JICE: Joint Data Compression and Encryption for Wireless Energy Auditing Networks

Sheng-Yuan Chiu[1,2], Hoang Hai Nguyen[1], Rui Tan[1], David K.Y. Yau[1,3],Deokwoo Jung[1]

[1]Advanced Digital Science Center, Illinois at Singapore
[2]National Tsing Hua University, Taiwan
[3]Singapore University of Technology and Design, Singapore

Outline

- **Motivation**
- Design of JICE
- Secrecy of JICE
- Experiment

Wireless Energy Auditing

- Buildings account for 40% electricity use
- Wireless appliance submetering

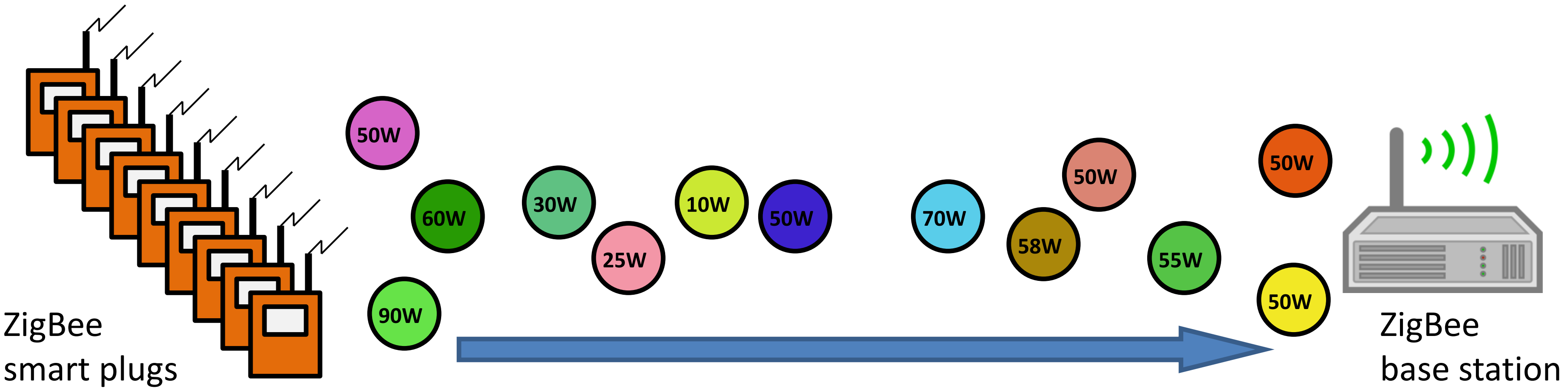
Smart plugs (ZigBee radio)

Wireless Energy Auditing

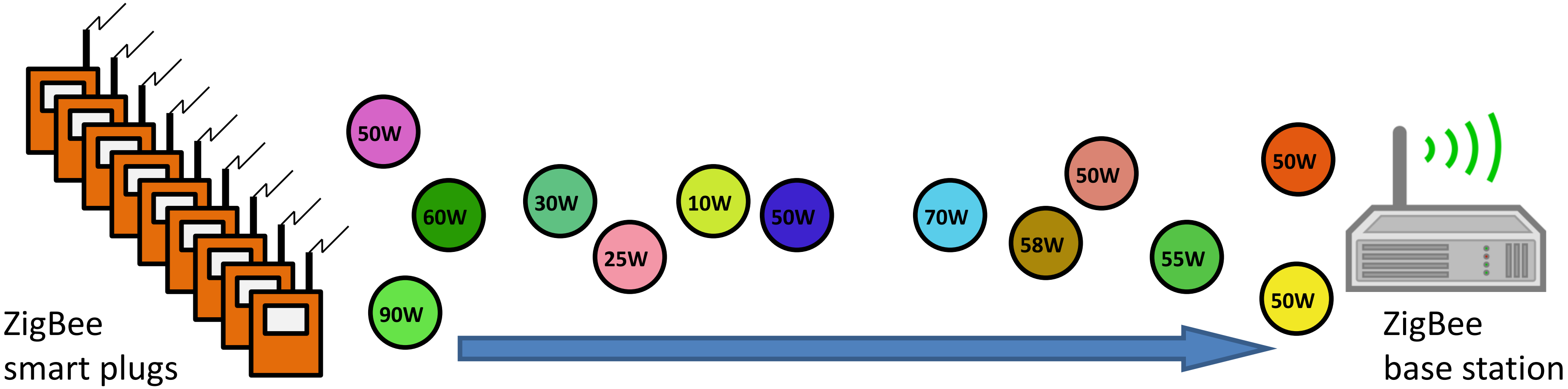
- Buildings account for 40% electricity use
- Wireless appliance submetering
  - Efficiency analysis
    56% energy wasted in our office[Jung 2013]

Smart plugs (ZigBee radio)

Objectives & Challenges

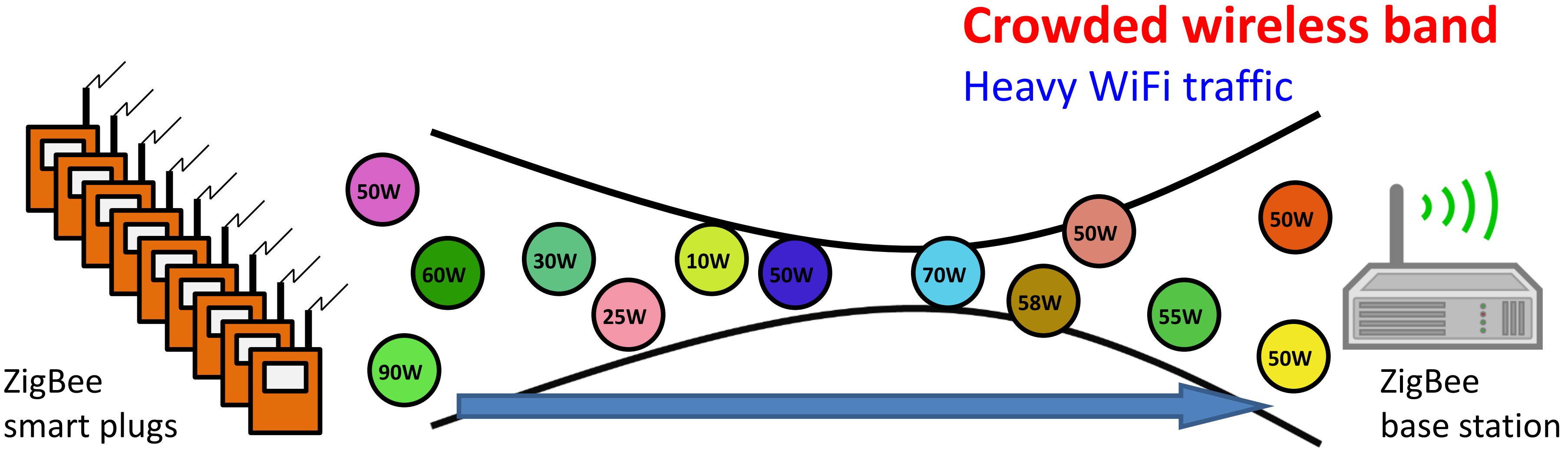


Objectives & Challenges



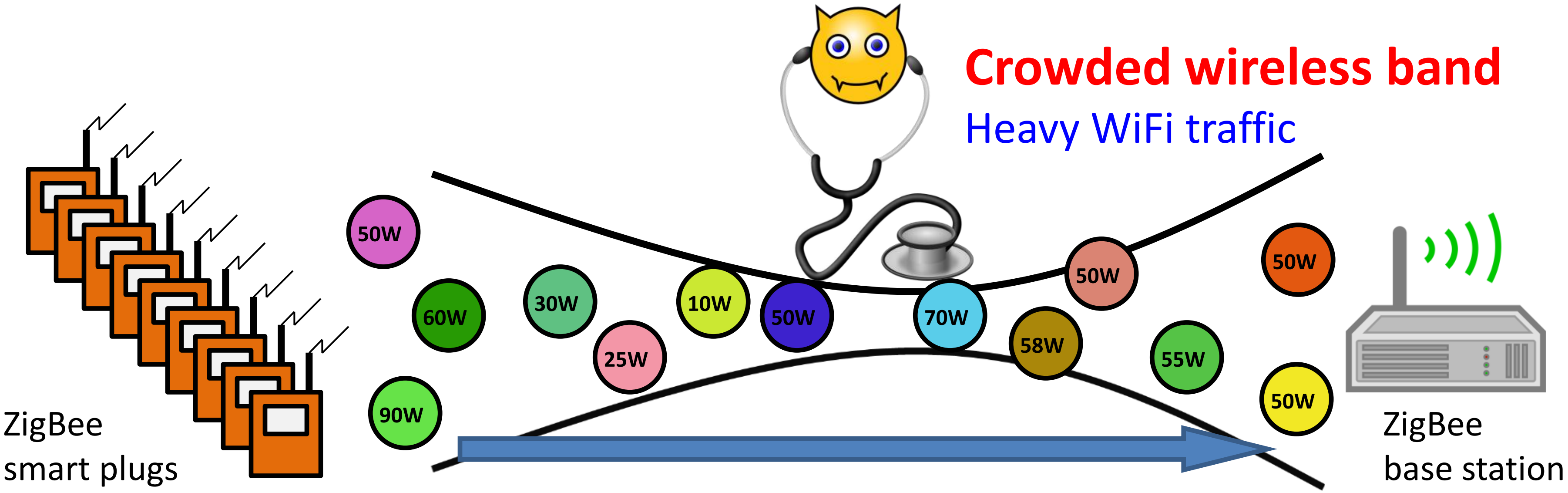
- **Increase coverage (# of meters) and sampling rate**
  - 10% coverage by 455 plugs [Haggerty 2012]
  - Down to 1Hz to support load profiling

Objectives & Challenges



- **Increase coverage (# of meters) and sampling rate**
  - 10% coverage by 455 plugs [Haggerty 2012]
  - Down to 1Hz to support load profiling

Objectives & Challenges



- **Increase coverage (# of meters) and sampling rate**
  - 10% coverage by 455 plugs [Haggerty 2012]
  - Down to 1Hz to support load profiling

- **Data secrecy during wireless communication**
  - Threat model: wireless eavesdropping
  - Reveal TV channel [Enev 2011]

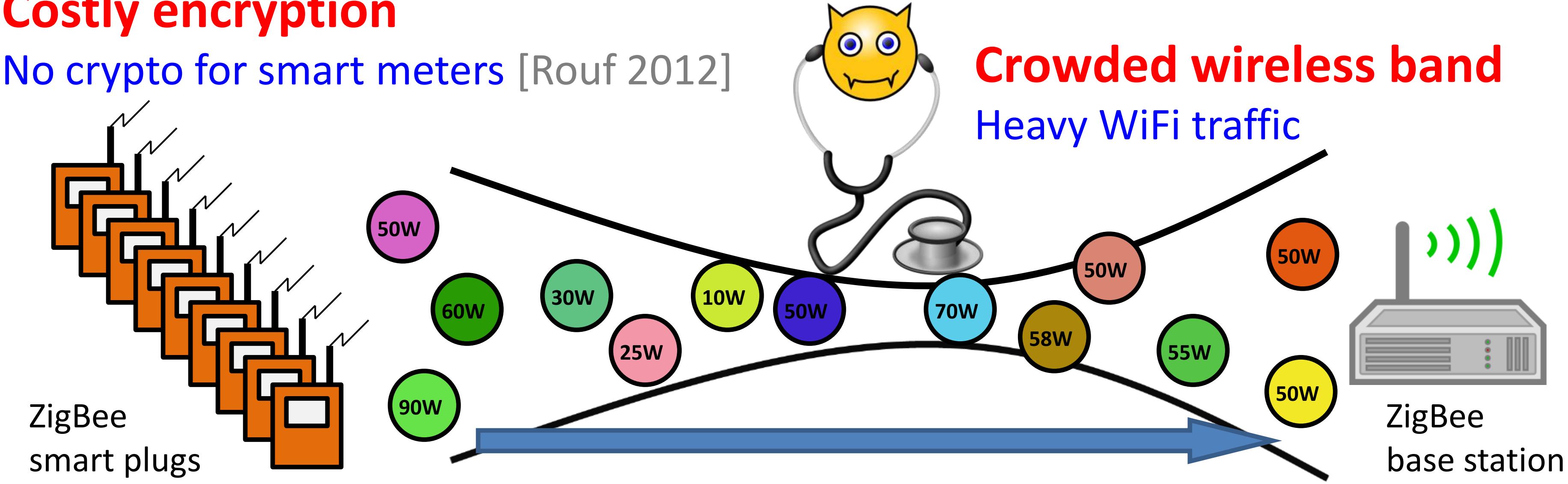
Objectives & Challenges

Costly encryption

No crypto for smart meters [Rouf 2012]

Crowded wireless band

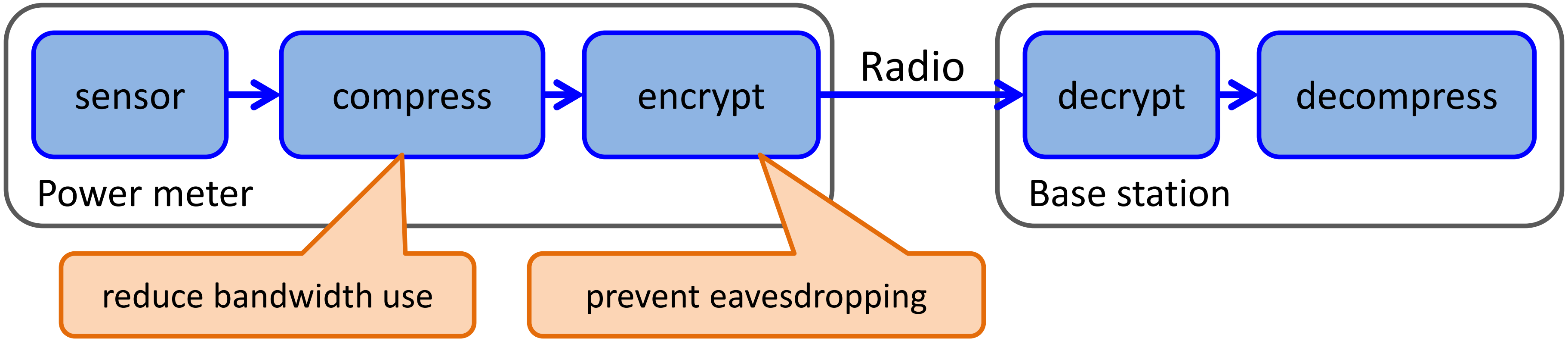
Heavy WiFi traffic



- **Increase coverage (# of meters) and sampling rate**
  - 10% coverage by 455 plugs [Haggerty 2012]
  - Down to 1Hz to support load profiling

