Secure Control Protocols for Resource-Constrained Embedded Systems

Jinkyu Koo
kooj@purdue.edu

School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

June 14, 2012

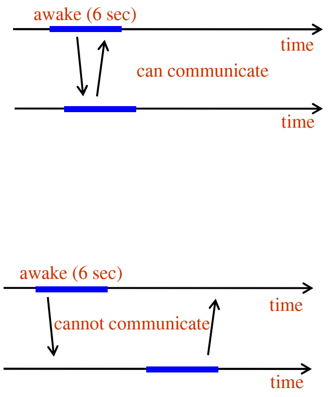
Ph.D. Topics

- Three issues regarding security (defined in a broad sense): reliability, timely reporting, and privacy
  - Situation-tailored solutions for given problems in resource-constrained systems with security in mind.

covered in Prelim

- Reliable and fast synchronization protocol with a good synchronization accuracy.
  - Clock synchronization in a large-scale sensor network, called CSOnet, which is deployed in the city of South Bend, Indiana for monitoring combined sewer overflow events.
- Timely event reporting in sensor networks.
  - In a multi-hop network scenario where all sensor nodes except the base-station node can be compromised, we attempt to secure the event reporting process, while reducing the operational overhead.
- Privacy-preserving data transmission in smart grids.
  - In smart grids, users' specific activity or behavior patterns—whether you are home or not—can be deduced from the fine-granular meter readings. To resolve this issue, we design a mechanism, by which a meter reading reported to the utility is probabilistically independent of the actual usage at any given time instant.

Reliable and Fast Clock Synchronization


awake (6 sec)

time

can communicate

time

awake (6 sec)

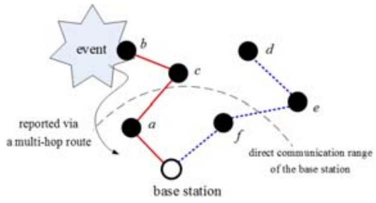
time

cannot communicate

time

- A wide-area Wireless Sensor Actuator Network (WSAN), called CSOnet is in operation in South Bend, IN for detecting and controlling wastewater flow to the treatment plant.
  - 150 wireless nodes monitoring 111 locations
  - CSOnet nodes, called Chasqui, have low duty cycle (2%): awake 6 seconds in a 5 minute period
- The synchronization has to be fast and reliable
  - Ideally entire network should be synchronized within the awake period of 6 seconds
  - The projected scale of the network is large, of the order of a few hundred nodes → high probability that at least one link is in failure
- Made a fast and reliable clock synchronization protocol.

Timely Event Reporting



- Event monitoring: wireless sensor nodes are deployed over a region where some phenomenon is to be monitored.
  - E.g., a number of sensor nodes could be deployed over a battlefield to detect enemy intrusion.
- If an event occurs at a sensor node, the BS gets informed of it as soon as possible in order for the network operator to take action in time.
- However, if a node in the middle of the routing path is compromised, the compromised node may drop/modify the event report, or delay it for a very long time.
- We devise a protocol that provides the following provable security guarantees.
  - As long as the compromised nodes want to stay undetected, a legitimate node can report an event to the BS within P time units.
  - If the compromised nodes launch an attack that causes the event report from a legitimate node not to reach the BS within P time units, the BS can identify a small set of nodes that is guaranteed to contain at least one compromised node.
- We reduce the operational overhead, compared to straw-man solutions.

Privacy Protection in Smart Grids

Problem Statement

- A smart grid is a type of the electrical grid in which electricity delivery systems are equipped with computer-based remote control and automation
  - The smart grid can revolutionize the way that energy is generated and consumed: demand prediction; load balancing by time-of-use pricing
- A key component of the smart grid is the use of the smart meters, which measure energy usage at a fine granularity.
  - e.g., once in a few minutes
- However, by gathering hundreds of data points even in a day via the smart meter, the utility companies and third parties may learn a lot about our daily lives,
  - e.g., when we wake up, when we go out for work, and when we come back after work.

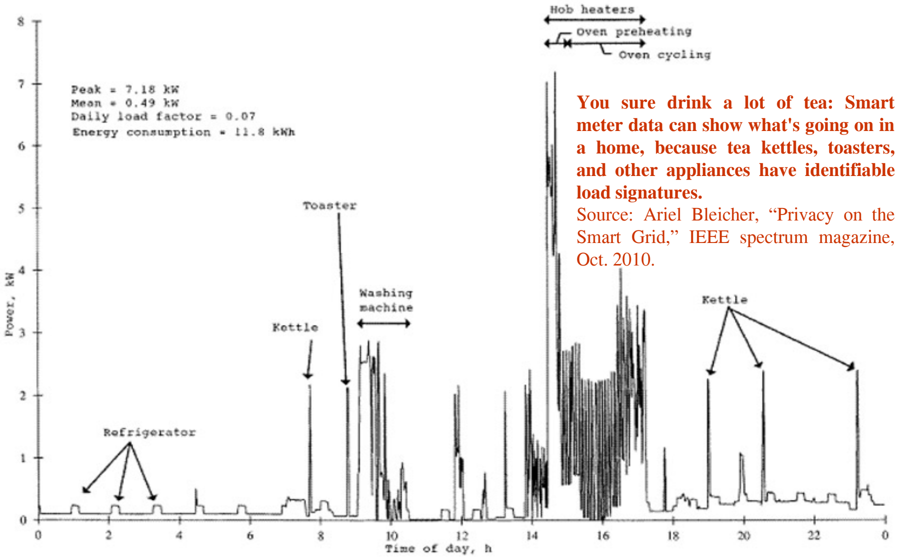
Threat Model

- The collection, retention, and use of detailed usage data put individual privacy at risk.
- Fact: we do need to report our energy usage profile to the utility company for billing purpose.
- An important privacy threat.
  - The metering data may be unwittingly disclosed from the utility company to third-party vendors.

> *Would you sign up for a discount with your power company in exchange for surrendering control of your thermostat? What if it means that, one day, your auto insurance company will know that you regularly arrive home on weekends at 2:15 a.m., just after the bars close? (MSNBC Red Tape Chronicles 2009)*

- Privacy concern has led to lawsuits filed to stop installation of smart meters (NapervilleSun, Dec. 30, 2011).

Example



You sure drink a lot of tea: Smart meter data can show what's going on in a home, because tea kettles, toasters, and other appliances have identifiable load signatures.

