maximum sampling rate (MSR) under various block sizes.

Table VII. Maximum block size (unit: samples)

JICE	**JICE**	**JICE**	**Native**	**Matrix**	**Down-**	**Loss-**
(γ=2)	(γ=4)	(γ=8)	**pipeline**	**pipeline**	**sampling***	**less**
1024	2048	4096	128	16	2048	256

* For downsampling approach, the number is the maximum size of the circular queue.

† For the native pipeline, matrix pipeline, and down sampling approaches, $\gamma = 4$.

circular queue to coordinate the sensor sampling and data transmission. A data packet encapsulating new sensor readings is added to the queue. An infinite loop removes a packet from the queue and transmits it to the gateway. When a new packet is available and the queue is full, the meter stops sending the oldest packet, removes it, and adds the new packet to the queue.

Lossless: This approach first applies SLZW [Sadler and Martonosi 2006], a lossless compression algorithm designed for embedded sensors, then applies AES to encrypt the compressed data. As the compression ratio of SLZW is unpredictable, the lengths of all the signal buffers are set to be N to prevent overflow.

6.2. Benchmarking

We use the *maximum sampling rate* (MSR) to inversely characterize the computational overhead of an approach, which is defined as the reciprocal of the average processing time for a single reading. Fig. 15 plots the MSR of various approaches versus block size. As the computation complexity of HWT is superlinear with respect to N, we can see that the MSR of the native pipeline approach decreases with the block size. For JICE with binary matrix, the number of ones in each column is fixed. Hence, the number of arithmetic operations for each reading is fixed and the MSR is independent of the block size. For JICE with Rademacher matrix, the number of arithmetic operations for each reading is proportional to M, which depends on the block size and compression ratio. Thus, from Fig. 15, the MSR of JICE with Rademacher matrix decreases with the block size and increases with γ. Since Gaussian number generators, such as the Box-Muller transform used in our implementation, are often computation-intensive, the MSR of JICE with Gaussian matrix is low ($3\,$Hz when the block size is 128). Thus, the Gaussian measurement matrix is not suitable for capability-limited meters. In summary, from Fig. 15, JICE with either binary or Rademacher matrix consistently outperforms the pipeline approach. As a larger block size is preferable and the binary matrix leads to the highest MSRs for large block sizes, in the rest of this paper, we will adopt the binary matrix. Note that the MSR of JICE can be further increased if the meter generates a new measurement matrix Φ every few blocks instead of every block as in the current implementation.

We use the *maximum block size* (MBS) that can be achieved by an approach to inversely characterize the RAM storage overhead. The MBS is obtained by increasing the block size until the RAM usage reported by the TinyOS compiler exceeds the RAM capacity. Table VII lists the MBSs of various approaches. Note that the MBS is a power

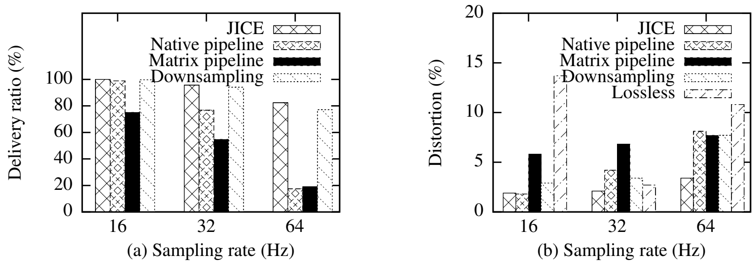


Fig. 16. Data delivery ratio and distortion vs. sampling rate ($\gamma = 4$).

of two, which is a requirement of HWT. From the table, we can see that JICE has larger MBSs than the pipeline and lossless approaches.

In summary, JICE with binary matrix has lower computation and space complexities than the pipeline approach. For the pipeline approach with $\gamma = 4$, the MBS is 128 to meet the RAM capacity constraint. From Fig. 15, this MBS may not be achieved if the required sampling rate is high. For instance, if the required sampling rate is 500 Hz, the pipeline approach needs to adopt a block size of 32. In contrast, JICE can achieve a block size of 2048 from Fig. 15 and Table VII. Such a much larger block size will make JICE more resilient to wireless link quality deterioration as shown in Section 3.3.

7. TESTBED EXPERIMENTS

In this section, we conduct extensive testbed experiments to compare the performance of the different approaches' implementations presented in Section 6.