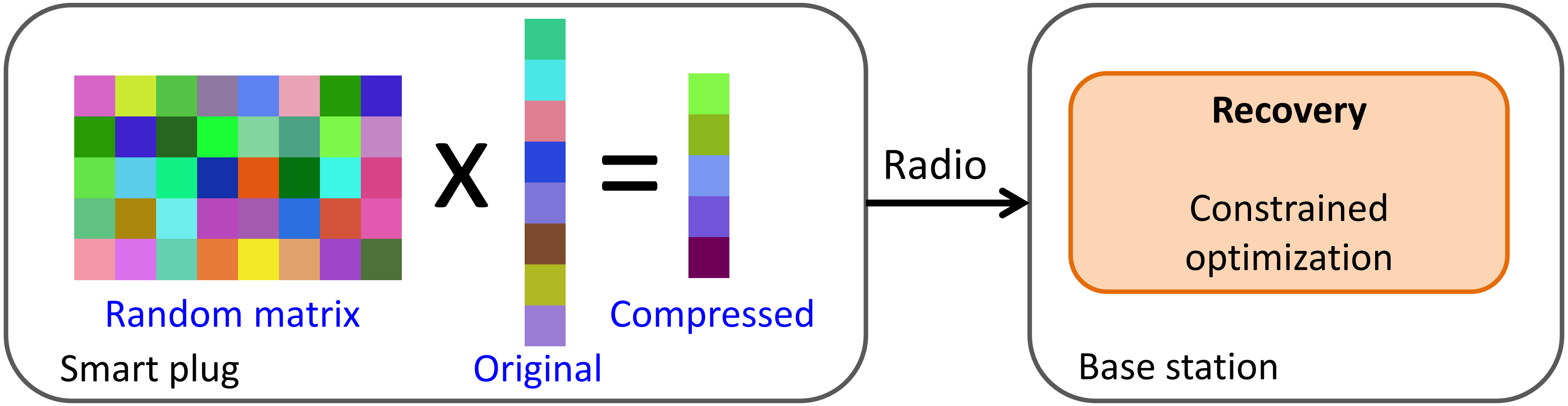
- **Data secrecy during wireless communication**
  - Threat model: wireless eavesdropping
  - Reveal TV channel [Enev 2011]

Conventional Scheme (Pipeline)



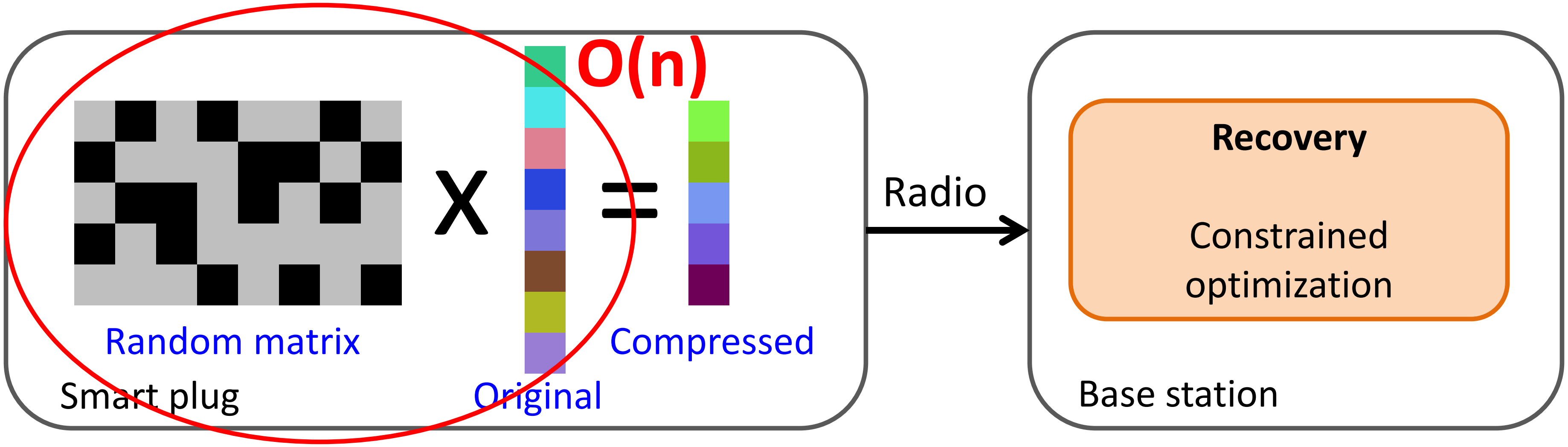
- Inefficient for resource-constrained plugs
  - Computation-intensive compressor and cipher

Compressive Sensing



- Efficient compression
  - Simple matrix multiplication
  - Most computation to recovery side
- Weakly encrypt signal [Rachlin 2008]
  - Shared secret random matrix

Compressive Sensing

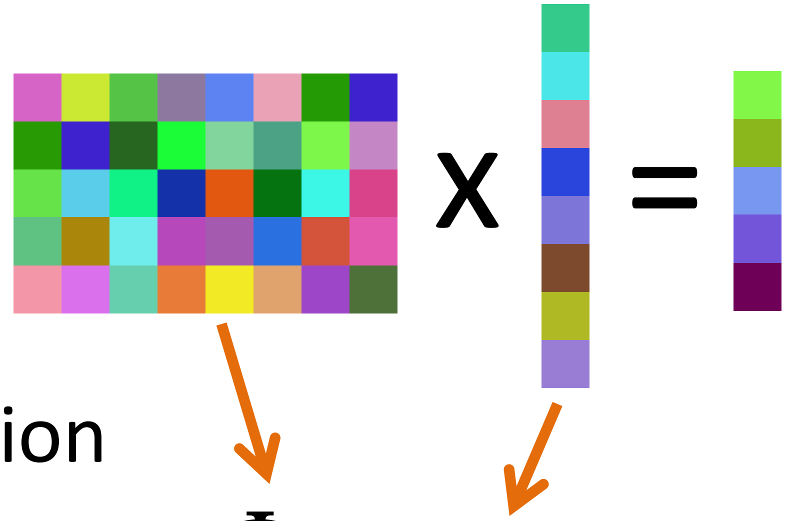


- Efficient compression
  - Simple matrix multiplication
  - Most computation to recovery side
- Weakly encrypt signal [Rachlin 2008]
  - Shared secret random matrix

Outline

- Motivation
- **Design of JICE**
- Secrecy of JICE
- Experiment

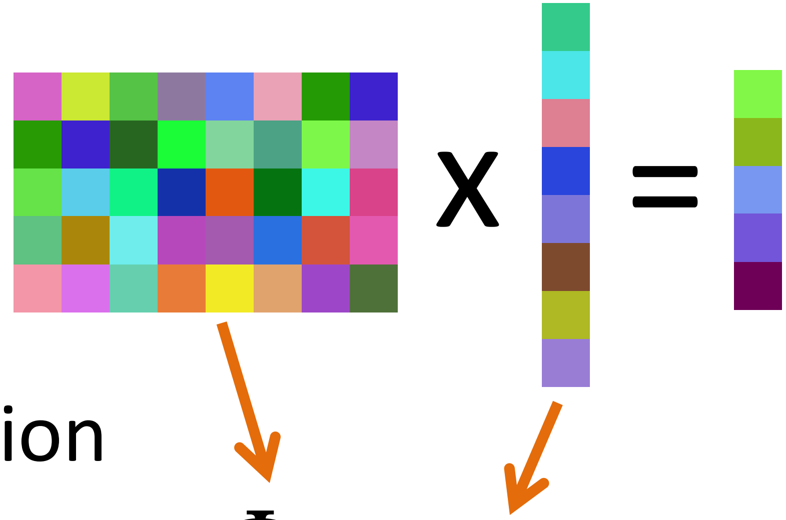
Compressive Sensing Basics



- Compression

$$\boldsymbol{y}_{M\times 1} = \boldsymbol{\Phi}_{M\times N}\,\boldsymbol{x}_{N\times 1} \qquad \boldsymbol{M} < \boldsymbol{N}$$

Compressive Sensing Basics



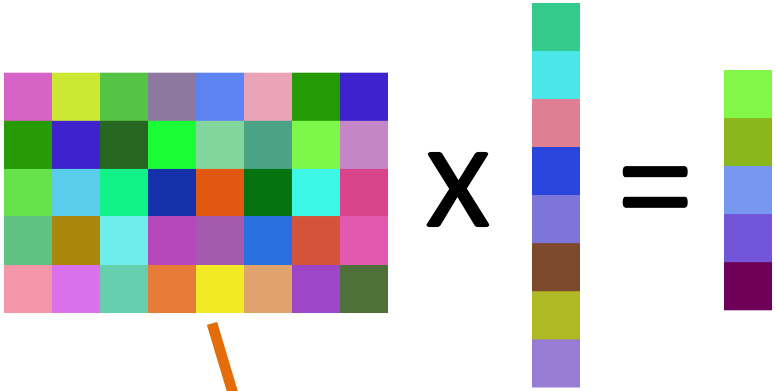
- Compression

$$y_{M\times 1} = \Phi_{M\times N}\, x_{N\times 1} \qquad M < N$$

- Recovery: compute *x* from *y* by

$$x = \Psi_{N\times N} \cdot \arg\min_{z} |z|_1 \quad \text{s.t.} \quad y = \Phi\Psi z$$

Compressive Sensing Basics



- Compression

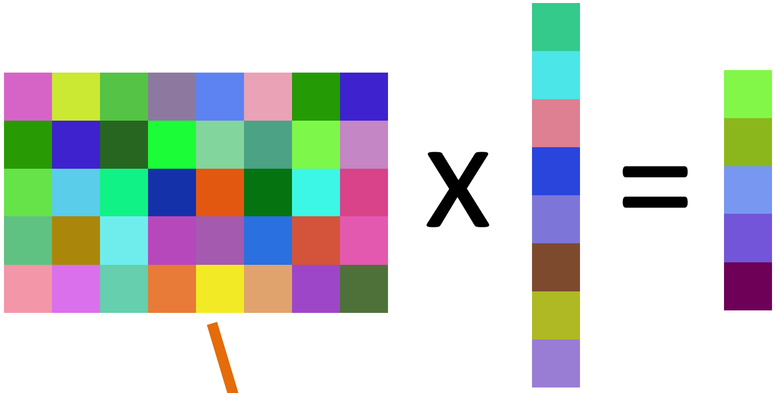
$$y_{M\times 1} = \Phi_{M\times N}\, x_{N\times 1} \qquad M < N$$

- Recovery: compute x from y by

Representation basis (only used for recovery)

$$x = \boxed{\Psi_{N\times N}} \cdot \arg\min_{z} |z|_1 \quad \text{s.t.} \quad y = \Phi\Psi z$$

Compressive Sensing Basics



- Compression

$$\boldsymbol{y}_{M\times 1} = \boldsymbol{\Phi}_{M\times N}\,\boldsymbol{x}_{N\times 1} \qquad M < N$$

- Recovery: compute x from y by

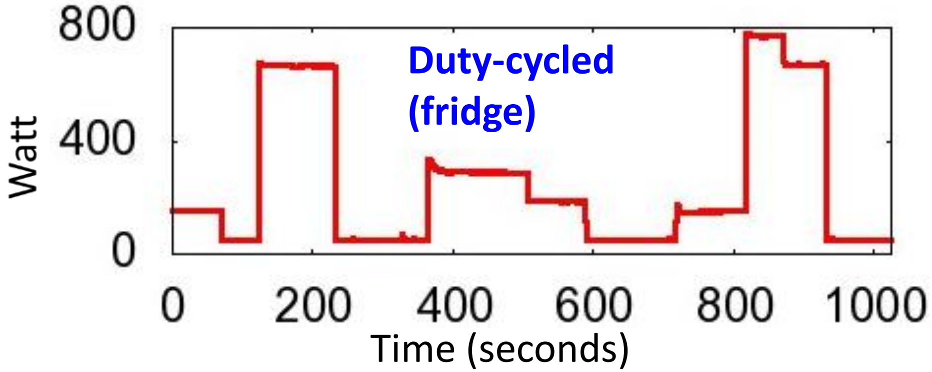
Representation basis (only used for recovery)

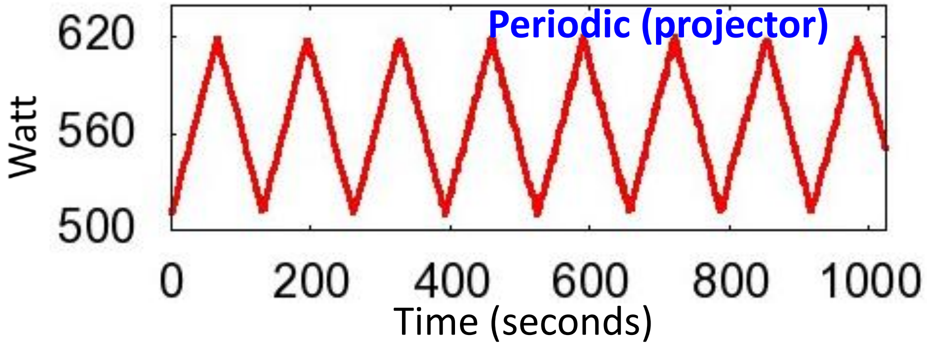
$$x = \boxed{\Psi_{N\times N}} \cdot \arg\min_{z} |z|_1 \quad \text{s.t.} \quad y = \Phi\Psi z$$

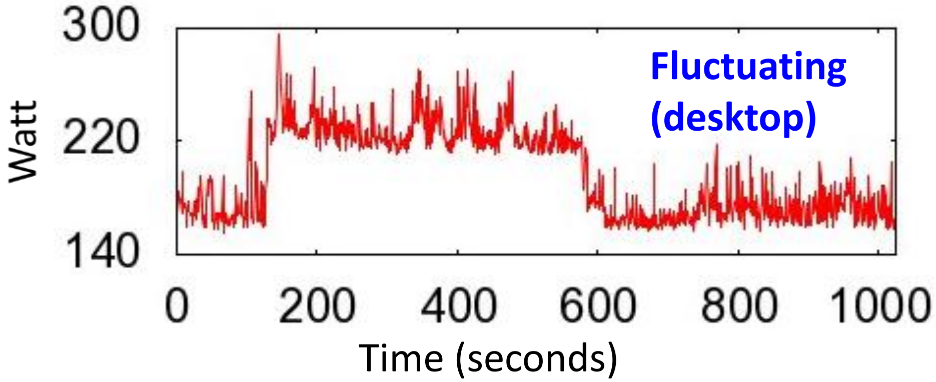
- For better recovery
  - Ψ sparsify *x* ⇨ Ψ^{-1}*x* has many zeros

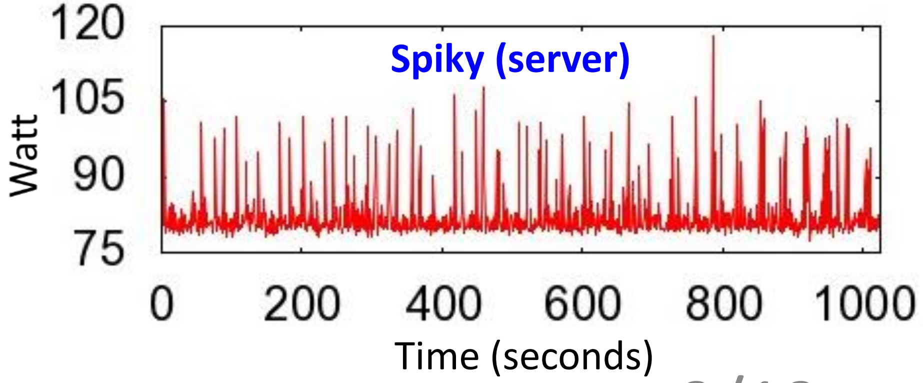
Trace-Driven Design

- Select Φ and Ψ based on traces
  - Data traces from 40 branches for 18 hours
  - Classify power consumption patterns