Source: Ariel Bleicher, "Privacy on the Smart Grid," IEEE spectrum magazine, Oct. 2010.

Contribution

- We propose a privacy-protection mechanism, called PRIVATUS, that uses a rechargeable battery.

Short-term (within a day): hides load signatures

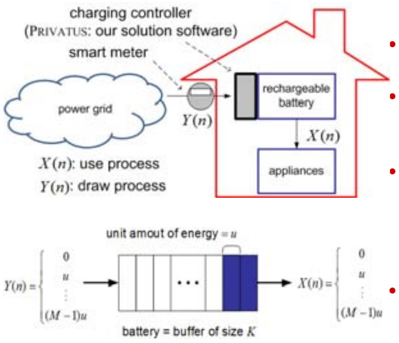
  - In PRIVATUS, the meter reading reported to the utility is probabilistically independent of the actual usage at any given time instant.
  - PRIVATUS also considerably reduces the correlation between the meter readings and the actual usage pattern over time windows.
  - Further, using stochastic dynamic programming, PRIVATUS charges/discharges the battery in the optimal way to maximize savings in the energy cost, given prior knowledge of time periods for the various price zones.

Long-term (within a week): hides whether home or not

  - PRIVATUS can flatten per-day usage.

Part 1: short-term window (within a day)
What are you doing?

System Model (1/2)


charging controller
(PRIVATUS: our solution software)
smart meter
power grid
$Y(n)$
rechargeable battery
$X(n)$
appliances
$X(n)$: use process
$Y(n)$: draw process
unit amout of energy = u
$Y(n) = \{0, u, \vdots, (M-1)u$
$X(n) = \{0, u, \vdots, (M-1)u$
battery = buffer of size K

	low-price zone	high-price zone
intervals	1 to n_L	(n_L+1) to n_H
price rate	R_L	R_H

- Meter reading is measured once every measurement interval (e.g., 15 minutes).
- **X(n)**: the amount of energy consumed by home appliances for the n-th measurement interval.
- **Y(n)**: the amount of energy that we draw from the power grid and charge the battery for the n-th measurement interval.
- X(n) and Y(n) are both represented as one of the **M** different symbols.
  - The i-th symbol is defined as (i-1)u, where u is the unit amount of energy.
  - e.g., when M=4, X(n) and Y(n) is 0,u,2u, or 3u.
- Electricity price per unit amount of energy varies from time to time: there exist two time zones within a day
- Low-price zone: has a low rate R_L (\$/u)
  - The measurement intervals from n = 1 to n = n_L
- High-price zone: has a high rate R_H (\$/u)
  - The measurement intervals from n = n_L+1 to n = n_H
- Can be extended to multiple price zones.

System Model (2/2)

- Denote by B(n) the energy level remaining in the battery at the end of the n-th measurement interval.
  - Assume for simplicity that there is no energy loss when charging and discharging the battery.

$$B(n) = B(0) + \sum_{m=1}^{n} D(m)$$

$B(0)$: the initial energy level of the battery

$$D(m) = Y(m) - X(m)$$

$$0 \leq B(n) \leq Ku$$

- The probability distributions of X(n) and Y (n)

$$P_X(n) = [p_X(0;n), p_X(1;n), \ldots, p_X(M-1;n)]$$

$$P_Y(n) = [p_Y(0;n), p_Y(1;n), \ldots, p_Y(M-1;n)]$$

$$p_X(i;n) = P(X(n) = iu)$$

$$p_Y(i;n) = P(Y(n) = iu)$$

- We assume that $P_X(n)$ is known to the user (i.e., the home owner).
- We also assume that X(n) is independent, but does not need to be identically distributed across the measurement interval index n.
  - This means that for instance, X(5) is independent of X(11), and $P_X(5)$ can be different from $P_X(11)$.
  - If the family leaves home for work/school at 8 a.m., then clearly the usage before 8 a.m. and after 8 a.m. will be different

Mapping between X(n) and Y(n) (1/2)

- We make Y(n) be independent of X(n).
  - This implies that observing Y (n) gives no meaningful information about X(n).
  - This is achieved when we map X(n) to Y (n) in such a way that

$$p_Y(i;n) \equiv P(Y(n) = iu) = P(Y(n) = iu|X(n) = ju)$$

- Practically, we achieve this by probabilistically choosing the value of Y(n) according to P_Y(n), which is decided before the n-th measurement interval starts, without considering what the value of X(n) will be.
- However, selecting Y(n) randomly without being aware of X(n) may cause energy shortage or overflow in the battery.
  - When B(n-1) = 0 (i.e., there is no energy remaining in the battery before the n-th measurement interval starts), if Y (n) is chosen to be zero, we cannot feed any non-zero value of X(n). This means that sometimes we cannot use the appliances when we want!
  - Similarly, when B(n-1) = Ku (i.e., the battery is full), a non-zero value of Y(n) does not make sense if X(n) = 0, since we cannot draw the energy from the power grid unless we throw it away.

B(n-1)=0 (empty)
Y(n)=0, X(n)=3u
B(n)=B(n-1)+Y(n)-X(n)=-3u → shortage

B(n-1)=Ku (full)
Y(n)=3u, X(n)=0
B(n)=Ku+3u → overflow

PURDUE
UNIVERSITY

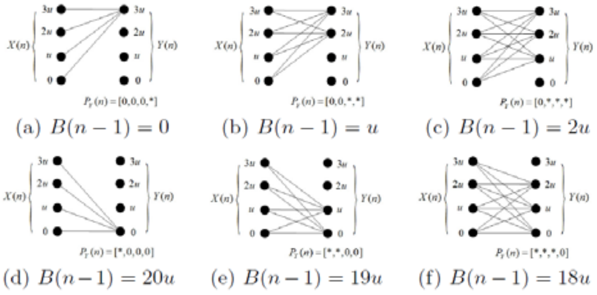
Mapping between X(n) and Y(n) (2/2)

- We put a restriction on P_Y(n) in the corner cases, i.e., when the energy left in the battery is smaller than (M-1)u (near-empty) or larger than (K-(M-1))u (near-full).

$$B(n-1) = ju \text{ for } j < (M-1) \longrightarrow p_Y(i;n) = 0 \text{ for } i < (M-1) - j$$

$$B(n-1) = (K-j)u \text{ for } j < (M-1) \longrightarrow p_Y(i;n) = 0 \text{ for } i > j$$

- An example of the probabilistic symbol mapping between X(n) and Y(n) in the corner cases when K=20 and M=4. The symbol '*' in P_Y(n) represents the element that can be non-zero.