7.1. Experiment Methodology

To make the results of various approaches (e.g., the recovered signals) comparable, we need them to measure the same power signal. As an SPlug has both plug and socket interfaces, multiple SPlugs can be connected in series to measure the same appliance. We use five SPlugs loaded with JICE, native pipeline, matrix pipeline, downsampling, and lossless approaches, respectively, and a sixth SPlug loaded with a ground-truth data collection program. Note that each approach adopts its MBS shown in Table VII. The ground-truth SPlug transmits the raw data without any processing. We use two gateways, which are two Kmotes connected to two computers. The five SPlugs with the JICE and baseline approaches communicate with a gateway using the same Zigbee channel, while the ground-truth SPlug communicates with the other gateway using another Zigbee channel. We use our setup to measure the power consumption of a 29-inch LCD that repeatedly displays a video. In this section, we evaluate the performance of JICE and the baseline approaches under various settings for sampling rate, compression ratio, and network size. For each setting, we conduct an experiment run that lasts for one hour.

7.2. Experimental Results

7.2.1. Data Delivery and Fidelity. Fig. 16(a) shows the data delivery ratios of the various approaches under different sampling rates. We can see that the pipeline approaches do not scale well with the sampling rate. In contrast, JICE maintains a much better data delivery ratio (82%) than the pipeline approach when the sampling rate is 64 Hz. When the sampling rate increases, each SPlug will transmit more data, causing increased wireless channel contention. As shown in the illustrating example in Section 3.3, JICE can better handle deteriorated link quality, which explains the higher data delivery ratios of JICE in Fig. 16(a). As the downsampling approach uses a large circular buffer

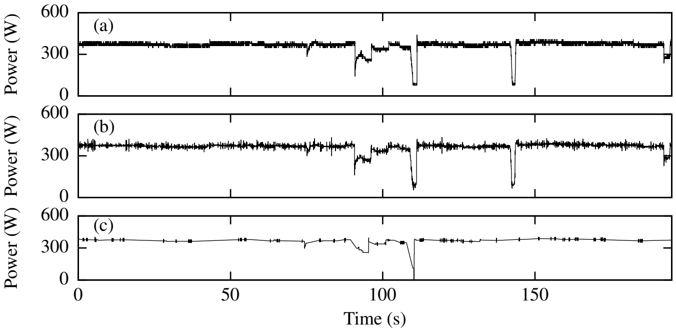


Fig. 17. Signal recovery example. (a) Ground truth; (b) Recovered by JICE; (c) Recovered by native pipeline.

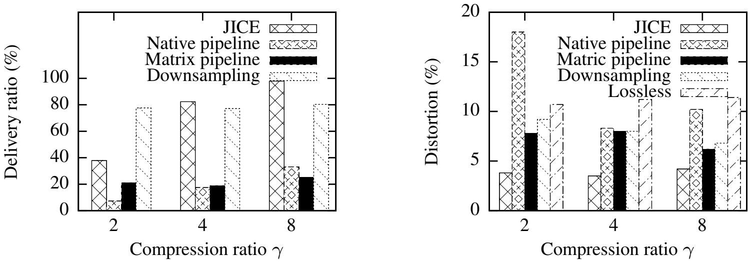


Fig. 18. Data delivery ratio and distortion vs. γ (sampling rate: 64 Hz).

which tolerates variable link quality, it yields comparable data delivery ratios as JICE. Note that as the lossless approach generates a variable number of packets, we omit its data delivery ratio.

Fig. 16(b) shows the distortions of the signals recovered by the various approaches. Note that the gateway can detect lost packets from the sequence numbers in the received packets. Under JICE, the gateway uses the rows in $\Phi\Psi$ that correspond to the received data points only to recover the signal. For the pipeline approaches, the lost coefficients are set to zeros. For the downsampling approach, omitted and lost readings are interpolated. For the lossless approach, a whole block is discarded if any packets are lost since SLZW requires complete data for decompression. We can see that JICE generally yields the lowest distortions. An exception is the native pipeline approach when the sampling rate is $16\,\text{Hz}$. Under this setting, both JICE and native pipeline have nearly 100% data delivery ratios and hence their distortions are comparable. Fig. 17 plots segments of the ground truth and recovered signals by JICE and the native pipeline, at a sampling rate of $32\,\text{Hz}$ and $\gamma = 4$. We can see that JICE well preserves the shape of the signal whereas the native pipeline approach has significant recovery errors.