Random Matrix Φ

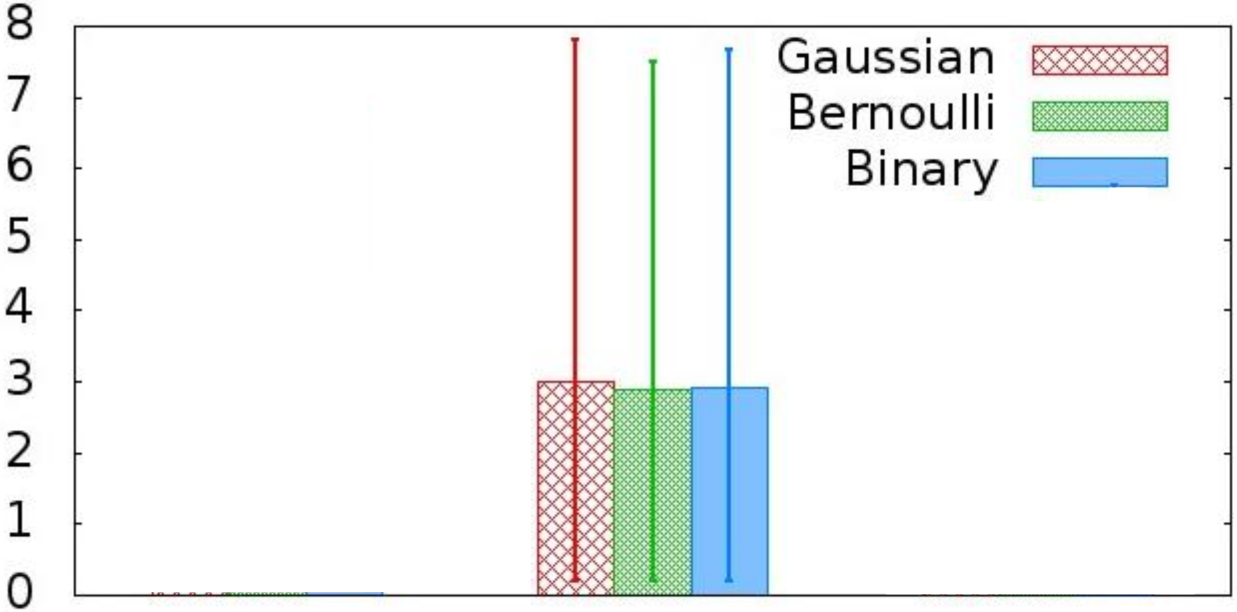
- Gaussian, Bernoulli, Binary

$$\textbf{recovery error} = \frac{\| \tilde{x} - x \|_2}{\| x \|_2}$$

x : **original**

$\tilde{x}$: **recovered**

Recovery error (%)



Representation basis

cosine transform

Random Matrix Φ

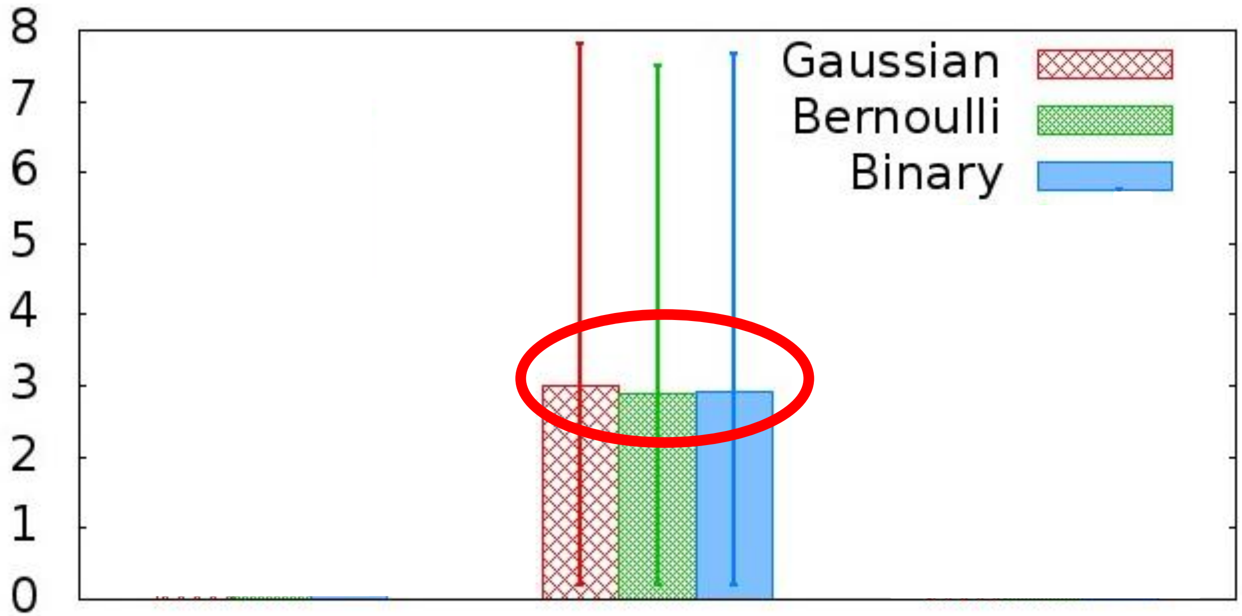
- Gaussian, Bernoulli, Binary

$$\textbf{recovery error} = \frac{\| \tilde{x} - x \|_2}{\| x \|_2}$$

x : **original**

$\tilde{x}$: **recovered**

Recovery error (%)



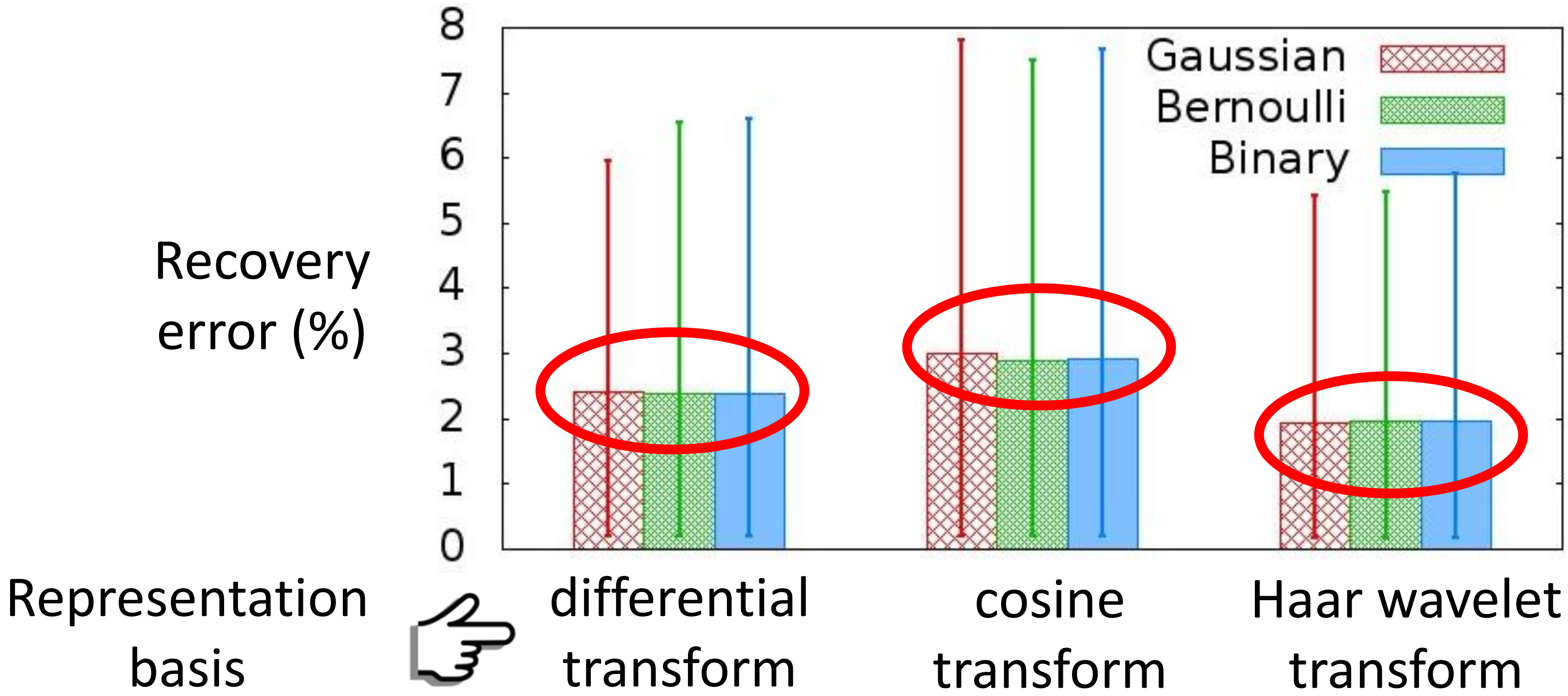
Representation basis

cosine transform

Random Matrix Φ

- Gaussian, Bernoulli, Binary

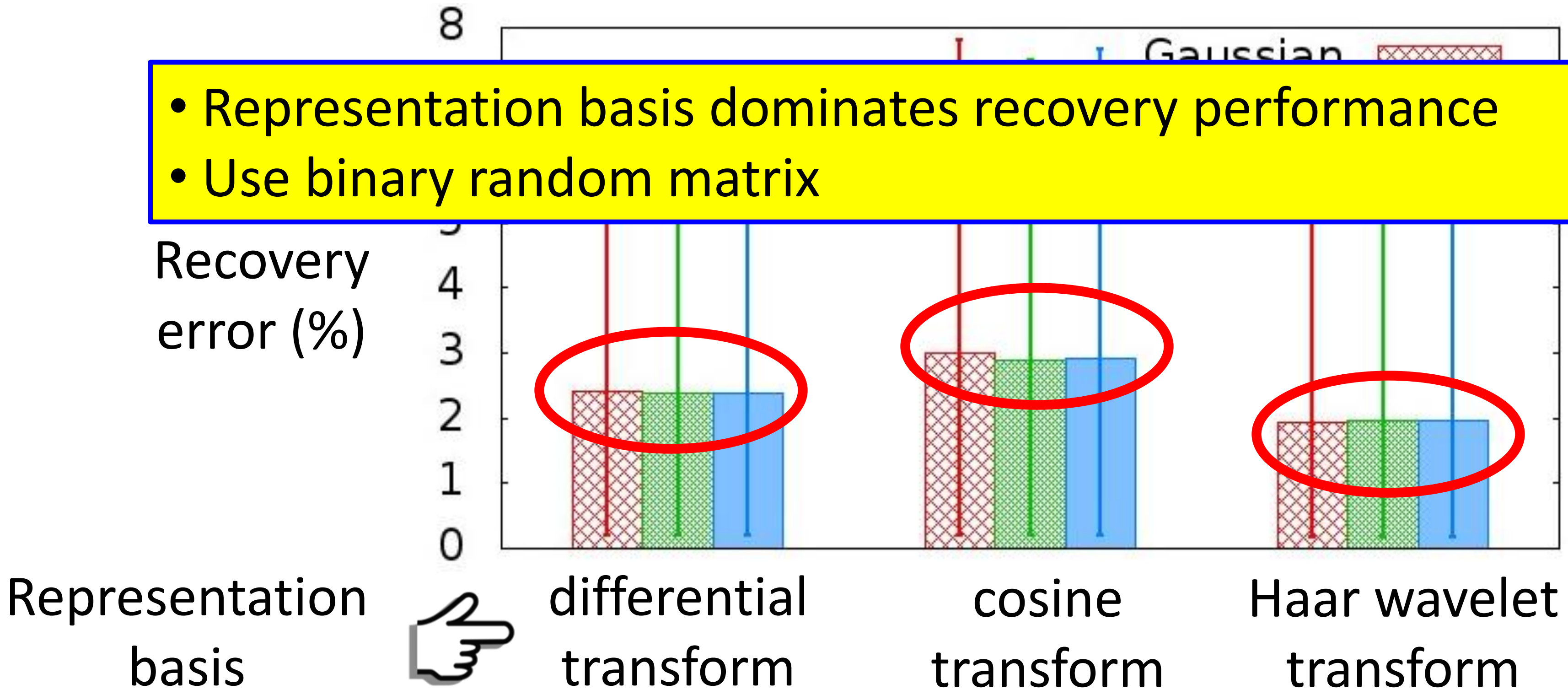
$$\textbf{recovery error} = \frac{\| \tilde{x} - x \|_2}{\| x \|_2} \qquad x : \textbf{original} \qquad \tilde{x} : \textbf{recovered}$$



Random Matrix Φ

- Gaussian, Bernoulli, Binary

$$\textbf{recovery error} = \frac{\| \tilde{x} - x \|_2}{\| x \|_2} \qquad \begin{array}{l} x : \textbf{original} \\ \tilde{x} : \textbf{recovered} \end{array}$$

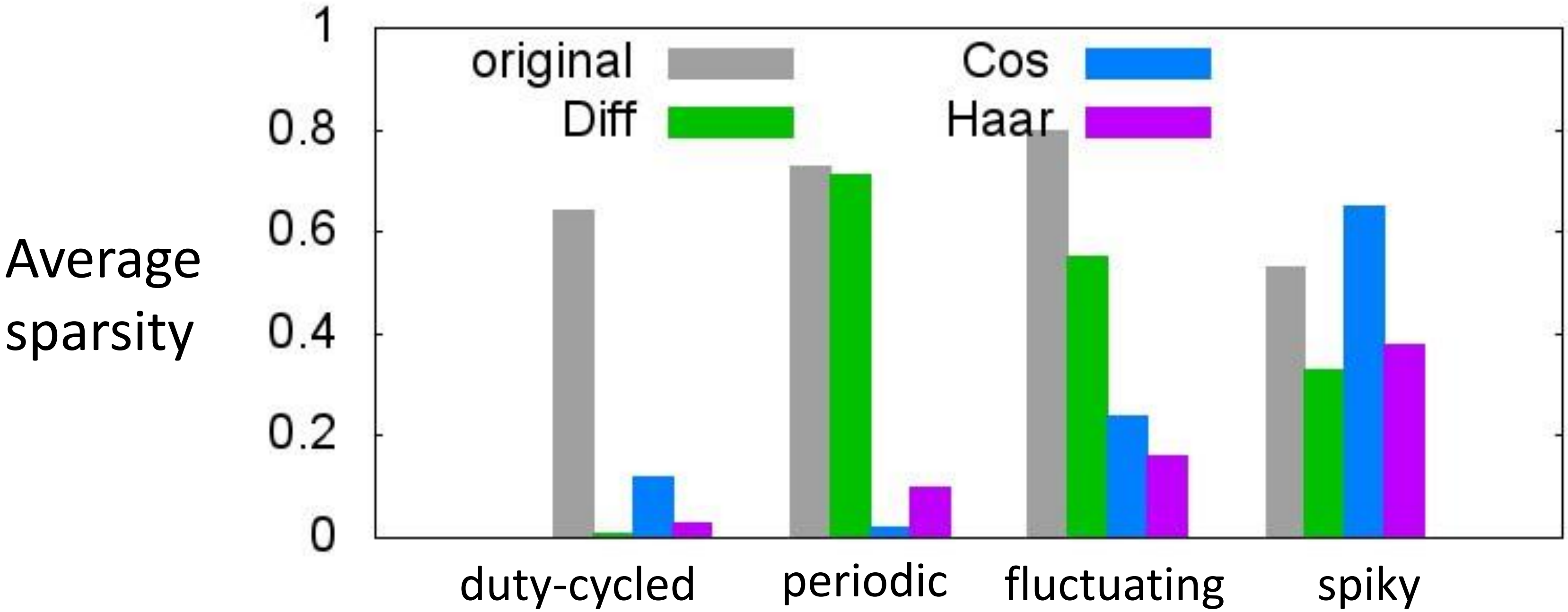


- Representation basis dominates recovery performance
- Use binary random matrix

Representation Basis Ψ

- Differential transform (**Diff**)
- Cosine transform (**Cos**)
- Haar wavelet transform (**Haar**)