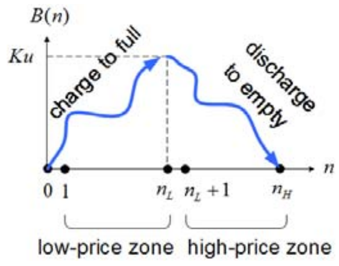


(a) $B(n-1) = 0$ (b) $B(n-1) = u$ (c) $B(n-1) = 2u$

(d) $B(n-1) = 20u$ (e) $B(n-1) = 19u$ (f) $B(n-1) = 18u$

PURDUE
UNIVERSITY

Strategy to Achieve the Maximum Cost Saving



R_L dollar per u R_H dollar per u

(u: the unit amount of energy)

charge *iu* in the low-price zone
→pay R_Li
use the stored *iu* in the high-price zone
→pay 0, instead of paying R_Hi
→save (R_H-R_L)i

- How can we achieve cost saving by exploiting the time-of-use pricing policy in smart grid?
  - The only way to achieve the cost saving is to charge the battery in the low-price zone and use the stored energy in the high-price zone.
- If we charge *iu* amount of energy in the low-price zone and use it in the high price zone, we can save (R_H -R_L)i dollars.
- The maximum possible cost saving per day is (R_H -R_L)K dollars, which is obtained
  - when we charge the battery from empty to full in the low-price zone and
  - discharge the battery to zero by feeding X(n) in the high-price zone.
- We implement this by changing P_Y(n) for every n.

Basic Approach (1/2)

- Define the distribution vector space

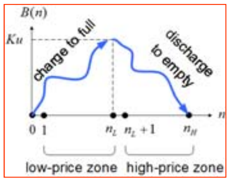
$$\mathcal{P} = \left\{ [p_0, p_1, \ldots, p_{(M-1)}] : \sum_{i=0}^{M-1} p_i = 1, 0 \le p_i \le 1 \right\}$$

[0,0,0,1]
[0,0,0.5,0.5]
[0.1,0.2,0.3,0.4]
…

- We limit the value of p_i to be a multiple of a constant c (0<c<1), in order to make P be a finite set.
- P_Y(n) is assigned one element in P in the n-th measurement interval.
  - Recall that we force some elements of P_Y(n) to be zero, depending on the battery state.
- Therefore, the possible choice set in the n-th measurement interval is dependent on B(n-1) and we denote it by $P_{B(n-1)}$.

Basic Approach (2/2)

- Now, what would be the best choice for $P_Y(n)$ in $P_{B(n-1)}$ for each n to maximize the cost saving?
- This question is answered by solving the following stochastic optimal control problems:



In the low-price zone

$$\max_{\substack{P_Y(n)\in\mathcal{P}_{B(n-1)}\\ 0<n\leq n_L}} E\left(B(n_L)|B(0),P_Y(1),P_Y(2),\ldots,P_Y(n_L)\right)$$

In the high-price zone

$$\min_{\substack{P_Y(n)\in\mathcal{P}_{B(n-1)}\\ n_L<n\leq n_H}} E\left(B(n_H)|B(n_L),P_Y(n_L+1),P_Y(n_L+2),\ldots,P_Y(n_H)\right)$$

Dynamic Programming (1/3)

- Consider a simple example in the low-price zone, where $n_L = 3$.

$$B(n) = B(0) + \sum_{m=1}^{n} D(m)$$

$$E\left(B(3)|B(0),P_Y(1),P_Y(2),P_Y(3)\right)$$
$$= B(0) + E\left(\sum_{n=1}^{3} D(n)|B(0),P_Y(1),P_Y(2),P_Y(3)\right)$$

$$E\left(\sum_{n=1}^{3} D(n)|B(0),P_Y(1),P_Y(2),P_Y(3)\right)$$
$$= E\left(D(1)|B(0),P_Y(1)\right)+E\left(D(2)|B(1),P_Y(2)\right)+E\left(D(3)|B(2),P_Y(3)\right)$$
$$= E\left(D(1)|B(0),P_Y(1)\right)+E\left(D(2)+E\left(D(3)|B(2),P_Y(3)\right)|B(1),P_Y(2)\right)$$
$$= \underbrace{E\left(D(1)+\underbrace{E\left(D(2)+\underbrace{E\left(D(3)|B(2),P_Y(3)\right)}_{\text{stage 3}}|B(1),P_Y(2)\right)}_{\text{stage 2}}|B(0),P_Y(1)\right)}_{\text{stage 1}}$$

B(n-1) n	1	2	3
0	H	D	A
1u	I	E	B
2u	J	F	C
⋮			

Optimal $P_Y(3)$ given $B(2)$=2u

- Note that the calculations can be done recursively: Stage 2 calculations are based on stage 3, stage 1 only on stage 2.
- Thus, maximizing this can be performed by maximizing the stage 3, stage 2, and stage 1 in this order.
- In this manner, we first compute the optimal value of $P_Y(3)$ given B(2), then we compute the optimal value of $P_Y(2)$ given B(1) until we reach and compute the optimal value of $P_Y(1)$.

Dynamic Programming (2/3)

- In general, this backward (time-wise) directional computation procedure can be described by the following recursive equation, called the Bellman equation:

$$J(n_L+1, B(n_L)) = 0,$$
$$J(n, B(n-1)) = \max_{P_Y(n) \in \mathcal{P}_{B(n-1)}} E\left(D(n) + J(n+1, B(n)) | B(n-1), P_Y(n)\right)$$

- Solving the Bellman equation in the backward direction (from $n=n_L$ to $n=1$) results in the optimal decision for $P_Y(n)$ when the value of $B(n-1)$ is given, in the sense that $P_Y(n)$ will maximize $E(B(n_L))$.

$$E\left(D(n) + J(n+1, B(n)) | B(n-1), P_Y(n)\right)$$
$$= \sum_{\substack{-(M-1) \le j \le (M-1), \\ 0 \le i+j \le K}} p_{i,(i+j)}(n) \left(j + J(n+1, (i+j)u)\right)$$