We conduct another set of experiments similar to the one in Fig. 16, except that we fix the sampling rate to $64\,\text{Hz}$ and vary the compression ratio. The results are shown in Fig. 18. When $\gamma = 2$, JICE has a smaller block size (cf. Table VII) and more data to be sent. As a result, JICE experiences a low data delivery ratio (38%). However, from Fig. 18(b), the distortion of JICE is just 4%. For JICE, the effect of packet loss is similar to that of choosing a larger γ. Therefore, the distortions of JICE for $\gamma = 2$ and $\gamma = 4$ are comparable since the data delivery ratio is doubled when γ increases to 4. In summary, JICE outperforms the baseline approaches in terms of distortion.

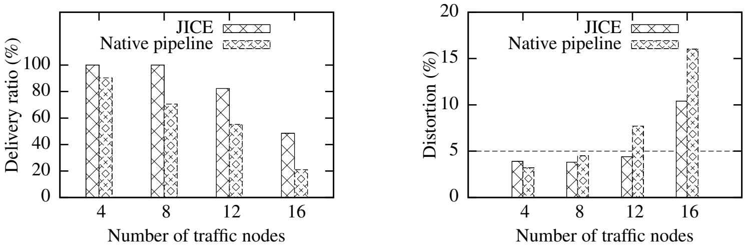


Fig. 19. Data delivery ratio and distortion vs. the number of traffic nodes ($\gamma = 4$, sampling rate = 8 Hz)

7.2.2. Scalability. This section evaluates the scalability of JICE with respect to the network size. For this testbed experiment, we tune down the transmission power of the SPlugs to simulate long distances from the gateway in real large-scale networks. The resultant RSSIs at the gateway are within $[-40, -30]$, a typical range observed in practice when meters use the maximum transmission power. To simulate a large number of meters, we use several Kmotes as *traffic nodes*, which continuously transmit packets. Fig. 19 plots the data delivery ratio and distortion versus the number of traffic nodes, when the sampling rate is $8\,$Hz. With more traffic nodes, the wireless channel contention is severer. JICE outperforms the native pipeline approach except when the number of traffic nodes is 4. Under this setting, both approaches have comparable delivery ratios and the pipeline approach has a slightly lower distortion. From Fig. 19(b), to maintain a distortion of 5%, JICE can increase the supported meters by 50%, compared with the pipeline approach. According to the generated traffic volume, each traffic node can be projected to 12 nodes running the JICE or pipeline approaches sampling at $8\,$Hz. Thus, to maintain a distortion of 5%, JICE supports up to 144 meters.

8. JICE FOR VOLTAGE MONITORING AND OTHER APPLICATIONS

In this section, we extend JICE to address voltage monitoring and discuss several other scenarios where JICE can be potentially used.

8.1. JICE for Voltage Monitoring

8.1.1. Background of Voltage Monitoring. Real-time monitoring of voltage waveforms at different locations of a power transmission/distribution system can provide crucial information for power system condition assessment. Several large-scale voltage monitoring projects have been initiated in the past decade. For instance, right after the 2003 blackout in North America, more than 1,000 phasor measurement units (PMUs) have been deployed in power systems to monitor abnormal voltage disturbances. This deployment is known as the North American SynchroPhasor Initiative (NASPI). These PMUs collect real-time voltage waveforms (also called *synchrophasor* data) and transmit them to phasor data concentrators to generate real-time grid condition reports. Besides, in 2014, Cisco announced the deployment of a voltage monitoring system called LineWatch [QinetiQ 2014] to monitor the voltage waveforms in its power distribution networks [John 2014].

To reduce deployment overhead and cost, as well as to enable one-shot diagnostic installations, it is desirable to adopt a wireless design, i.e., transmit the voltage waveform data from meters to gateways via wireless communications. In contrast to power signals, voltage signals are often sampled at relatively higher rates, e.g., $30\,$Hz adopted by NASPI. Higher sampling rates are needed for monitoring voltage harmonics, which are caused by non-linear loads and efficiently indicate power quality [Santoso et al. 1996]. For instance, to capture the 3rd harmonic component, a sampling rate of at least three times the nominal grid frequency is needed. Because of the high sampling rates, data compression before the wireless transmission is highly desirable to ensure the fidelity of the collected voltage data.