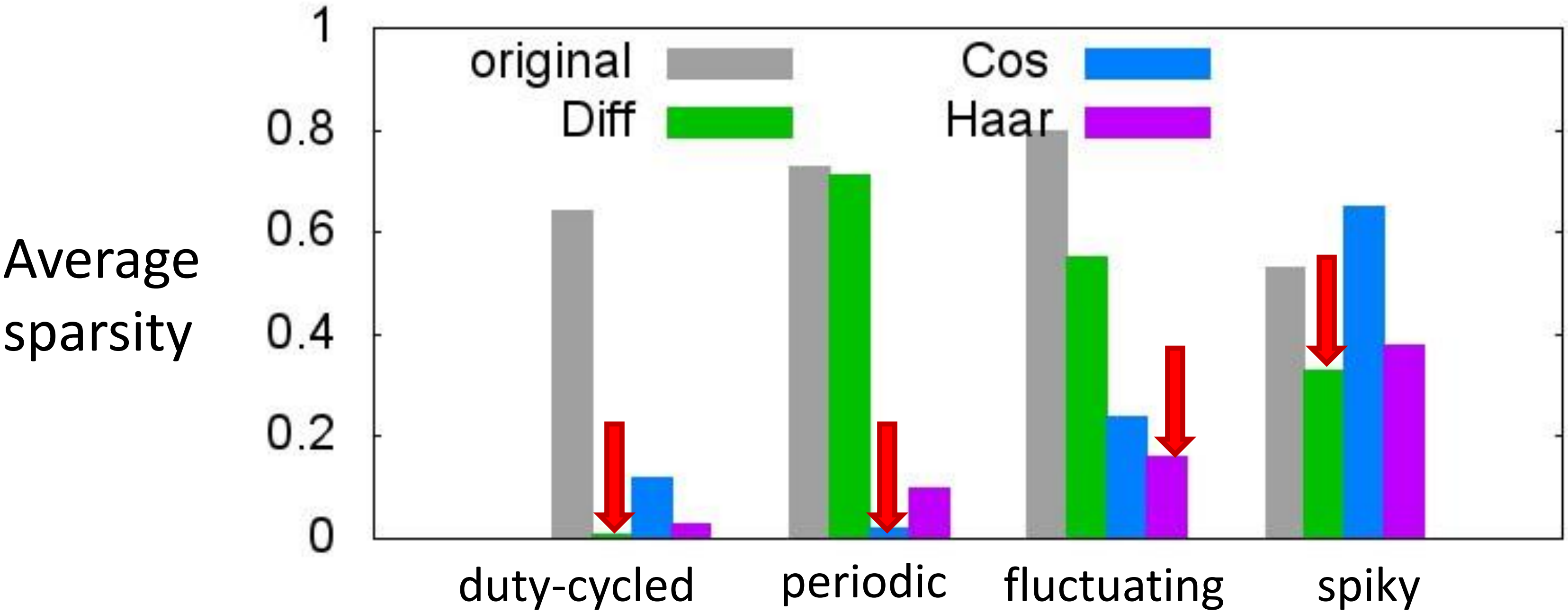
$$\text{sparsity} = \frac{\#\ \text{of nonzeros}}{\text{signal length}}$$



Representation Basis Ψ

- Differential transform (**Diff**)
- Cosine transform (**Cos**)
- Haar wavelet transform (**Haar**)

$$\text{sparsity} = \frac{\#\text{ of nonzeros}}{\text{signal length}}$$



Best choice: Diff Cos Haar Diff

Representation Basis Ψ

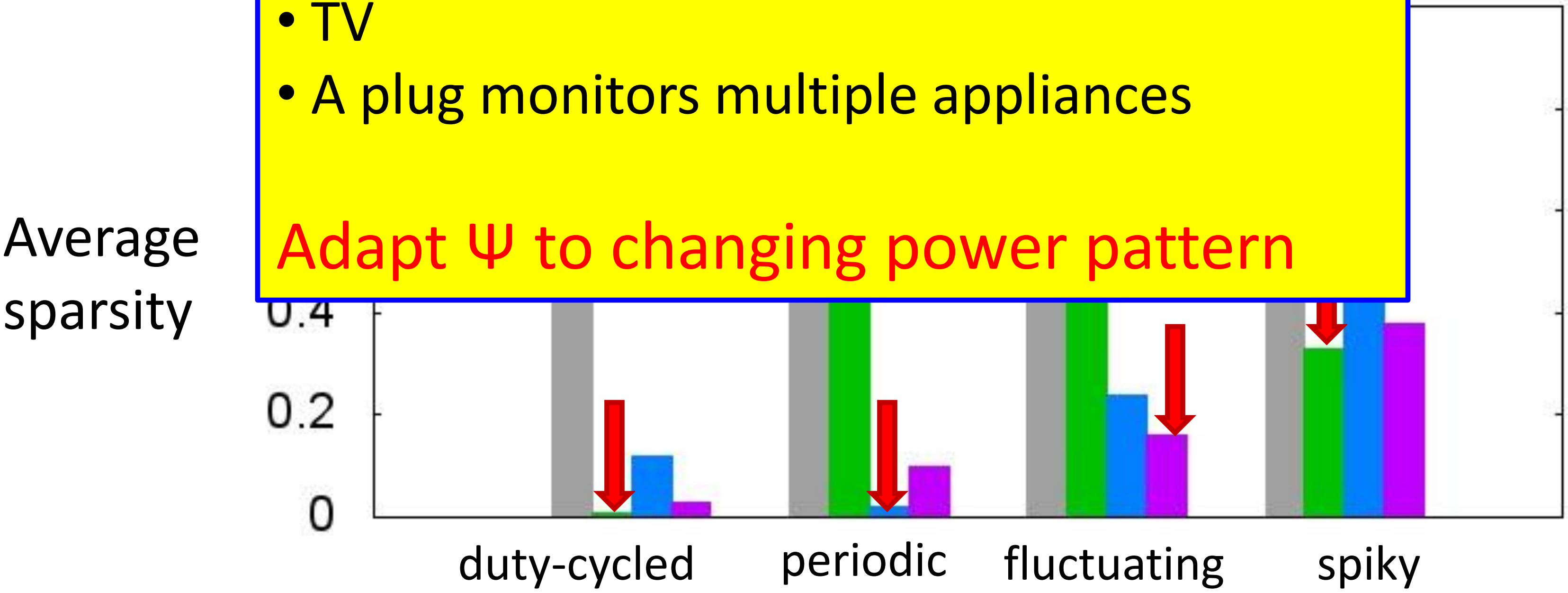
- Differential transform (**Diff**)
- Cosine transform (**Cos**)
- Haar wavelet transform (**Haar**)

$$\text{sparsity} = \frac{\#\ \text{of}\ \ \text{nonzeros}}{\text{signal}\ \ \text{length}}$$

Changing power pattern:

- TV
- A plug monitors multiple appliances

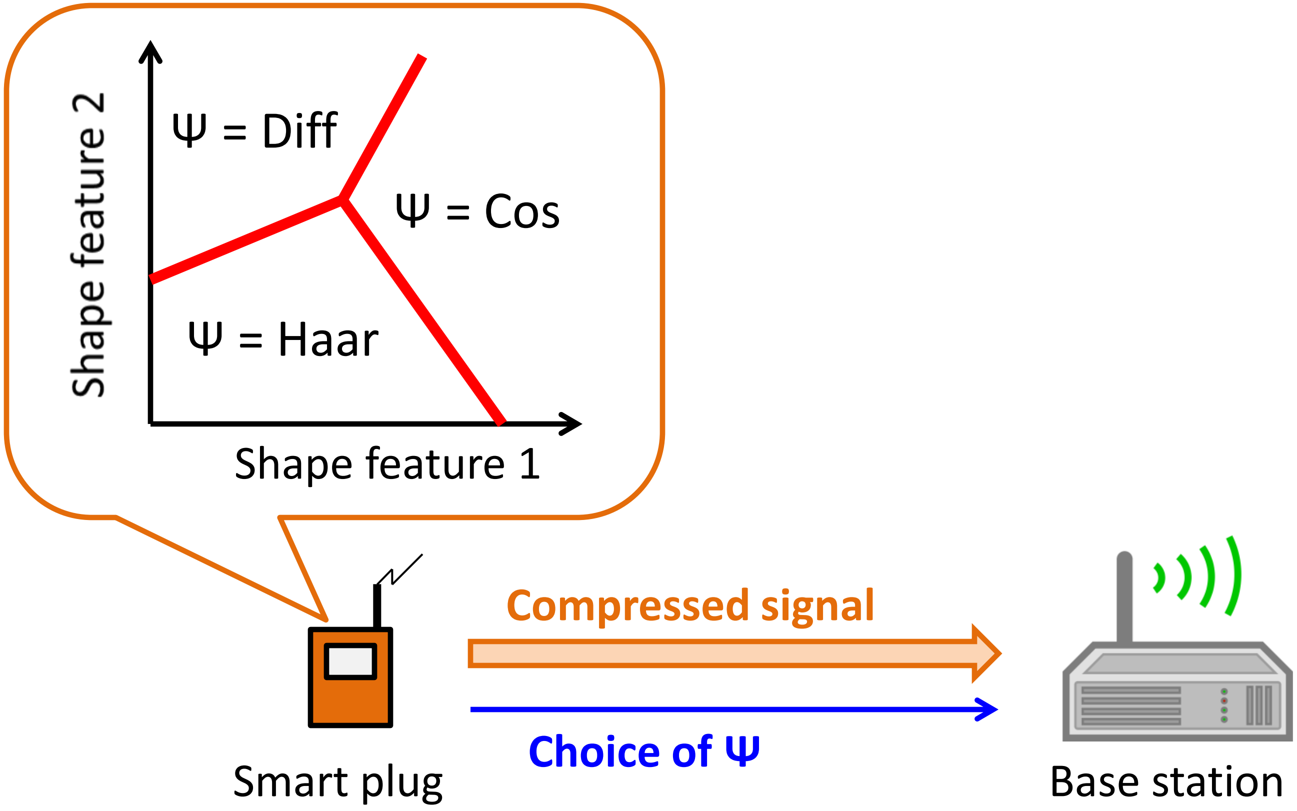
Adapt Ψ to changing power pattern



Best choice: Diff Cos Haar Diff

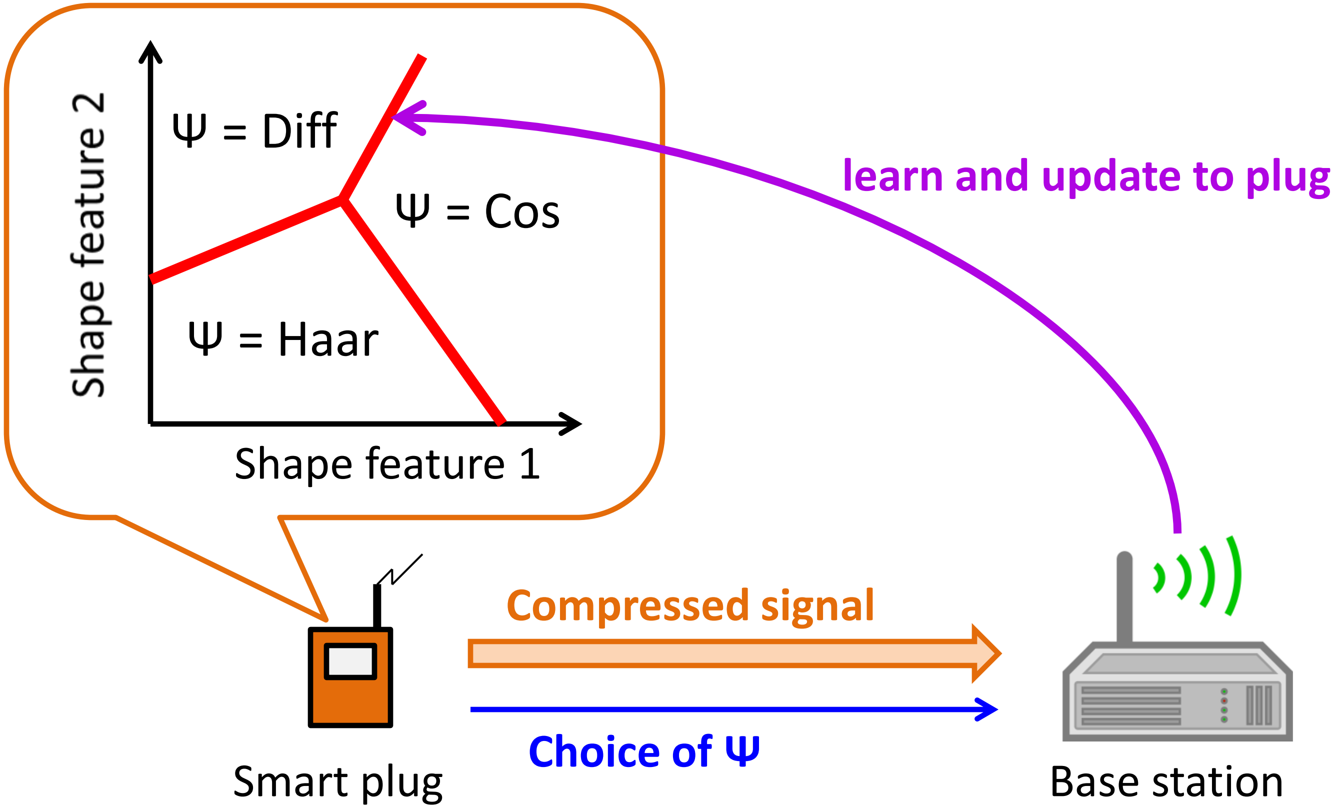
Adaptive Representation Basis

- Machine learning approach
  - Plug selects Ψ based on shape features



Adaptive Representation Basis

- Machine learning approach
  - Plug selects Ψ based on shape features
  - Base station learns decision boundaries



Shape Feature & Decision Table

$$\text{shape feature vector} = \begin{bmatrix} \text{\# of zero crossings} \\ \text{\# of sharp changes} \\ \text{standard deviation} \end{bmatrix}$$