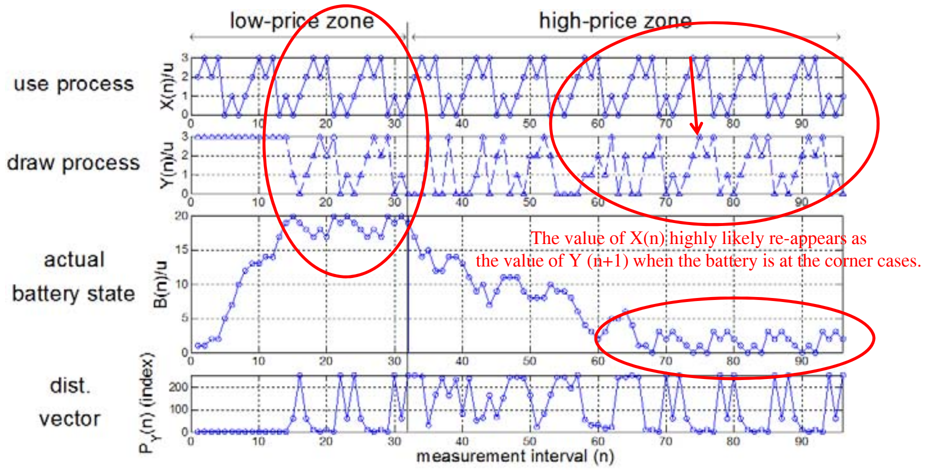
•Here, $p_{i,(i+j)}(n)$ denotes the probability of the transition from $B(n-1) = iu$ to $B(n) = (i+j)u$, resulting from $D(n) = ju$.
•Simply speaking, $p_{i,(i+j)}(n)$ is a function of $P_X(n)$ and $P_Y(n)$.

Dynamic Programming (3/3)

- In summary, what we have done is to calculate a decision table.
- Each entry in the decision table maps the given values of n and B(n-1) to the optimal vector $P_Y(n)$ at the state (n,B(n-1)).
- Note that the decision table can be pre-calculated before the run-time.
- During the run-time, we just look up the decision table for a given state, i.e., (n,B(n-1)), and probabilistically choose the value of Y(n) via the distribution specified by the decision table entry.

Simulation Results for the Basic Approach

There exist similar patterns between the sequences of X(n) and Y (n)



The value of X(n) highly likely re-appears as the value of Y (n+1) when the battery is at the corner cases.

Issues with the Basic Approach

- First, we charge/discharge the battery too fast.
  - In the low-price zone, once at the full state, the battery stays close to the near-full states, since there is no benefit to bring the energy level down to a lower one according to our optimization objective.
  - The near-constant energy level of the battery implies that whatever the value of X(n) is, the draw process Y(n) should somehow compensate for it.
  - Since the value of Y(n) is chosen before the value of X(n), we see this compensation effect in Y(n+1).
- Second, we have too much freedom when choosing $P_Y(n)$.
  - As a result, the draw process can take a specific symbol with a very high probability to compensate the use process.
  - In other words, due to the high degree of freedom to choose $P_Y(n)$, Y(n) is chosen to be very similar to X(n-1) in the corner cases.

$$\max_{\substack{P_Y(n)\in\mathcal{P}_{B(n-1)}\\ 0<n\le n_L}} E\left(B(n_L)|B(0), P_Y(1), P_Y(2), \ldots, P_Y(n_L)\right)$$

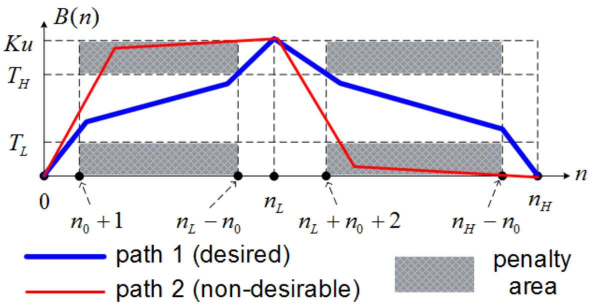
B(n-1)=20u (full)
Y(n)=0 (must), X(n)=3u (random)
B(n)=17u
Y(n+1)=3u (highly likely), X(n+1)=u (random)
…

probability
of choosing 3u

How likely?
→$P_Y(n)$ = [0, 0, 0, 1]
→with 100% probability
→deterministic!

Advanced Approach: PRIVATUS (1/3)

- How to control the speed of charging/discharging?
  - We modify our optimization objective in such a way that we incur some penalty, whenever the battery state B(n) falls into the penalty areas.
  - Most of the corner cases are covered by the penalty areas.
- The optimal decision for $P_Y(n)$ would be changed to the one that still charges or discharges the battery according to the trend as before, but does not hit the penalty areas in the middle of the zones.



Advanced Approach: PRIVATUS (2/3)

- Maximize the effective battery state $B_e(n)$ in the optimization objective function, instead of the actual battery state B(n).
  - $B_e(n)$ is designed to increase as the actual battery state B(n) increases in the low-price zone.
  - However, every time B(n) goes into a penalty area, $B_e(n)$ is deducted by some penalty amount.

$$\boxed{B(n) = B(0) + \sum_{m=1}^{n} D(m)}$$

$$B_e(n) = B_e(0) + \sum_{m=1}^{n} D_e(m) \quad \text{with } B_e(0) = \alpha B(0)$$

If m $\le n_0$ or m $> n_L - n_0$: $D_e(m) = \alpha D(m)$

(i.e., in near-beginning or near-end of the low-price zone)

$$\boxed{D(m) = Y(m) - X(m)}$$

If m $> n_0$ and m $\le n_L - n_0$: $D_e(m) = \alpha D(m) - \beta\left(\left[B(m) - T_H\right]^+ + \left[T_L - B(m)\right]^+\right)$

α, β : integers; determine how sensitive to the penalty $\left[x\right]^+$: x if x>0; otherwise 0

$T_H = K - (M-1)$ (too high)

$T_L = M - 1$ (too low)

: thresholds of the corner cases

Advanced Approach: PRIVATUS (3/3)

- How to limit the freedom of choosing $P_Y(n)$?
  - Force the different elements of $P_Y(n)$ in $P_{B(n-1)}$ to be more or less equal, thus eliminating the possibility that Y(n) is chosen deterministically (or with a high probability).

$$\mathcal{P}_{ku} = \{v \in \mathcal{P} : \|v - V_k\| < T_k\}$$

- T_k is a threshold at B(n-1) = ku.
- V_k is the distribution vector of Y(n) for which the possible values of Y(n) at B(n-1) = ku are selected equi-probably.

e.g., $V_5 = [0.25, 0.25, 0.25, 0.25]$, $V_1 = [0.5, 0.5, 0, 0]$

  - Put a restriction on $P_Y(n)$ in non-corner cases (i.e., battery neither empty nor full) such that it does not differ significantly from $P_Y(n-1)$.

$$\|P_Y(n) - P_Y(n-1)\| < T_D$$: distance threshold

    - In the extreme case, $P_Y(n-1) = P_Y(n)$ implying that Y(n) is independent of X(n-1).
    - In order to quickly get out of the corner cases, we enforce this restriction to be applied only when the actual battery state stays in non-corner cases for two consecutive measurement intervals.