Voltage waveforms are also sensitive data that needs protection. For instance, given the PMU data at certain sites of the power grid, the topology of the whole grid and the power system state estimation model can be inferred [Baldwin et al. 1993]. With such knowledge, the adversary can launch undetectable attacks that perturb the system's state, making the power grid unstable [Liu et al. 2011]. Moreover, by studying transient patterns in voltage waveforms, electrical activities including the operations of specific appliances can be identified [Cox et al. 2006].

In summary, the above data fidelity and secrecy requirements of voltage monitoring motivate the use of JICE for this application. Since the secrecy analysis of JICE in Section 5 is general, in this section, we focus on the design of the measurement matrix and the representation basis for data fidelity.

8.1.2. Design of JICE for Alternating Current Voltage Signals. In Section 4, we have demonstrated how JICE adapts to various power signals by autonomously choosing an appropriate representation basis. Voltage signals pose different challenges for CS-based compression. Since voltage signals are periodic time series data, we choose the DCT as the representation basis. The DCT transform of a voltage signal will have a large spike at the nominal grid frequency. According to [Candès et al. 2008], large coefficients often negatively affect the performance of most CS recovery algorithms. Several re-weighting algorithms have been proposed to mitigate this effect [Candès et al. 2008]. However, these algorithms often introduce significant computational overhead [Candès et al. 2008]. Instead, we adopt an efficient and simple regularization approach. It is based on a smoothing filter called *Hamming window*. Because the DCT coefficients of the original voltage signal are normally large at low frequencies and small at high frequencies, we aim to reduce the amplitudes of the large coefficients, while preserving the small ones. Thus, we design a diagonal N-by-N matrix H with the diagonal elements given by $H_{nn} = \frac{1}{0.54+0.46\cdot\cos(\frac{\pi n}{N-1})}$, which is the reciprocal of a left-half element in the Hamming window. The recovery is based on a new representation basis $\Psi'_D = H\Psi_D$. We conduct an experiment to evaluate this regularization approach. Fig. 20(a) plots a 50 Hz voltage signal measured in our office. Fig. 20(b) and Fig. 20(c) show the recovered voltage signals using the original and regularized DCT bases, respectively. We can observe that the regularization approach can significantly improve the reconstruction quality.

8.1.3. Evaluation. As discussed in Section 8.1.1, voltage harmonics provide important information about power quality and load activities. Thus, the ability of preserving voltage harmonics is an important performance metric. In this section, we conduct a set of experiments to evaluate the performance of JICE in the presence of significant voltage harmonics. Our evaluation methodology is as follows. We inject harmonic waves into a fundamental waveform which is a pure 50 Hz sinusoid wave, and adjust the magnitudes of the harmonic waves to evaluate the ability of JICE in preserving the harmonics. The *total harmonic distortion* (THD) is a widely used metric to characterize the magnitudes of harmonic components. It is defined as $\text{THD} = \frac{\sqrt{\sum_{n=1}^{\infty} V_n^2}}{V_1}$, where V_n represents the amplitude of the nth harmonic component and V_1 is the amplitude of the fundamental sinusoid wave. In this experiment, only the 3rd, 5th, and 7th harmonics are considered. Moreover, for simplicity, all the harmonics are configured with the same amplitude. For instance, to create a distorted sinusoid of 5% THD, the amplitude of each injected harmonic is approximately 2.9% of the fundamental sinusoid wave.

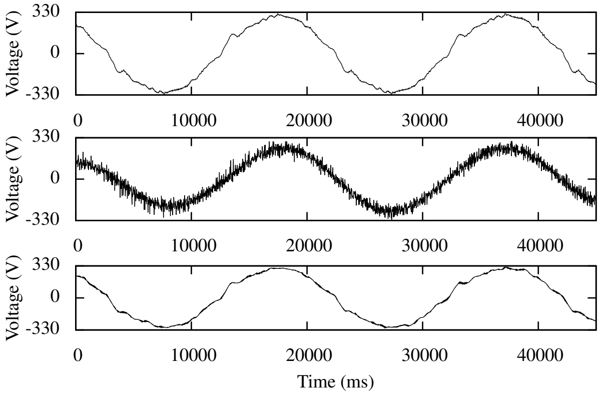


Fig. 20. (a) Original voltage signal; (b) Reconstructed signal using DCT; (c) Reconstructed signal using regularized DCT.

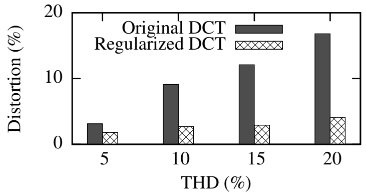


Fig. 21. Recovery distortion in the presence of harmonics of different levels.