# of zero crossings > Δ_1 ?	N	N	N	N	Y	Y	Y	Y
# of sharp changes > Δ_2 ?	N	N	Y	Y	N	N	Y	Y
Standard deviation > Δ_3 ?	N	Y	N	Y	N	Y	N	Y
Choice of basis	**ADT**	**ADT**	**HWT**	**DCT**	**HWT**	**HWT**	**ADT**	**DCT**

Shape Feature & Decision Table

$$\text{shape feature vector} = \begin{bmatrix} \text{\# of zero crossings} \\ \text{\# of sharp changes} \\ \text{standard deviation} \end{bmatrix}$$

# of zero crossings > Δ_1 ?	N	N	N	N	Y	Y	Y	Y
# of sharp changes > Δ_2 ?	N	N	Y	Y	N	N	Y	Y
Standard deviation > Δ_3 ?	N	Y	N	Y	N	Y	N	Y
Choice of basis	**ADT**	**ADT**	**HWT**	**DCT**	**HWT**	**HWT**	**ADT**	**DCT**

Shape Feature & Decision Table

$$\text{shape feature vector} = \begin{bmatrix} \text{\# of zero crossings} \\ \text{\# of sharp changes} \\ \text{standard deviation} \end{bmatrix}$$

# of zero crossings > Δ_1 ?	N	N	N	N	Y	Y	Y	Y
# of sharp changes > Δ_2 ?	N	N	Y	Y	N	N	Y	Y
Standard deviation > Δ_3 ?	N	Y	N	Y	N	Y	N	Y
Choice of basis	**ADT**	**ADT**	**HWT**	**DCT**	**HWT**	**HWT**	**ADT**	**DCT**

- Trained at base station
  - Minimize recovery error

Outline

- Motivation
- Design of JICE
- **Secrecy of JICE**
- Experiment

Statistics Leak and Perturbation

- Φ is unknown to attacker
  - “Provide a computational guarantee of secrecy” [Rachlin 2008]
- **Leak ℓ_2-norm, mean and variance**

Zero mean: appliance is off
High mean: appliance is on

Statistics Leak and Perturbation

- Φ is unknown to attacker
  - "Provide a computational guarantee of secrecy" [Rachlin 2008]
- **Leak ℓ$_2$-norm, mean and variance**

$$\tilde{x} = x + n$$

$$n = \Psi \cdot [k, 0, 0, \cdots, 0]^T$$

Statistics Leak and Perturbation

- Φ is unknown to attacker
  - "Provide a computational guarantee of secrecy" [Rachlin 2008]
- **Leak ℓ$_2$-norm, mean and variance**

$$\tilde{x} = x + n$$

$$n = \Psi \cdot [k, 0, 0, \cdots, 0]^T$$

Representation basis

Statistics Leak and Perturbation

- Φ is unknown to attacker
  - "Provide a computational guarantee of secrecy" [Rachlin 2008]
- **Leak ℓ$_2$-norm, mean and variance**

$$\tilde{x} = x + n$$

Representation basis

$$n = \Psi \cdot [k, 0, 0, \cdots, 0]^T$$

shared secret

Statistics Leak and Perturbation

- Φ is unknown to attacker
  - "Provide a computational guarantee of secrecy" [Rachlin 2008]
- **Leak ℓ_2-norm, mean and variance**

$$\tilde{x} = x + n$$

Representation basis

$$n = \Psi \cdot [k, 0, 0, \cdots, 0]^T$$

shared secret

- **Statistics depend on *k***

Statistics Leak and Perturbation

- Φ is unknown to attacker
  - "Provide a computational guarantee of secrecy" [Rachlin 2008]
- **Leak ℓ$_2$-norm, mean and variance**

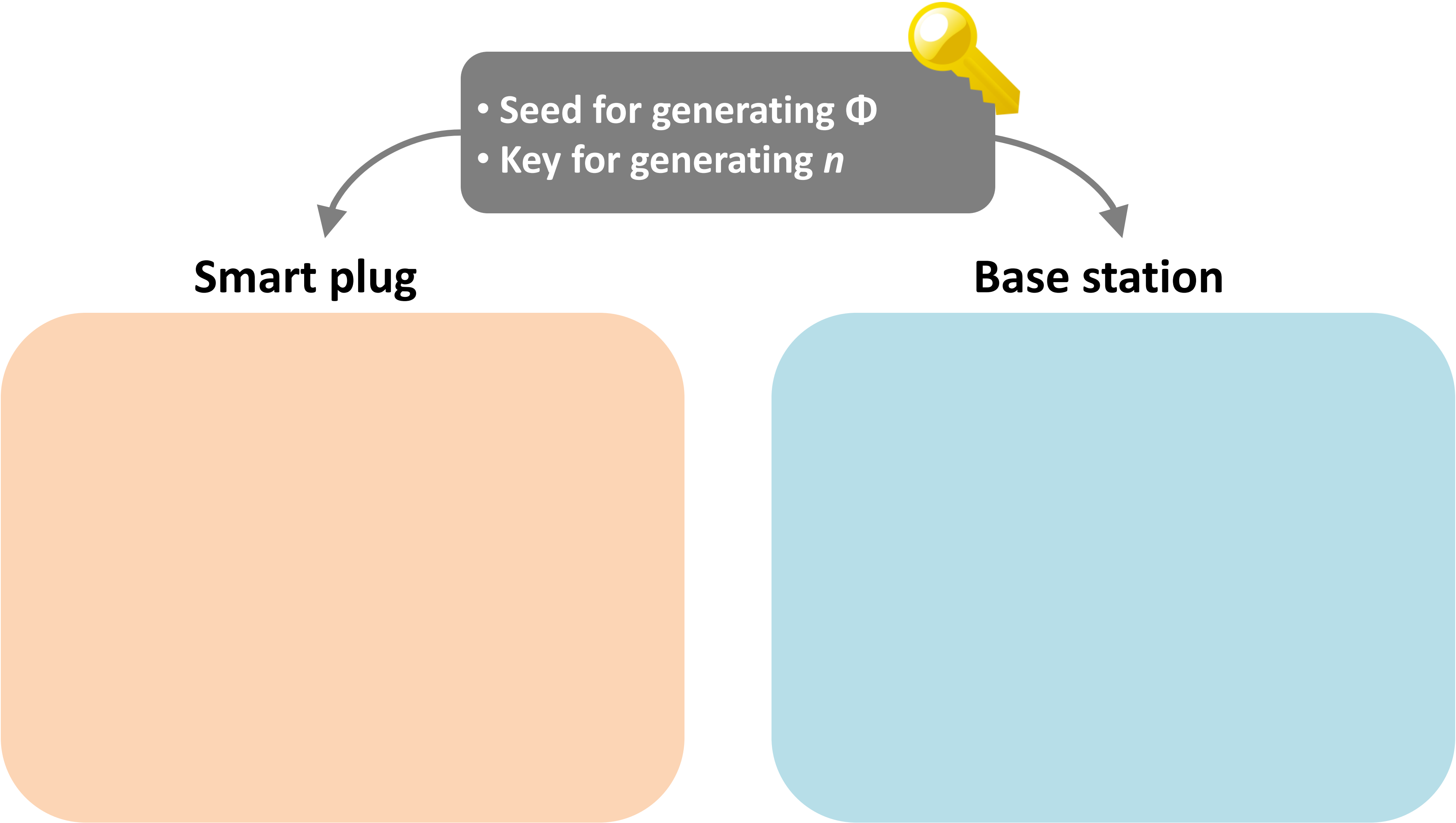
$$\tilde{x} = x + n$$

$$\boxed{n = \Psi \cdot [k, 0, 0, \cdots, 0]^T}$$

Very sparse in transform domain

- ***Statistics depend on k***
- Little (no) change to sparsity
  - Little impact on recovery