Dynamic Programming in PRIVATUS

- In the low-price zone

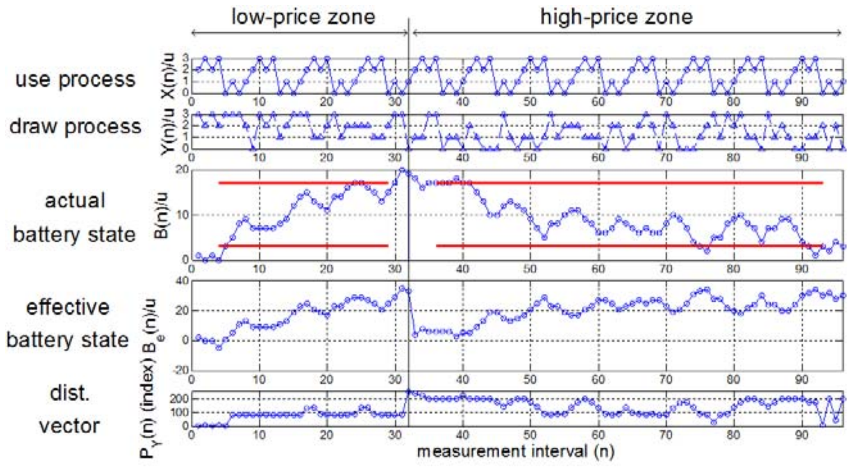
$$\boxed{\begin{aligned} &J(n_L+1, B(n_L)) = 0, \\ &J(n, B(n-1)) = \max_{P_Y(n) \in \mathcal{P}_{B(n-1)}} E\left(D(n) + J(n+1, B(n)) | B(n-1), P_Y(n)\right) \end{aligned}}$$

$$J(S(n_L+1)) = 0,$$
$$J(S(n)) = \max_{P_Y(n) \in \mathcal{P}^*_{B(n-1)}} E\left(D_e(n) + J(S(n+1)) | S(n)\right)$$

$$S(n) = [n, B(n-1), B_e(n-1), P_Y(n-1)]$$: state vector of four different dimensions

$$\mathcal{P}^*_{B(n-1)} = \mathcal{P}_{B(n-1)} \cap \underbrace{\{v \in \mathcal{P} : \|v - P_Y(n-1)\| < T_D\}}_{\text{if } T_L \le B(n-2) \le T_H \text{ and } T_L \le B(n-1) \le T_H}$$

Simulation Results for PRIVATUS



Evaluations

- Two aspects
  - How much privacy information is revealed
  - How much electricity cost is saved
- The metric of information leakage from the use process to the draw process

$$L^s_{(n,m)} = I(\bar{X}_{(n,m)}; \bar{Y}^s_{(n,m)})/H(\bar{X}_{(n,m)})$$

$$\bar{X}_{(n,m)}=[X(n-m+1),X(n-m),\ldots,X(n)] \qquad \bar{Y}^s_{(n,m)}=[Y(n-m+1+s),Y(n-m+s),\ldots,Y(n+s)]$$

$$H(\mathcal{X}) = -\textstyle\sum_i P(\mathcal{X}=i)\log P(\mathcal{X}=i) \qquad I(\bar{X}_{(n,m)};\bar{Y}^s_{(n,m)}) = H(\bar{X}_{(n,m)}) - H(\bar{X}_{(n,m)})|\bar{Y}^s_{(n,m)})$$

a measure of uncertainty

a measure of the uncertainty reduction

- The metric for the cost saving for a day

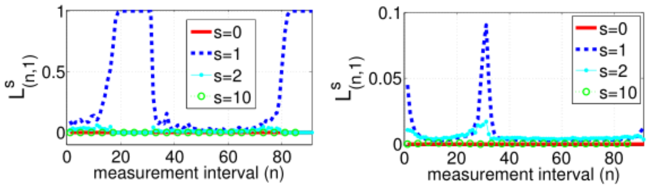
$$\boxed{r = R_L/R_H}$$

$$S_{(r,K)} = E\left(-\sum_{m=1}^{n_L} rR_H D(m) - \sum_{m=n_L+1}^{n_H} R_H D(m)\right)$$
$$= E(\textstyle\sum_{m=1}^{n_L} rR_H X(m) + \sum_{m=n_L+1}^{n_H} R_H X(m)$$
$$- \textstyle\sum_{m=1}^{n_L} rR_H Y(m) - \sum_{m=n_L+1}^{n_H} R_H Y(m))$$

the original cost for what the user actually consumes

the money that a user pays to the utility company

General Performance Trend



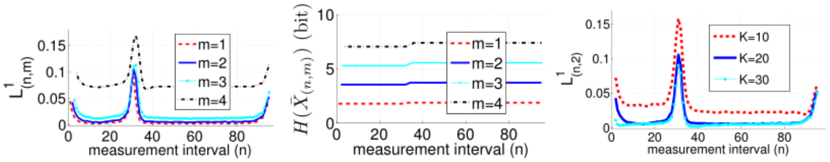
(a) Basic approach.   (b) PRIVATUS ($\alpha = 2$; $\beta = 1$).

Information leakage when $K = 20u$ and $m = 1$.

$$\bar{X}_{(n,m)}=[X(n-m+1),X(n-m),\ldots,X(n)] \qquad \bar{Y}^s_{(n,m)}=[Y(n-m+1+s),Y(n-m+s),\ldots,Y(n+s)]$$

- Information leakage is the highest when s = 1, i.e., X(n-1) and Y (n) has the highest dependency in our solution approaches.
  - This is due to our solution's inherent nature that Y (n) is chosen to change the current battery state resulting mainly from X(n-1).
- The worst-case information leakage in the advanced approach occurs around the price zone boundaries.
  - This is because around the price zone boundaries, there is no penalty defined and thus the battery state has a relatively higher chance to remain constant, which again makes it more likely that Y (n) tries to compensate for X(n-1).