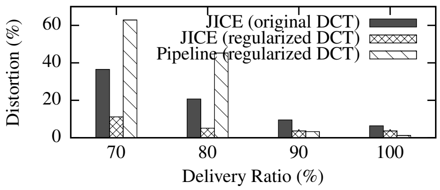


Fig. 22. Recovery distortion in the presence of harmonics versus packet delivery ratio.

The block length for the CS compression is equivalent to 4 cycles of the fundamental sinusoid wave, and the compression ratio is 8.

Fig. 21 shows the recovery distortion at the gateway under different THD settings when JICE adopts the original and the regularized DCT basis. We note that many electric appliances require that THD is less than 5% [Csanyi 2016]. However, large non-linear load and power electronics in industrial settings can increase THD beyond 5%. From Fig. 21, when THD is 5%, the regularized DCT can help JICE reduce the recovery distortion rate by half, compared with the original DCT. Moreover, with the regularized DCT, JICE maintains small recovery distortion rates under larger THD settings. This result shows that regularized DCT is more effective in preserving the harmonics than the original DCT. We also evaluate JICE's robustness against packet losses under the following settings: i) JICE with the original Ψ_D; ii) JICE with the regularized Ψ_D; iii) pipeline approach with the regularized Ψ_D. Fig. 22 shows the recovery distortion rates versus the packet delivery ratio under different approaches. With higher delivery rates, pipeline approach achieves the best recovery quality. This is because either the fundamental wave or the harmonics can be well represented by a few coefficients. However, JICE with the regularized DCT is the most robust to packet losses. In particular, the pipeline approach performs poorly (about 60% distortion) when 30% packets are lost (i.e., 70% delivery ratio). This result is consistent with our previous result for the power signals (cf. Section 7.2.1).

8.2. JICE for Other Applications

Many pervasive sensing and Internet of things (IoT) applications based on capability-limited sensors face the same challenges in ensuring data fidelity and secrecy as WEANs. Examples include residential activity sensing and wireless medical monitoring. Sensors for residential activity monitoring often stream their high-resolution data to a central node for complex cognitive processing [Phillips et al. 2013]. However, the data can reveal the user's daily activities, from usage of appliances [Srinivasan et al. 2008], to keystroke sequence on a computer keyboard [Zhuang et al. 2009] if acoustic sensors are used, as in [Phillips et al. 2013]. Mote-class body-worn sensors have been used for monitoring patients' vital signs (e.g., pulse) at low rates in general hospital units [Chipara et al. 2010]. To support cardiac and epilepsy care that requires high-rate (up to $100\,\mathrm{Hz}$) electroencephalography (EEG), electrocardiogram (ECG), and/or acceleration measurements, compressing and encrypting this privacy-sensitive data become imperative. In [Wang et al. 2010], body-worn sensors apply the pipeline approach to compress and encrypt ECG data before wireless transmission. JICE can be applied to these other emerging applications. As illustrated in the voltage monitoring application in Section 8.1, specific best choice of key CS elements such as representation basis and measurement matrix may be application specific. However, several generic design elements advanced in this paper, such as the adaptive representation basis configuration and the perturbation approach for privacy preservation, can be readily applied to new application domains.

9. CONCLUSION

This paper applied CS to jointly compress and encrypt measurements from wireless power meters in a WEAN. We designed JICE through analysis and extensive empirical studies based on real data traces. We developed a lightweight algorithm to reconfigure the representation basis of JICE adaptively at run time to optimize performance. For privacy, we identified leak of statistical information by CS and proposed a perturbation approach to solving the vulnerability. Besides, we considered a threat in which the adversary can obtain a noisy version of the original signal and thus can estimate the random measurement matrix that is used as a secret key in JICE. Our analysis suggests the update rate for the random measurement matrix before it can be accurately estimated by the adversary. Extensive benchmarking and testbed experiments showed that JICE outperforms various baseline approaches under different realistic settings. This paper also presented an extension to address voltage monitoring and discussed other application scenarios.

REFERENCES

S. Agrawal and S. Vishwanath. 2011. Secrecy using compressive sensing. In *IEEE Info. Theory Workshop*.

W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak. 2006. Compressive wireless sensing. In *IPSN*. ACM, 134–142.

TL Baldwin, L Mili, MB Boisen Jr, and R Adapa. 1993. Power system observability with minimal phasor measurement placement. *IEEE Transactions on Power Systems* 8, 2 (1993), 707–715.