Recap of JICE

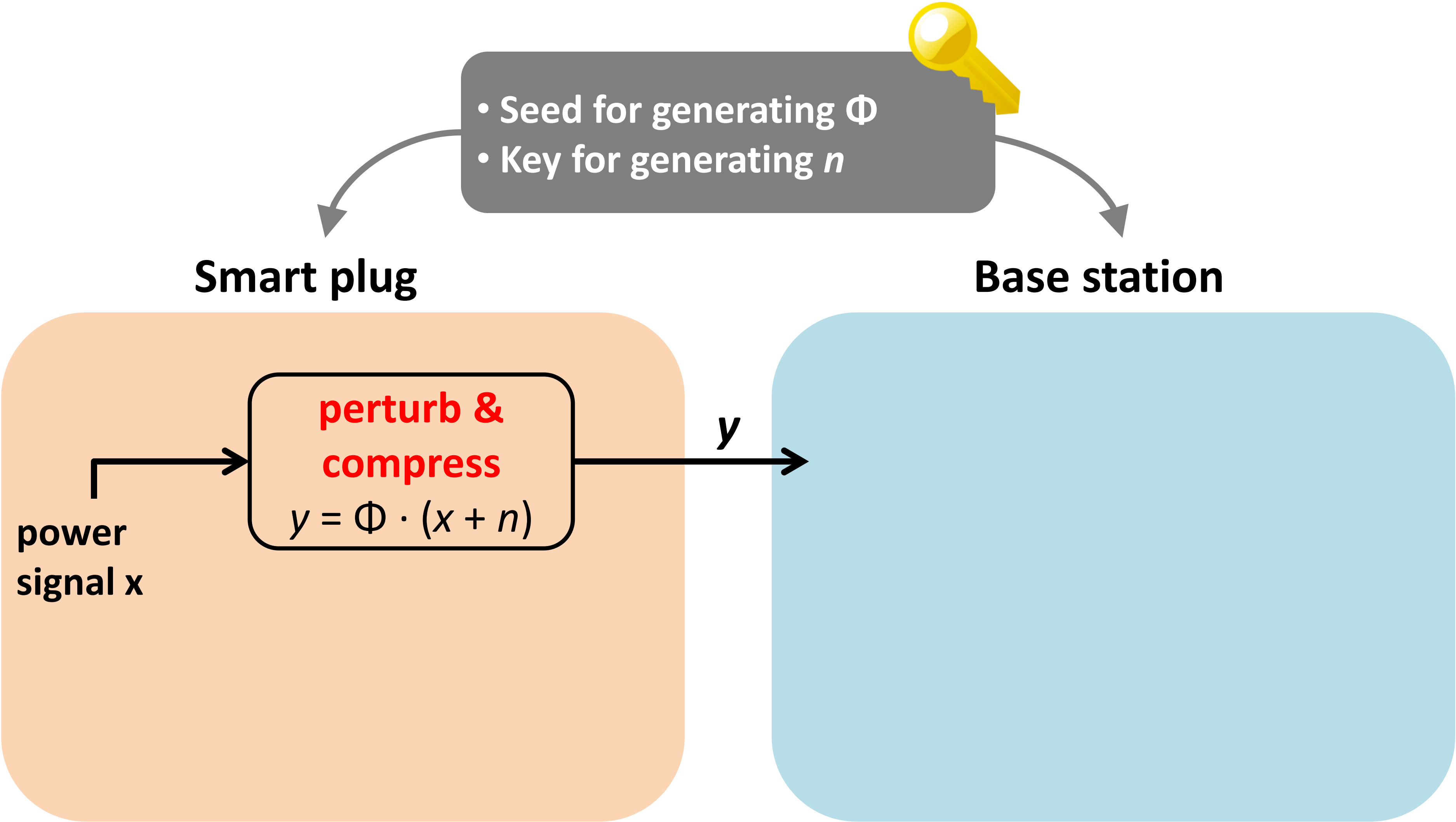


- Seed for generating Φ
- Key for generating *n*

Smart plug

Base station

Recap of JICE



- Seed for generating Φ
- Key for generating *n*

Smart plug

Base station

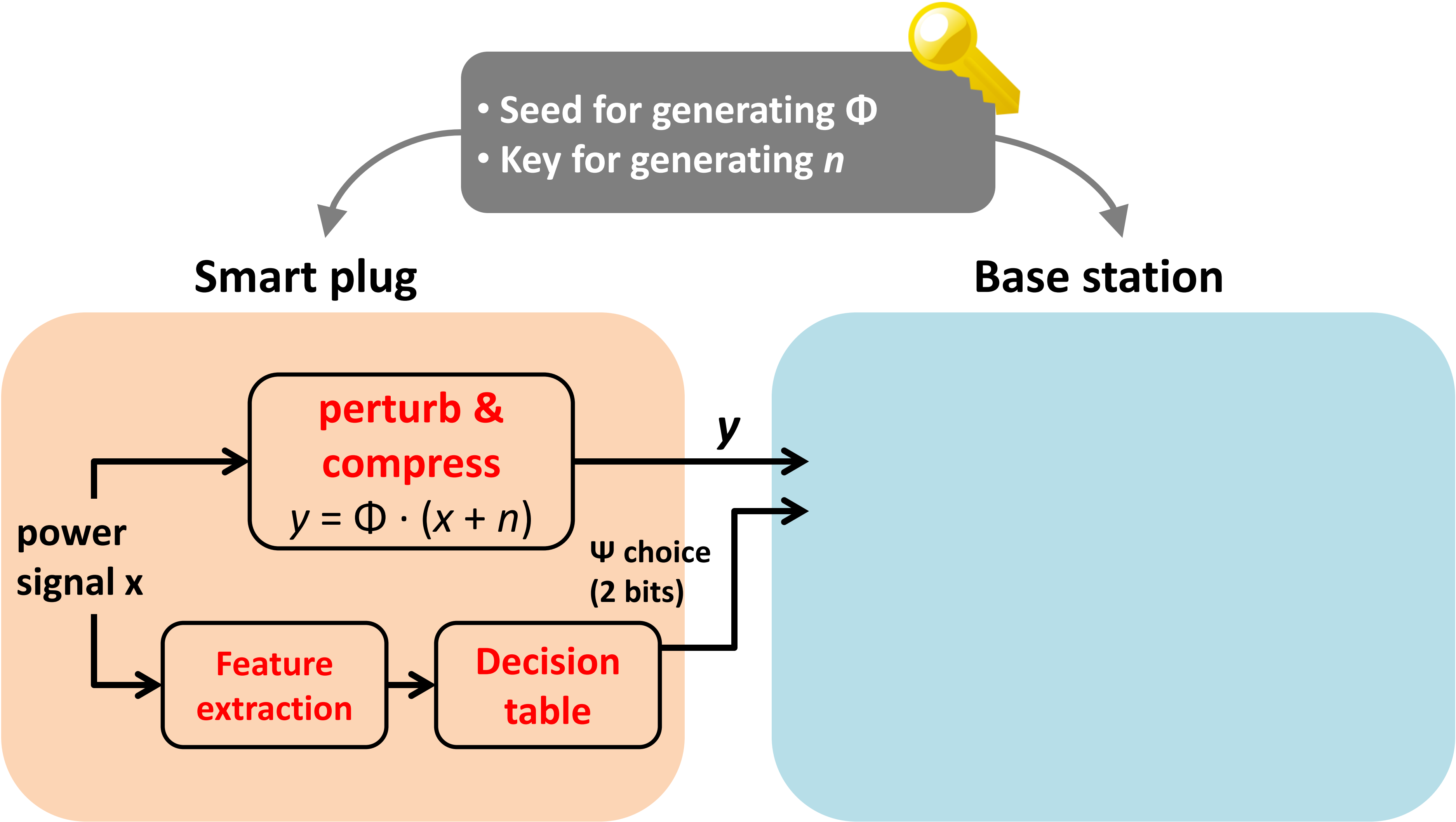
perturb & compress

$y = \Phi \cdot (x + n)$

power signal x

y

Recap of JICE



- Seed for generating Φ
- Key for generating *n*

Smart plug

Base station

power signal x

perturb & compress

$y = \Phi \cdot (x + n)$

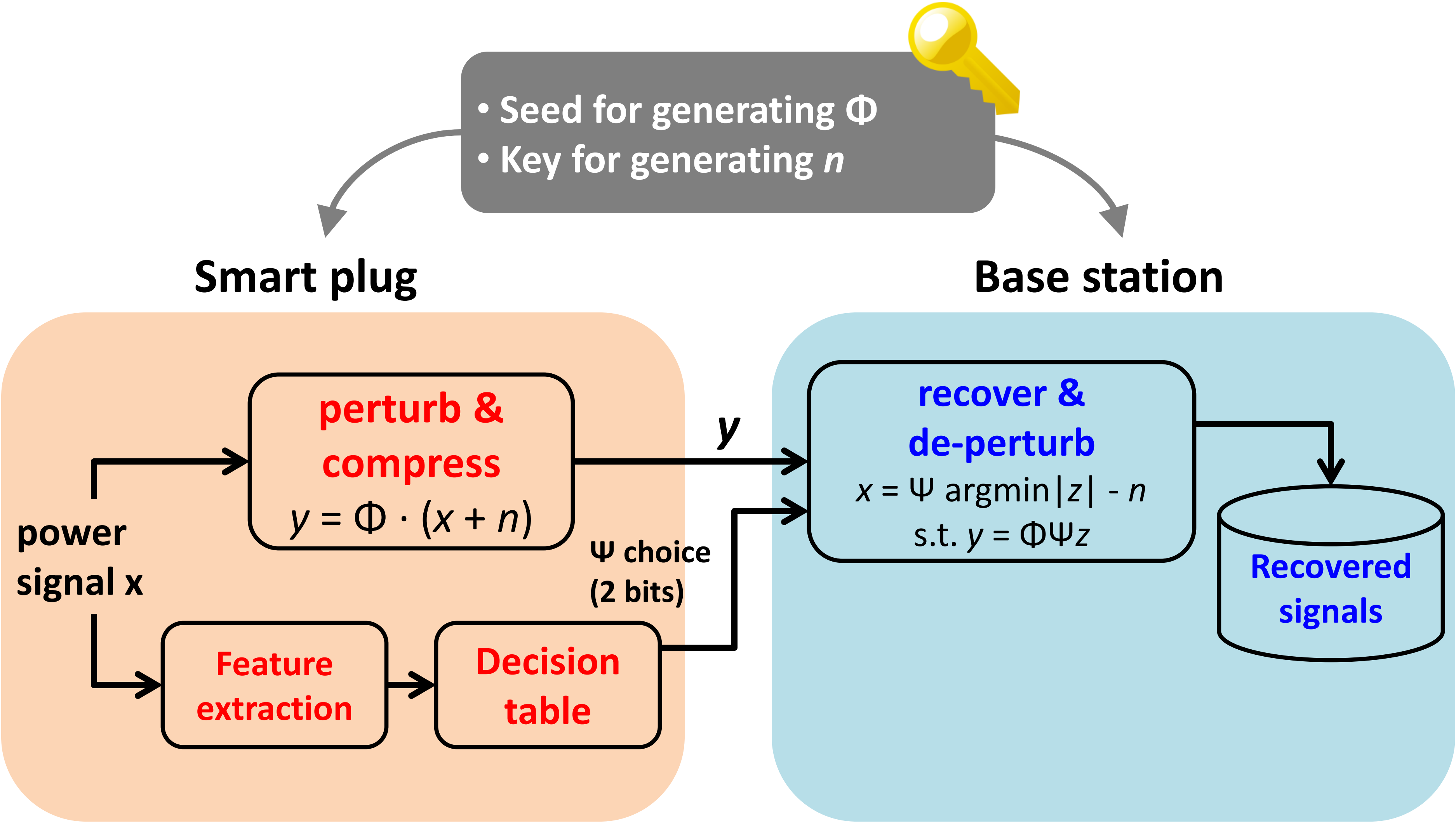
y

Feature extraction

Decision table

Ψ choice (2 bits)

Recap of JICE



- Seed for generating Φ
- Key for generating *n*

Smart plug

Base station

power signal x

perturb & compress

$y = \Phi \cdot (x + n)$

Feature extraction

Decision table

Ψ choice (2 bits)

y

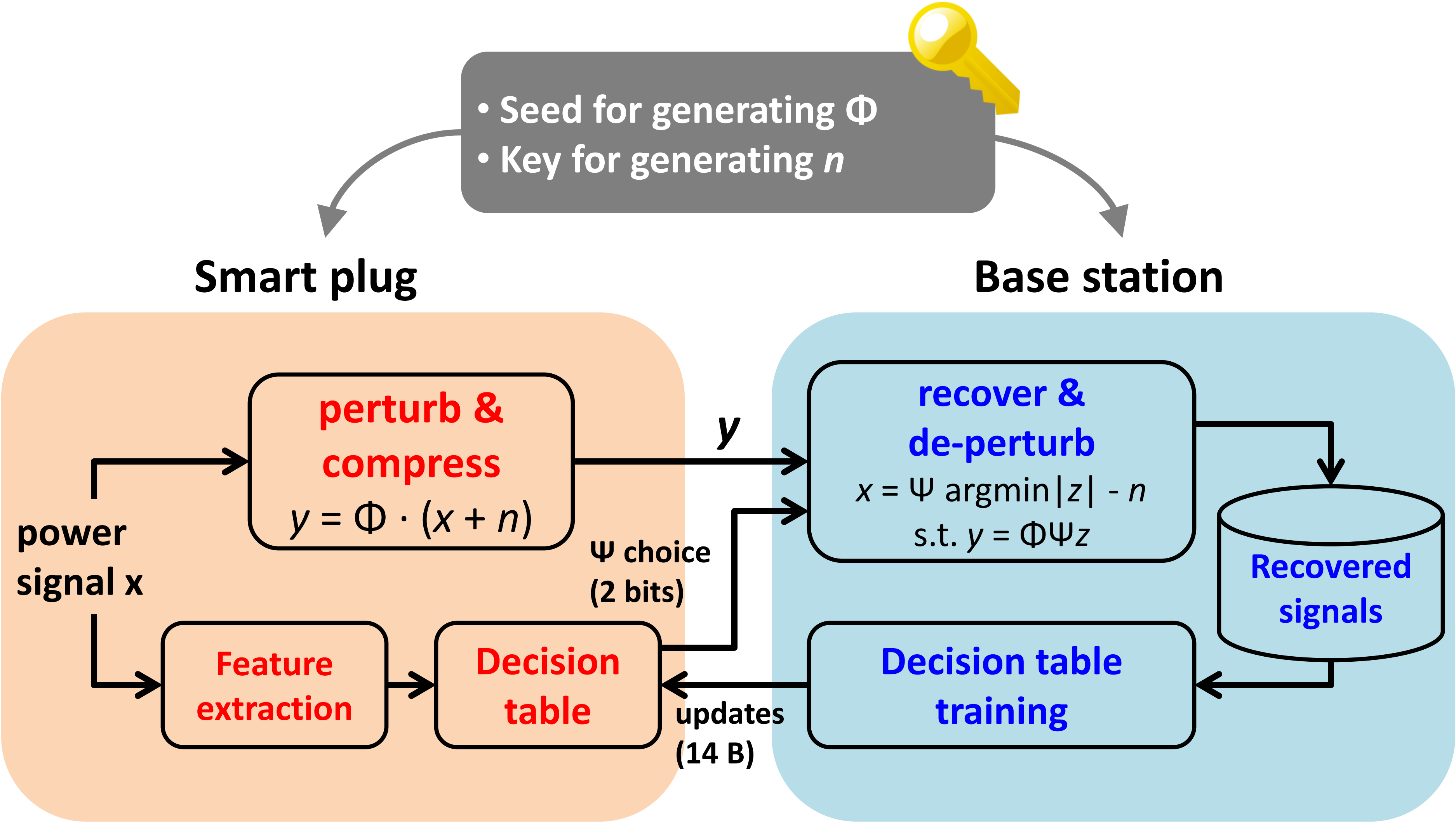
recover & de-perturb

$x = \Psi \,\mathrm{argmin}|z| - n$

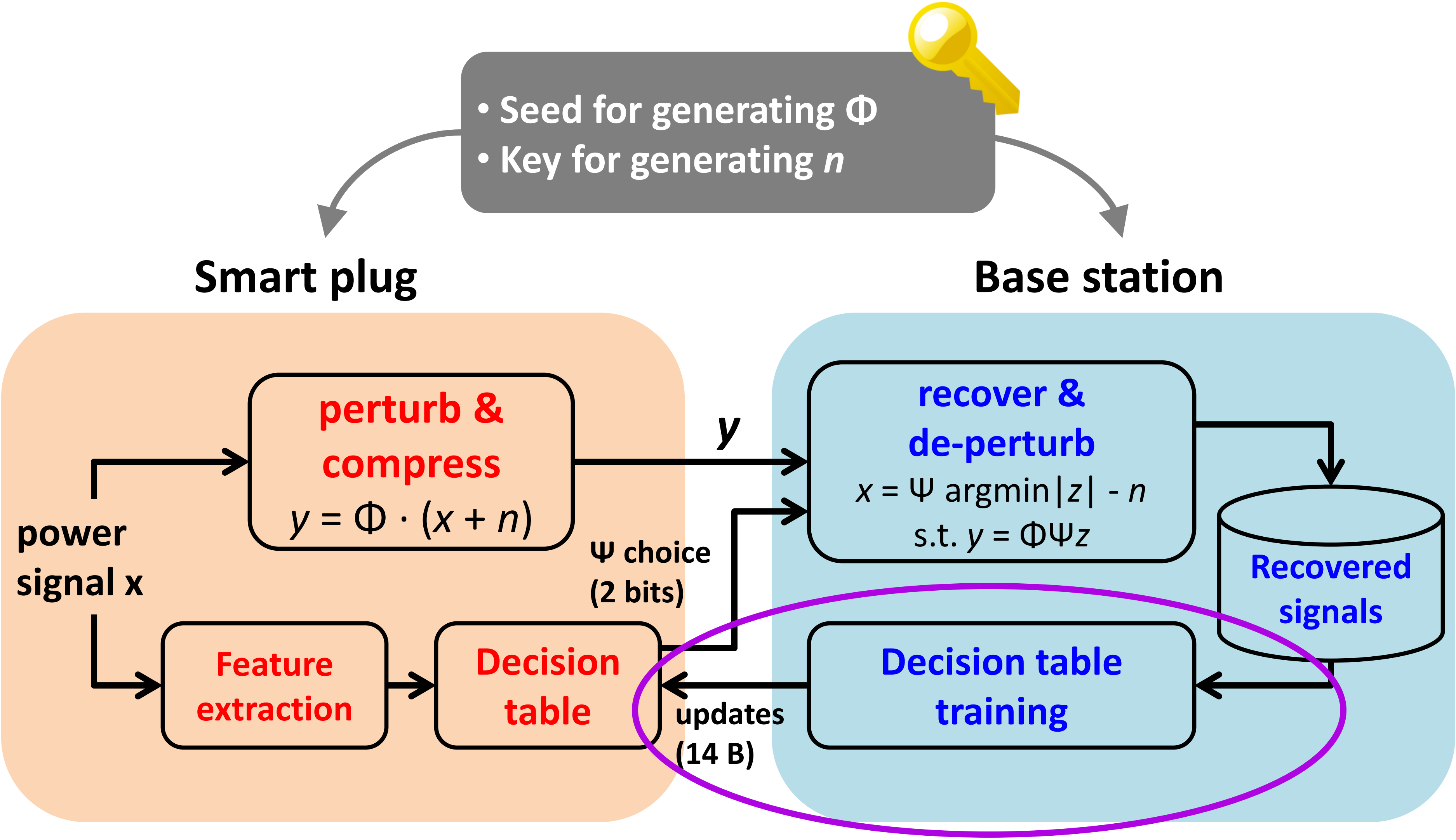
s.t. $y = \Phi \Psi z$

Recovered signals

Recap of JICE



Recap of JICE



- Seed for generating Φ
- Key for generating *n*

Smart plug

Base station

power signal x

perturb & compress

$y = \Phi \cdot (x + n)$

y

recover & de-perturb

$x = \Psi \operatorname{argmin}|z| - n$

s.t. $y = \Phi\Psi z$

Recovered signals

Feature extraction

Decision table

Ψ choice (2 bits)

updates (14 B)

Decision table training

executed every a few hours

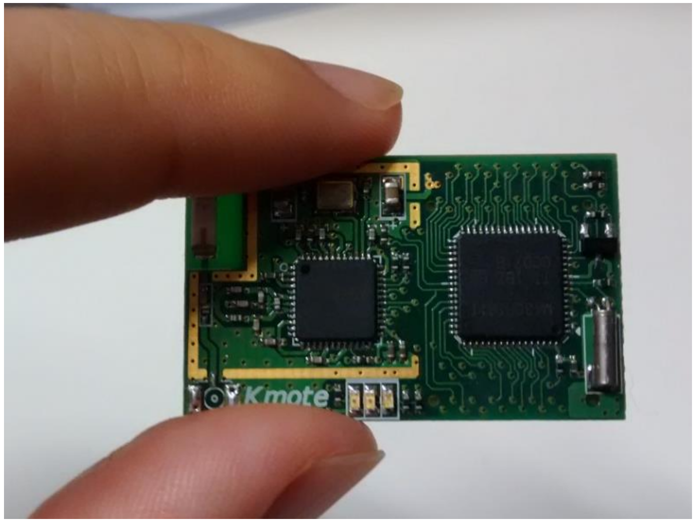
Outline

- Motivation
- Design of JICE
- Secrecy of JICE
- **Experiment**

Implementation

Warning
The maximum allowable current is 16A.
Do not shock and heating.
Decomposition can cause electric shock.
Power Meter
sonnonet

Smart plug
[Sonnonet]



Kmote

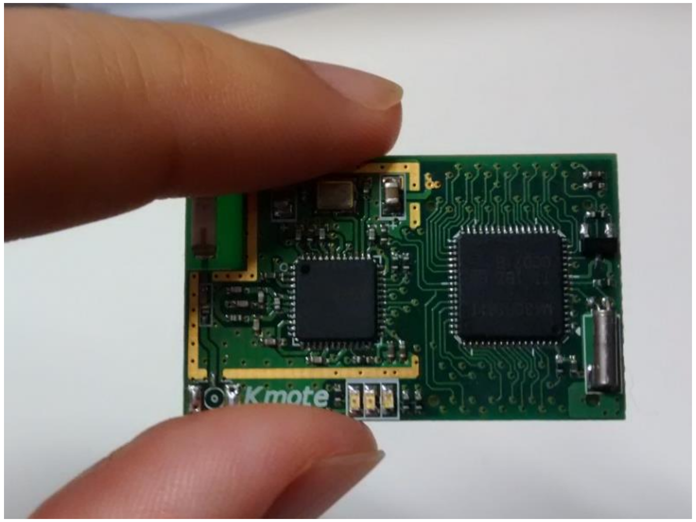
- Smart plug
  - Kmote (8MHz MCU, 10KB RAM, ZigBee, TinyOS)