Effects of sequence length and capacity



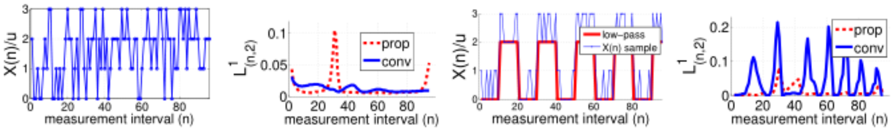
(a) Information leakage according to m ($s = 1$; $K = 20$).

(b) Uncertainty of $\bar{X}_{(n,m)}$ ($K = 20$).

(c) Information leakage according to K.

- Adversary has no advantage in observing a longer sequence in the draw process.
  - x-bit uncertainty can be understood in such a way that approximately the use process sequence has 2^x possible realizations with an equal probability $1/2^x$.
  - As m increases, there is a minor increment in percentagewise uncertainty reduction, while the uncertainty of the use process sequence increases significantly.
  - m=3: use-process uncertainty = 5.3 bits; reduction 11% at worst; $2^{5.3(1-0.11)}$=26.3 possible sequences
  - m=4: use-process uncertainty = 7 bits; reduction 17% at worst; $2^{7(1-0.17)}$=56.1 possible sequences
- When the battery capacity is too small, information leakage may be significant.
  - Once the battery capacity is above a threshold, further increasing the battery capacity leads to little benefit in terms of further reducing the information leakage.

Comparison (1/2)



(a) $X(n)$ w/o a significant low-pass component.

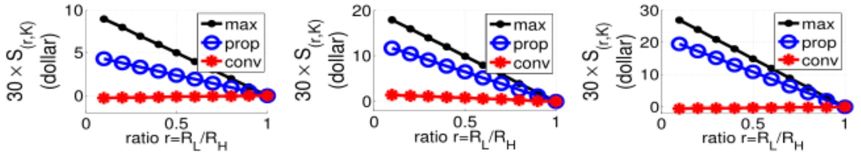
(b) Information leakage for (a).

(c) $X(n)$ w/ a significant low-pass component.

(d) Information leakage for (c).

- Kalogridis' scheme (published in Smart Grid Comm. 2010, 'conv' in the figure) performs a simple low-pass filtering over the use process in a best-effort manner, using a battery.
  - Without considering the energy cost factor.
- Thus, it reduces the high frequency variations in the resulting draw process.
  - Still allows the low-pass component of load profile to be revealed.
- If there is no significant low-pass component in X(n), Privatus performs slightly better than Kalogridis' to keep the privacy information, except at the price zone boundaries.
- If there is a significant low-pass component in X(n), Privatus will provide much better privacy protection than Kalogridis'.

Comparison (2/2)



(a) $K = 10u$ (2.15kWh).

(b) $K = 20u$ (4.3kWh).

(c) $K = 30u$ (6.43kWh).

- The unit energy u = 0.2143kWh and R_H = \$0.033/u = \$0.155/kWh.
- The average daily usage is 30kWh (U.S. residential customer).
- A typical home can achieve about \$16 saving for a month with a 6.43kWh battery, based on the following tariff example: R_L = \$0.04/kWh and R_H = \$0.15/kWh

Summary of Part 1

- In order to resolve the privacy issue in smart grid, we proposed PRIVATUS.
- PRIVATUS uses a rechargeable battery to make the meter reading reported to the utilities look different from the actual usage.
- PRIVATUS is also geared to the future of time-of-use pricing of electricity and it ensures that the battery is charged to achieve the maximal savings in the energy cost.

Part 2: long-term window (week)
Are you home or not?

Per-Day Energy Usage Flattening

- So far, we have seen that PRIVATUS hides the energy consumption pattern within a day.
  - The short-term part of PRIVATUS does not change the total amount of energy consumption.
- However, the total usage per day may be different across days, and this information can still be revealed to the adversary (by which the adversary may know whether you are home or not for a given day).
- PRIVATUS handles this issue by flattening the energy use across days.

- Assume that the average energy consumption per day varies in a cycle of P
  - P = 7 implies that a regular pattern of living is repeated every week.
  - Weekday vs. weekend
- We categorize the days into two types
  - Type 1 days: total amount of usage per day is less than average
  - Type 2 days: total amount of usage per day is more than average

Notations

- P: the period of the long-term pattern.
- U(d): the average amount of energy consumption for the d-th day of the period.
- The index set of days → $I_d = \{1, 2, \ldots, P\}$

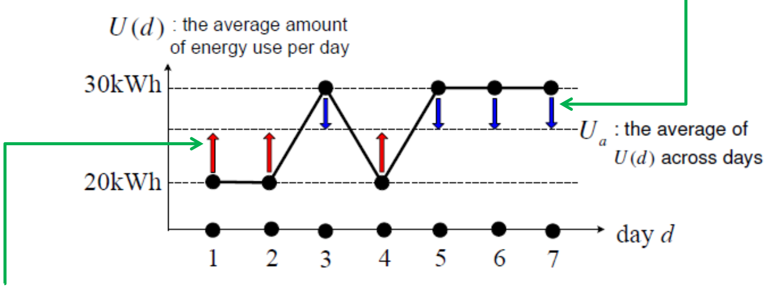
- U_a: the average of U(d) across days.

$$U_a = \frac{1}{P} \sum_{d \in I_d} U(d)$$

- The index set of type 1 days → $I_d^1 = \{d \in I_d : U(d) < U_a\}$

PRIVATUS's Approach

In type 2 days, PRIVATUS consumes less energy than U(d) by using the energy kept in type 1 days. It charges less energy than used in the high-price zone: the gap is supplied by the energy kept in type 1 days.



In type 1 days, PRIVATUS consumes more energy than U(d) by charging more energy in the low-price zone than used in the high-price zone, and by keeping the unused energy in the battery.