BBC. 2013. Smart meter project is delayed. (2013). http://www.bbc.com/news/business-22480068.

R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss. 2008. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Allerton*.

E. Candès and J. Romberg. 2007. Sparsity and incoherence in compressive sampling. *Inverse Problems* 23, 3 (2007), 969.

E. J. Candès and others. 2006. Compressive sampling. In *Intl. Congress of Mathematicians*, Vol. 3.

E. J. Candès and M. B. Wakin. 2008. An introduction to compressive sampling. *IEEE Signal Processing Magazine* 25, 2 (2008), 21–30.

E. J. Candès, M. B. Wakin, and S. P. Boyd. 2008. Enhancing sparsity by reweighted l_1-minimization. *Journal of Fourier analysis and applications* 14, 5-6 (2008), 877–905.

Octav Chipara, Chenyang Lu, Thomas C Bailey, and Gruia-Catalin Roman. 2010. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *SenSys*. ACM, 155–168.

S.-Y. Chiu and H. H. Nguyen. 2015. (2015). https://github.com/bigbitesaint/compress-sensing-nesc.

Shane S Clark, Hossen Mustafa, Benjamin Ransford, Jacob Sorber, Kevin Fu, and Wenyuan Xu. 2013. Current events: Identifying webpages by tapping the electrical outlet. In *ESORICS*. Springer, 700–717.

R. Cox, SB Leeb, SR Shaw, and LK Norford. 2006. Transient event detection for nonintrusive load monitoring and demand side management using voltage distortion. In *Appl. Power Electron. Conf. & Expo.*

Edvard Csanyi. 2016. Essential Basics of Total Harmonic Distortion. (2016). http://electrical-engineering-portal.com/essential-basics-of-total-harmonic-distortion-thd.

Philip L. David M. 2007. Packet Link Layer. (2007). http://www.tinyos.net/tinyos-2.x/doc/html/tep127.html.

Stephen Dawson-Haggerty, Steven Lanzisera, Jay Taneja, Richard Brown, and David Culler. 2012. @scale: insights from a large, long-lived appliance energy WSN. In *IPSN*.

Eaton. 2006. Next-Generation Power Quality Meters. (2006). http://bit.ly/1B9QOst.

Miro Enev, Sidhant Gupta, Tadayoshi Kohno, and Shwetak N Patel. 2011. Televisions, video privacy, and powerline electromagnetic interference. In *CCS*. ACM, 537–550.

Matthias Grawinkel, Michael Mardaus, Tim Suess, and André Brinkmann. 2015. Evaluation of a hash-compress-encrypt pipeline for storage system applications. In *Intl. Conf. Networking, Architecture and Storage (NAS)*. IEEE, 355–356.

Mareca Hatler, Darryl Gurganious, and Charlie Chi. 2013. *Smart Building Wireless Sensor Networks: A Market Dynamics Report*. Technical Report. ON World Inc.

Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. 2009a. Design and implementation of a high-fidelity ac metering network. In *IPSN*. IEEE, 253–264.

Xiaofan Jiang, Minh Van Ly, Jay Taneja, Prabal Dutta, and David Culler. 2009b. Experiences with a high-fidelity wireless building energy auditing network. In *SenSys*. ACM, 113–126.

Jeff St. John. 2014. Cisco Adds Distribution Automation to Its Grid Network. (2014). http://www.greentechmedia.com/articles/read/cisco-adds-distribution-automation-to-its-grid-network.

N. Lawson. 2010. Reverse-engineering a smart meter. (2010). http://bit.ly/1Oc5O4y.

An Liu and Peng Ning. 2008. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In *IPSN*. IEEE, 245–256.

Yao Liu, Peng Ning, and Michael K Reiter. 2011. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security* 14, 1 (2011), 13.

Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen. 2009. Compressive data gathering for large-scale wireless sensor networks. In *MobiCom*. ACM, 145–156.

MathWorks. 2010. Haar wavelet transform matrix. (2010). http://bit.ly/171YHXx.

MathWorks. 2015. DCT matrix. (2015). http://www.mathworks.com/help/images/ref/dctmtx.html.

John P McGregor and Ruby B Lee. 2000. Performance impact of data compression on virtual private network transactions. In *LCN*. IEEE, 500–510.

Prasant Misra, Wen Hu, Mingrui Yang, and Sanjay Jha. 2012. Efficient cross-correlation via sparse representation in sensor networks. In *IPSN*. ACM, 13–24.