- Baselines
  - **Pipeline**: Lossy compressor [Liu 2013] + AES
  - **Downsampling**
  - **Lossless pipeline**: SLZW + AES

Implementation

Warning
The maximum allowable current is 16A.
Do not shock and heating.
Decomposition can cause electric shock.
Power Meter
sonnonet

Smart plug
[Sonnonet]



Kmote

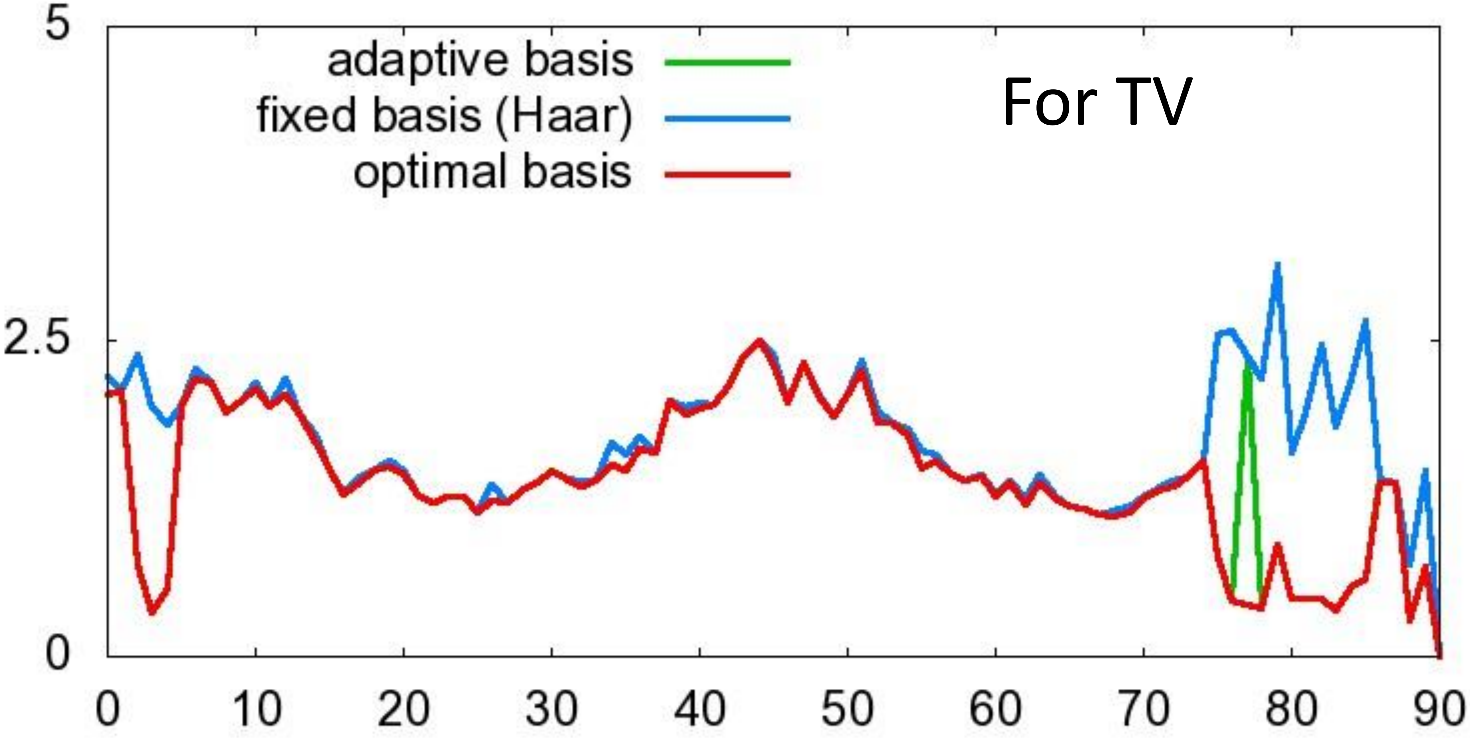
- Smart plug
  - Kmote (8MHz MCU, 10KB RAM, ZigBee, TinyOS)
- Baselines

Same compression ratio with JICE

  - **Pipeline**: Lossy compressor [Liu 2013] + AES
  - **Downsampling**
  - **Lossless pipeline**: SLZW + AES

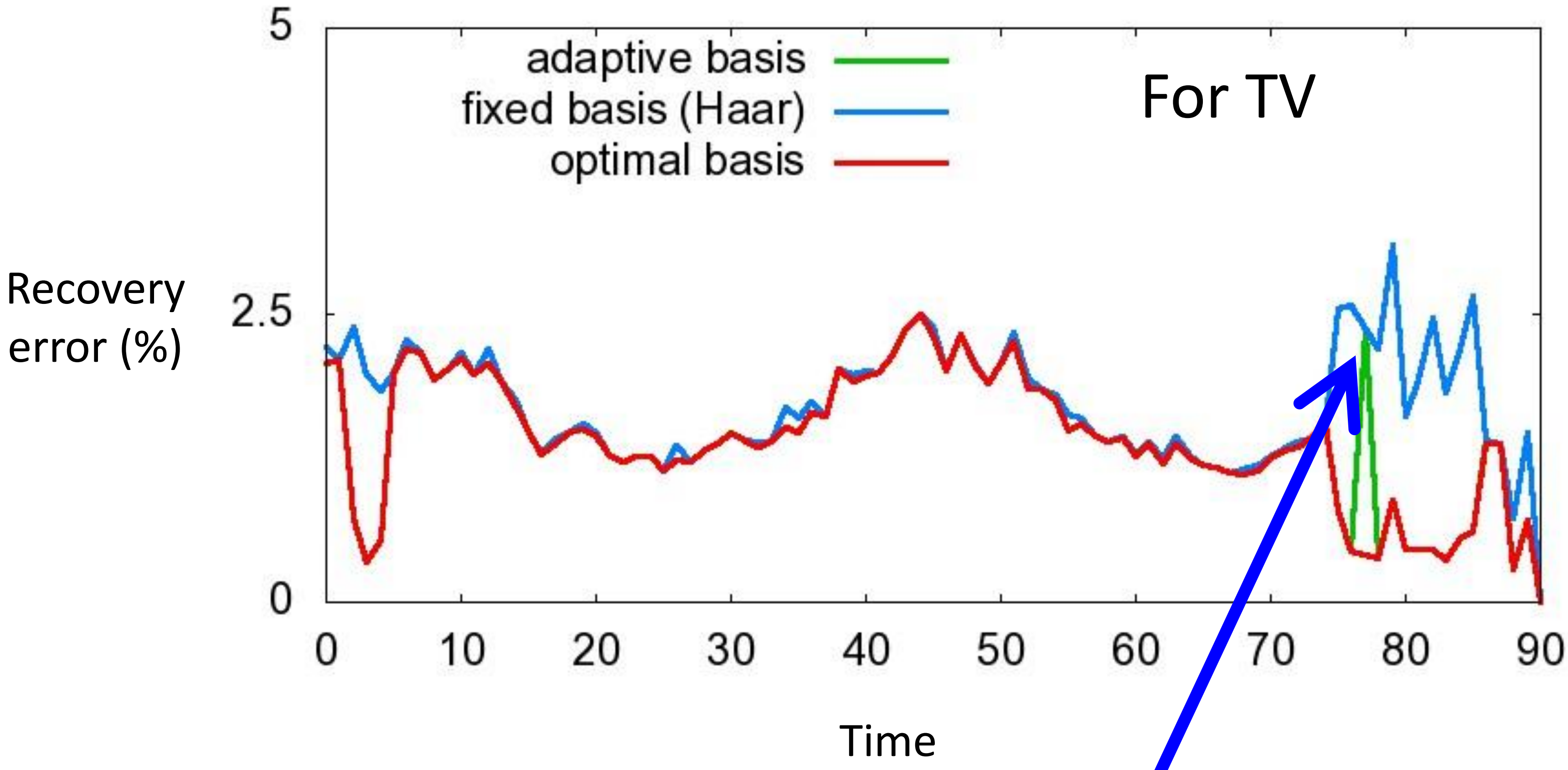
Adaptive Basis vs. Fixed Basis

Recovery error (%)



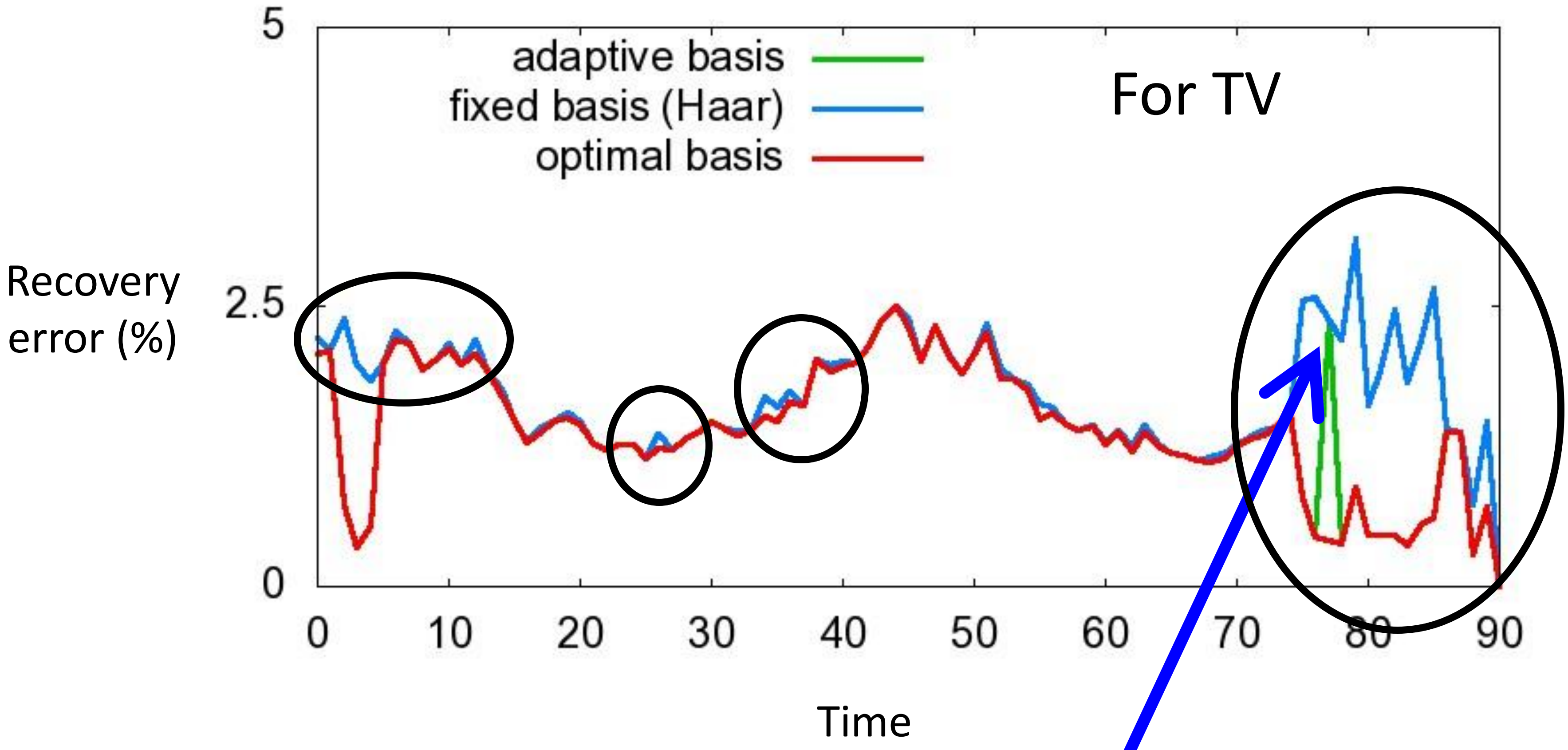
Time

Adaptive Basis vs. Fixed Basis



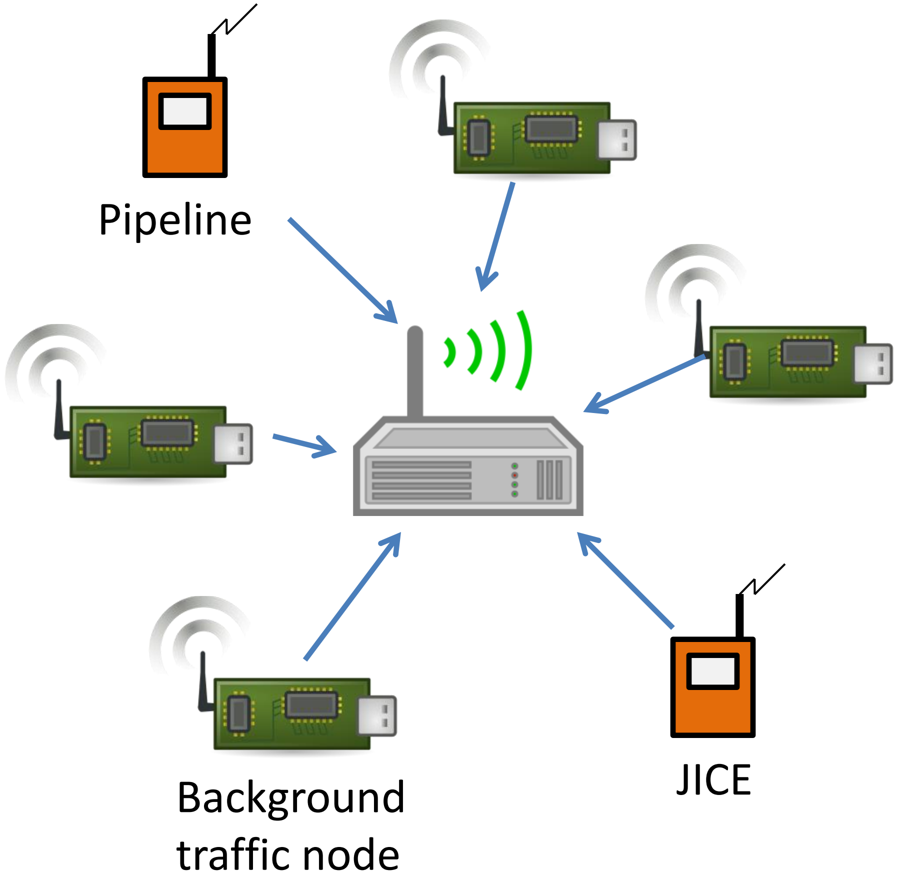
JICE achieves best performance with one exception