PRIVATUS's Approach

- The per-day usage flattening does not change the randomization framework of the short-term window.
  - Just changes the initial value of the actual battery state in a price zone → changes the amount of energy that is used or charged per day.
- Why flattening?
  - It requires smaller extra capacity for a battery compared to the randomization.
  - Minimum to maximum vs. minimum to average

Virtual Battery State

- In order to flatten the energy usage across days, we apply the virtual battery state $B_v(n)$ to the Bellman equation in the place of the actual battery state B(n).

$$\boxed{\begin{aligned} &J(S(n_L+1)) = 0,\\ &J(S(n)) = \max_{P_Y(n)\in\mathcal{P}^*_{B(n-1)}} E\left(D_e(n) + J(S(n+1))|S(n)\right)\\ &\quad S(n) = [n, \underline{B(n-1)}, B_e(n-1), P_Y(n-1)] \end{aligned}}$$

B(n) is replaced with $B_v(n)$

- The virtual battery state $B_v(n)$ is defined as follows:

$$B_v(n) = B(n) - E_k$$

the amount of energy that is kept for future use

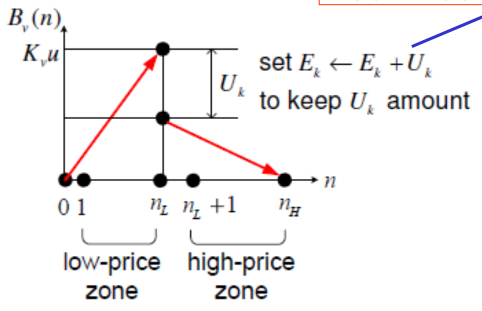
$$0 \le B_v(n) \le \underline{K_v u}$$

the virtual battery capacity:
determines the size of a decision table

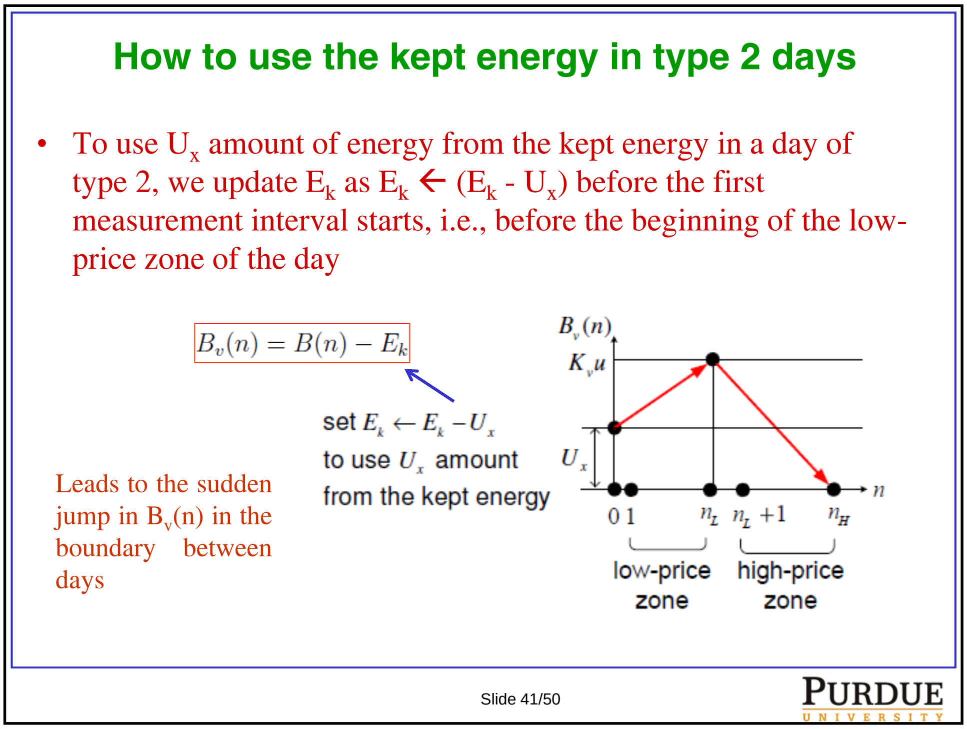
How to keep the energy in type 1 days

- To keep U_k amount of energy in a day of type 1, we update E_k as $E_k \leftarrow (E_k + U_k)$ before the (n_L+1)-th measurement interval starts, i.e., before the high-price zone begins.

$$\boxed{B_v(n) = B(n) - E_k}$$



Results in the sudden drop in $B_v(n)$ in the boundary between the low-price and high-price zones.



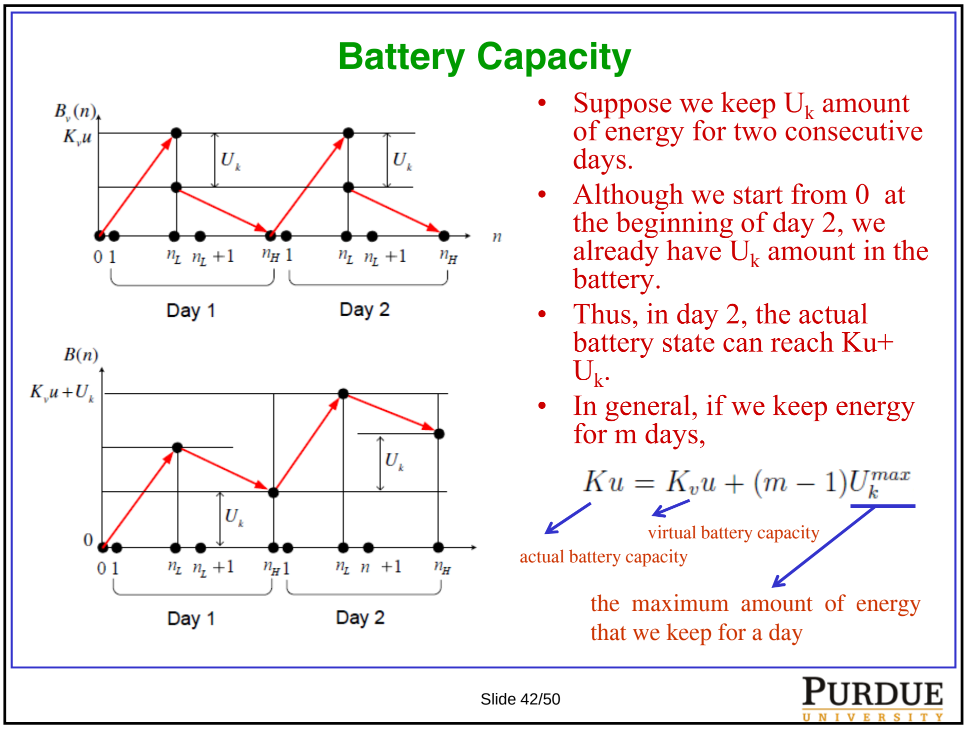
How to use the kept energy in type 2 days