Adem Orsdemir, H Oktay Altun, Gaurav Sharma, and Mark F Bocko. 2008. On the security and robustness of encryption via compressed sensing. In *International Conference for Military Communications*.

Shih-Ching Ou, Hung-Yuan Chung, and Wen-Tsai Sung. 2006. Improving the compression and encryption of images using FPGA-based cryptosystems. *Multimedia Tools and Applications* 28, 1 (2006), 5–22.

Sylvain Pelissier. 2010. Cryptography algorithms for TinyOS. (2010). http://bit.ly/1Vs0cYb.

D. E. Phillips, R. Tan, M.-M. Moazzami, G. Xing, J. Chen, and D. K. Y. Yau. 2013. Supero: A sensor system for unsupervised residential power usage monitoring. In *PerCom*. IEEE, 66–75.

QinetiQ. 2014. LineWatch. (2014). https://www.qinetiq-na.com/products/pscs/linewatchl/.

Yaron Rachlin and Dror Baron. 2008. The secrecy of compressed sensing measurements. In *Allerton*.

Ishtiaq Rouf, Hossen Mustafa, Miao Xu, Wenyuan Xu, Rob Miller, and Marco Gruteser. 2012. Neighborhood watch: security and privacy analysis of automatic meter reading systems. In *CCS*.

Christopher M Sadler and Margaret Martonosi. 2006. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *SenSys*. ACM, 265–278.

Surya Santoso, Edward J Powers, W Mack Grady, and Peter Hofmann. 1996. Power quality assessment via wavelet transform analysis. *IEEE Transactions on Power Delivery* 11, 2 (1996), 924–930.

Yiran Shen, Wen Hu, Junbin Liu, Mingrui Yang, Bo Wei, and Chun Tung Chou. 2012. Efficient background subtraction for real-time tracking in embedded camera networks. In *SenSys*. ACM, 295–308.

Sonnonet. 2011. (2011). http://www.sonnonet.com.

Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. 2008. Protecting your daily in-home activity information from a wireless snooping attack. In *UbiComp*. ACM, 202–211.

D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk. 2006. A new compressive imaging camera architecture using optical-domain compression. In *IS&T/SPIE Electronic Imaging*.

TinyOS. 2010. Security and Cryptography in TinyOS. (2010). http://stanford.io/1iQqpxN.

C. Wang, B. Zhang, K. Ren, and J. M. Roveda. 2013. Privacy-assured outsourcing of image reconstruction service in cloud. *IEEE Transactions on Emerging Topics in Computing* 1, 1 (2013), 166–177.

Honggang Wang, Dongming Peng, Wei Wang, Hamid Sharif, Hsiao-Hwa Chen, and Ali Khoynezhad. 2010. Resource-aware secure ECG healthcare monitoring through body sensor networks. *IEEE Wireless Communications* 17, 1 (2010), 12–19.

Ami Wiesel, Yonina C Eldar, and Arie Yeredor. 2008. Linear regression with Gaussian model uncertainty: Algorithms and bounds. *IEEE Transactions on Signal Processing* 56, 6 (2008), 2194–2205.

Xiaopei Wu and Mingyan Liu. 2012. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In *IPSN*. ACM, 1–12.

Li Zhuang, Feng Zhou, and J Doug Tygar. 2009. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security* 13, 1 (2009), 3.

A. PROOF OF PROPOSITION 4.1

PROOF. We can verify $\hat{\mathbf{x}}_{(k)} = \hat{\mathbf{u}}_{(k)} + \hat{\mathbf{w}}_{(k)}$ and $\mathbf{x}_{(k)} = \mathbf{u}_{(k)} + \mathbf{w}_{(k)}$. As $\|\mathbf{x}_{(k)} - \mathbf{x}\|_{\ell_2} = \|\hat{\mathbf{x}}_{(k)} - \hat{\mathbf{x}}\|_{\ell_2} = \|(\hat{\mathbf{u}}_{(k)} - \hat{\mathbf{u}}) + (\hat{\mathbf{w}}_{(k)} - \hat{\mathbf{w}})\|_{\ell_2}$, we have