Adaptive Basis vs. Fixed Basis

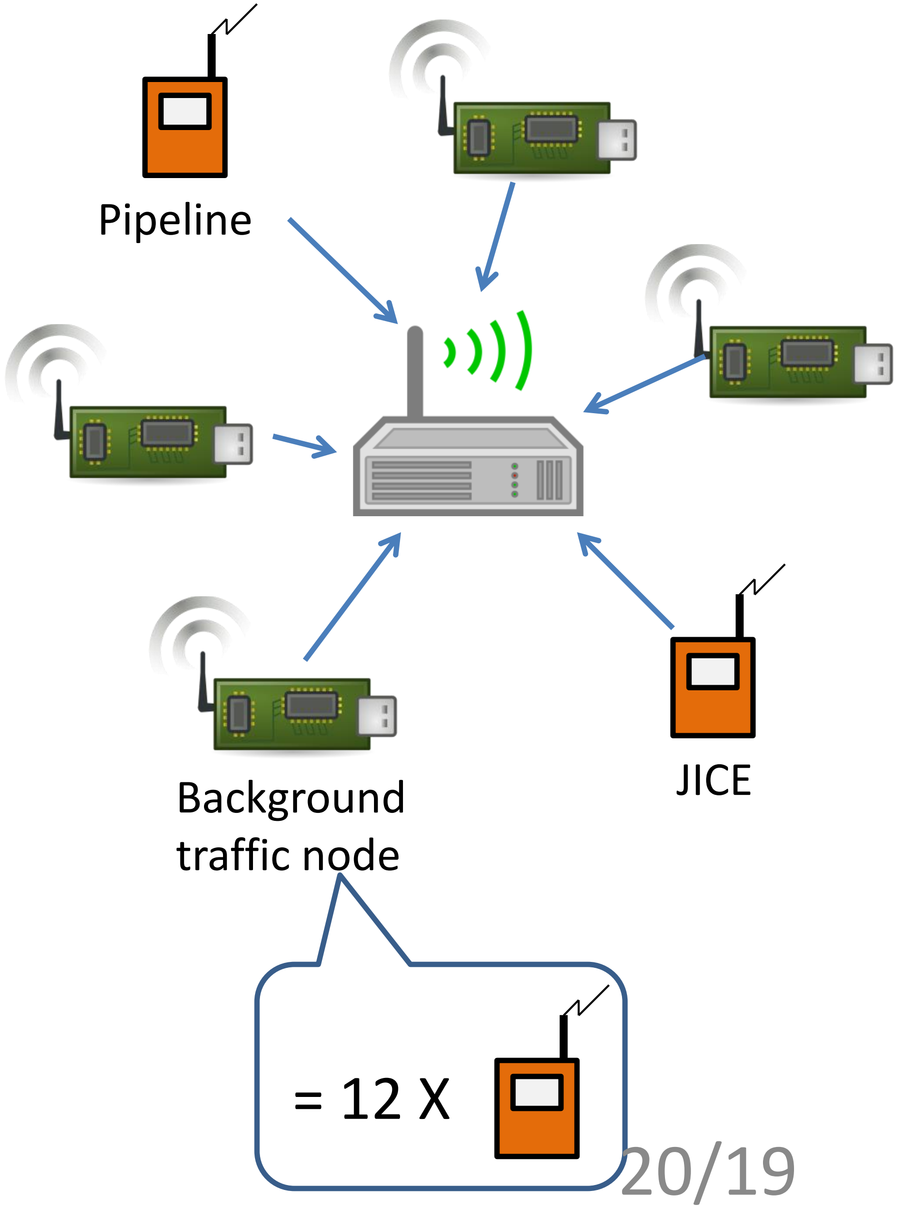


JICE achieves best performance with one exception

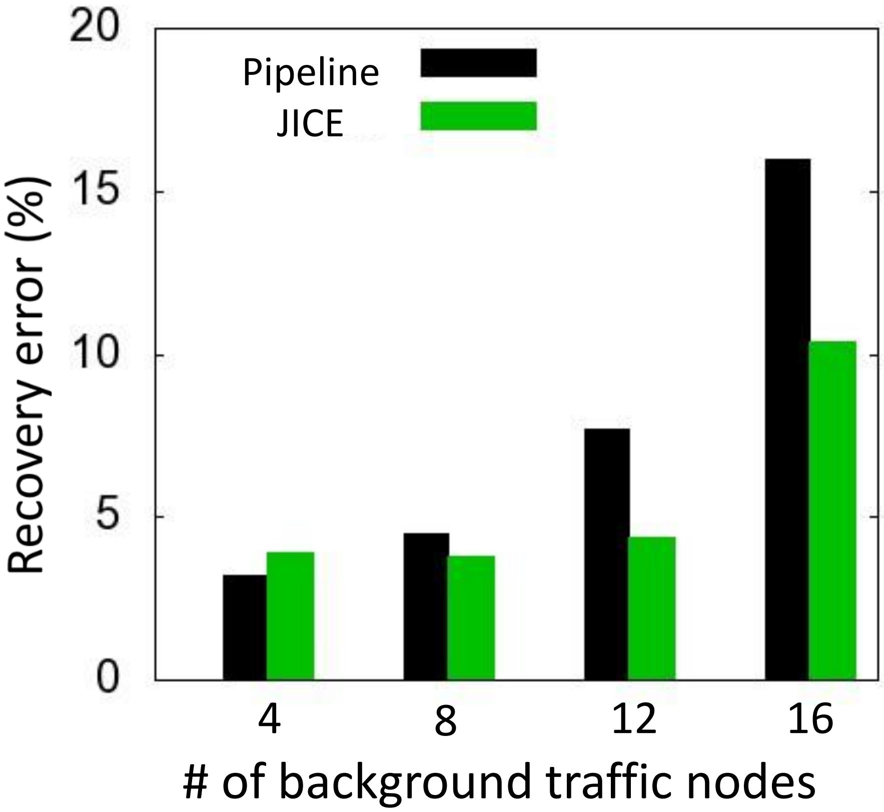
Data Fidelity and Scalability

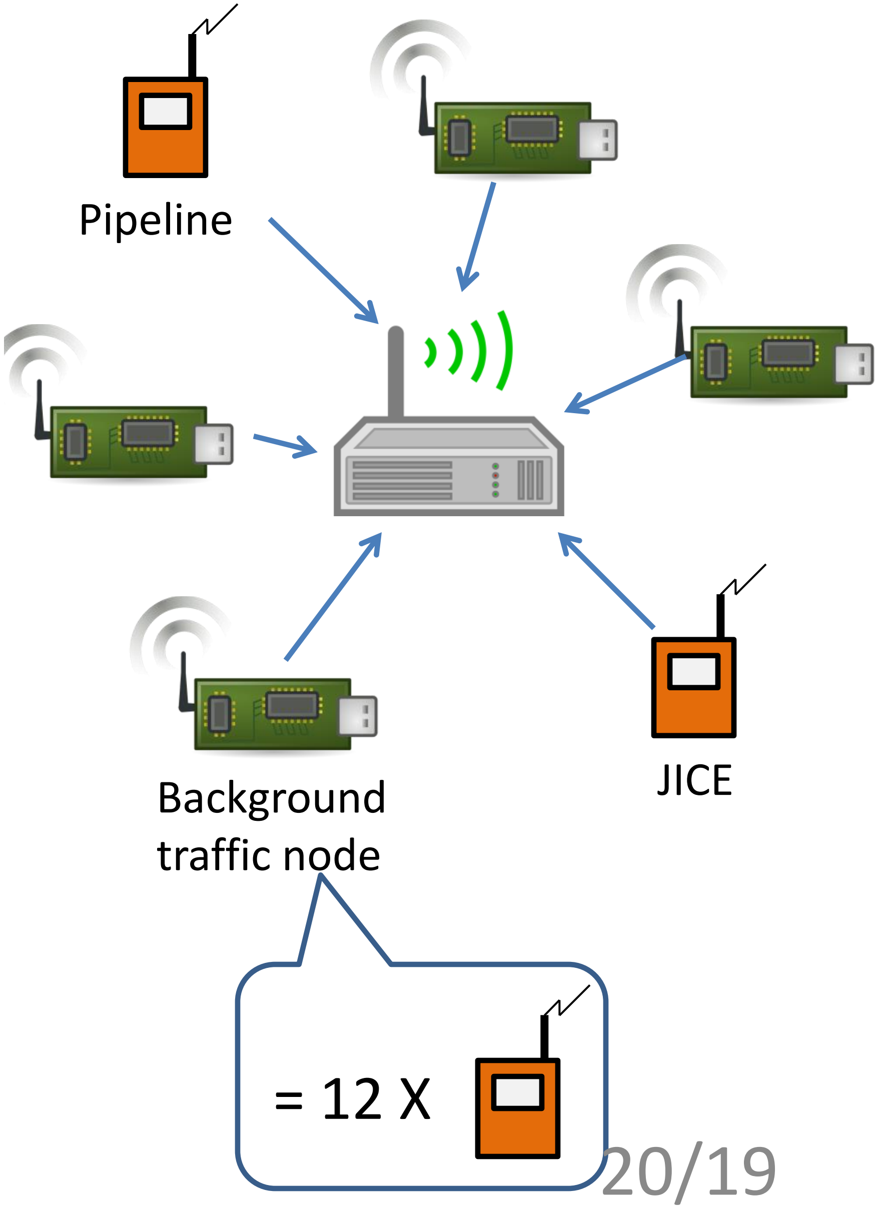


Data Fidelity and Scalability

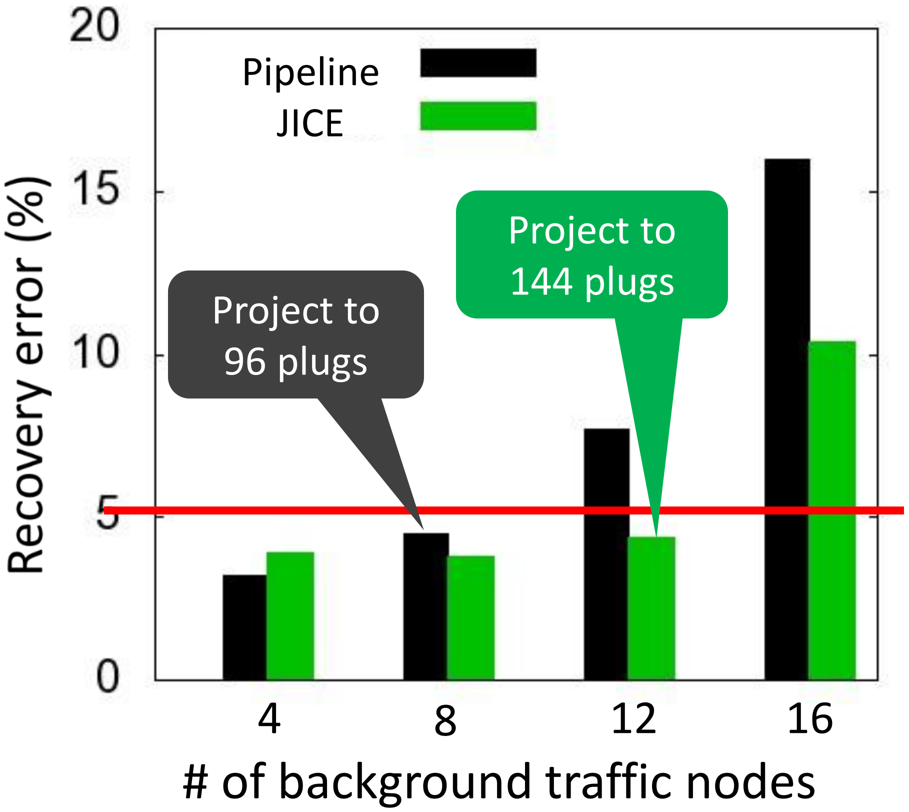


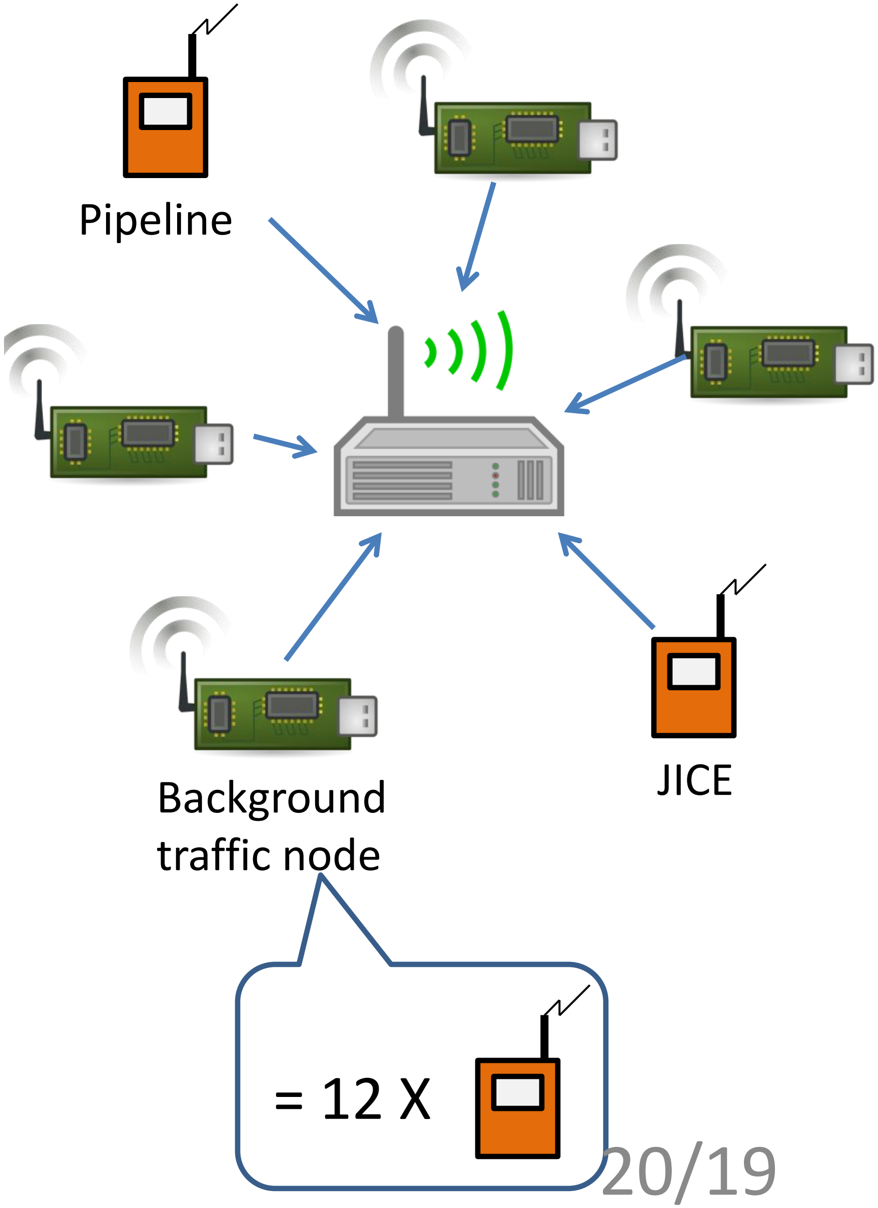
Data Fidelity and Scalability





Data Fidelity and Scalability





JICE supports 50% more plugs

Conclusion & Future work

- JICE
  - Supports more nodes for same data fidelity
  - Better data secrecy than pure compressive sensing
  - Adaptive to changing power pattern
- Future work
  - Other applications