$$\lim_{N\to\infty} \epsilon^2(\mathbf{x}_{(k)}, \mathbf{x}) = \lim_{N\to\infty} \frac{\frac{1}{N}\|(\hat{\mathbf{u}}_{(k)} - \hat{\mathbf{u}}) + (\hat{\mathbf{w}}_{(k)} - \hat{\mathbf{w}})\|_{\ell_2}^2}{\frac{1}{N}\|\mathbf{u} + \mathbf{w}\|_{\ell_2}^2} \overset{(*)}{=} \lim_{N\to\infty} \frac{\frac{1}{N}\|\hat{\mathbf{u}}_{(k)} - \hat{\mathbf{u}}\|_{\ell_2}^2 + \frac{1}{N}\|\hat{\mathbf{w}}_{(k)} - \hat{\mathbf{w}}\|_{\ell_2}^2}{\frac{1}{N}\|\mathbf{u}\|_{\ell_2}^2 + \frac{1}{N}\|\mathbf{w}\|_{\ell_2}^2}$$

$$= \frac{\lim_{N\to\infty} \frac{1}{N}\|\hat{\mathbf{u}}_{(k)} - \hat{\mathbf{u}}\|_{\ell_2}^2 + \lim_{N\to\infty} \frac{1}{N}\|\hat{\mathbf{w}}_{(k)} - \hat{\mathbf{w}}\|_{\ell_2}^2}{\sigma_{\mathbf{w}}^2(1 + \mathrm{SNR})} \tag{7}$$

where the step marked by (*) follows from $\lim_{N\to\infty} \frac{1}{N}\langle \mathbf{u}, \mathbf{w}\rangle = 0$ and $\lim_{N\to\infty} \frac{1}{N}\langle \hat{\mathbf{u}}_{(k)} - \hat{\mathbf{u}}, \hat{\mathbf{w}}_{(k)} - \hat{\mathbf{w}}\rangle = 0$. Moreover, we have

$$\begin{aligned}
&\lim_{N\to\infty} \frac{1}{N}\|\hat{\mathbf{w}}_{(k)} - \hat{\mathbf{w}}\|_{\ell_2}^2 = \mathbb{E}[(\hat{w}_{(k)} - \hat{w})^2] \\
&= \mathbb{E}[(\hat{w}_{(k)} - \hat{w})^2 | \hat{w}_{(k)} = 0] \cdot \mathbb{P}(\hat{w}_{(k)} = 0) + E[(\hat{w}_{(k)} - \hat{w})^2 | \hat{w}_{(k)} = \hat{w}] \cdot \mathbb{P}(\hat{w}_{(k)} = \hat{w}) \\
&= \mathbb{E}[\hat{w}^2](1 - \rho) = \sigma_{\mathbf{w}}^2(1 - \rho),
\end{aligned} \tag{8}$$

where $\hat{w}_{(k)} \in \hat{\mathbf{w}}_{(k)}$ and $\hat{w} \in \hat{\mathbf{w}}$. It is easy to verify that $\|\hat{\mathbf{u}}_{(k)} - \hat{\mathbf{u}}\|_{\ell_2}^2 \geq \sum_{i=k+1}^{N}([\hat{\mathbf{u}}_{\mathrm{desc}}]_i)^2$. Therefore, by applying this bound and Eq. (8) on Eq. (7), we have $\lim_{N\to\infty} \epsilon(\mathbf{x}_{(k)}, \mathbf{x}) \geq \sqrt{\frac{\frac{1}{\sigma_{\mathbf{w}}^2} f_{\hat{\mathbf{u}}}(\rho) + (1-\rho)}{1+\mathrm{SNR}}}$, where $f_{\hat{\mathbf{u}}}(\rho) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=\rho N+1}^{N}([\hat{\mathbf{u}}_{\mathrm{desc}}]_i)^2$ and $\hat{\mathbf{u}}_{\mathrm{desc}}$ denotes the sorted $\hat{\mathbf{u}}$ in a descending order. Let $I_A(x)$ denote the indicator function of A where $I_A(x) = 1$ if $x \in A$ and $I_A(x) = 0$ otherwise. It is easy to verify $\sum_{i=\rho N+1}^{N}([\hat{\mathbf{u}}_{\mathrm{desc}}]_i)^2 \geq \hat{\mathbf{u}}_{\min}^2(\rho_{\mathbf{u}} N - \rho N) I_{[0,\rho_{\mathbf{u}}]}(\rho)$. Thus, $f_{\hat{\mathbf{u}}}(\rho) \geq \hat{\mathbf{u}}_{\min}^2(\rho_{\mathbf{u}} - \rho)$ and we have Eq. (4). □