- To use U_x amount of energy from the kept energy in a day of type 2, we update E_k as $E_k \leftarrow (E_k - U_x)$ before the first measurement interval starts, i.e., before the beginning of the low-price zone of the day

$$\boxed{B_v(n) = B(n) - E_k}$$

set $E_k \leftarrow E_k - U_x$ to use U_x amount from the kept energy

Leads to the sudden jump in $B_v(n)$ in the boundary between days



Battery Capacity

- Suppose we keep U_k amount of energy for two consecutive days.
- Although we start from 0 at the beginning of day 2, we already have U_k amount in the battery.
- Thus, in day 2, the actual battery state can reach $Ku + U_k$.
- In general, if we keep energy for m days,

$$Ku = K_v u + (m-1)U_k^{max}$$

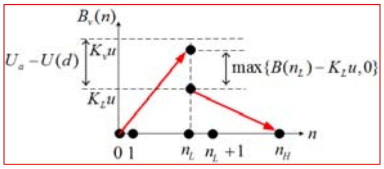
actual battery capacity

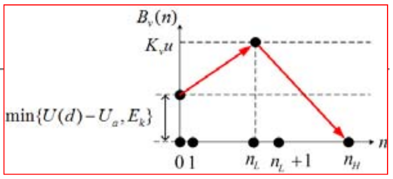
virtual battery capacity

the maximum amount of energy that we keep for a day

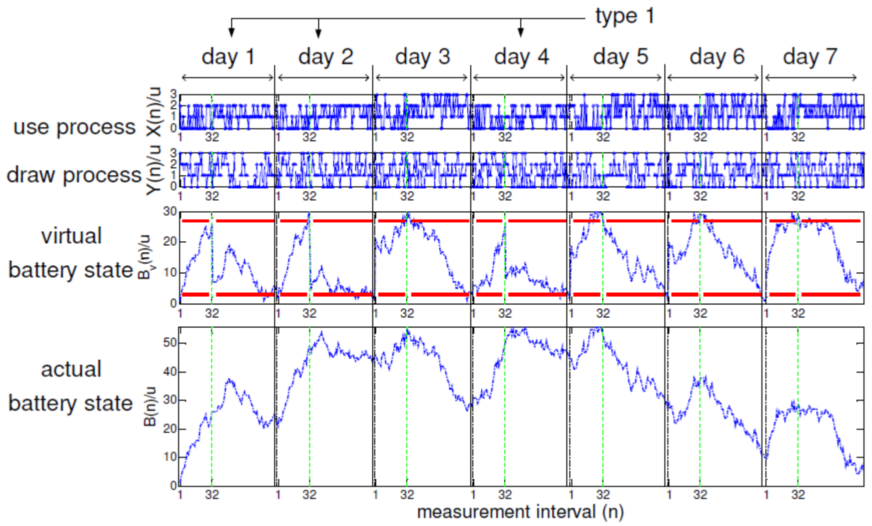
Algorithm 1 Energy Usage Flattening Across Days

```
for $d = 1$ to $P$ do
    if $d \in I_d^1$ then
        $K_L = K_v - (U_a - U(d))/u$
        $U_k = \max\{(B(n_L) - K_L u), 0\}$
        set $E_k \leftarrow (E_k + U_k)$ before the $(n+1)$-th measurement interval starts
    else
        $U_x = \min\{(U(d) - U_a), E_k\}$
        set $E_k \leftarrow (E_k - U_x)$ before the first measurement interval starts
    end if
end for
```

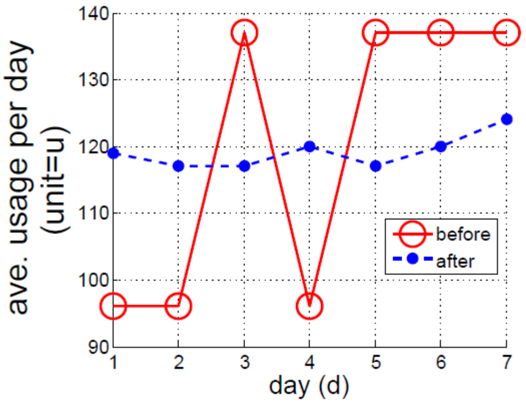




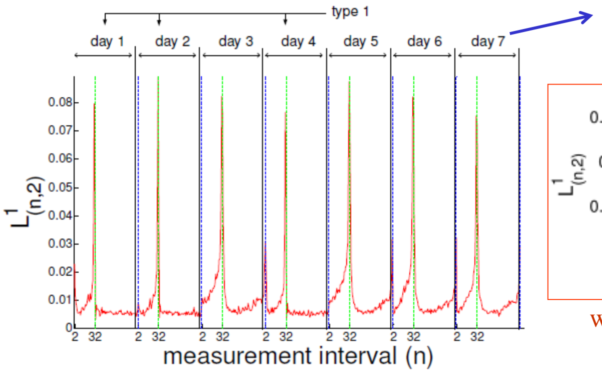
Simulation Results for Per-day Usage Flattening



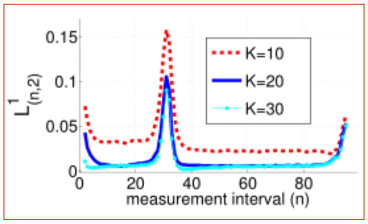
Change of Per-day Average Usage



Effects of Per-day Usage Flattening



Decision table with K_v=30

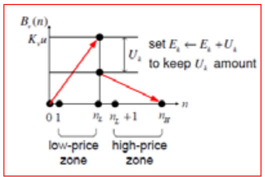


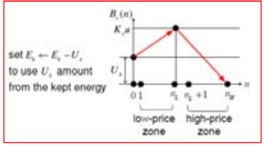
w/o the per-day flattening

- There is no significant difference in the information leakage across days.
- Per-day usage flattening of PRIVATUS does not change the privacy protection performance significantly at a similar condition (K=30).

Influence on Cost Saving (1/2)

- The maximum possible cost saving for a day is determined by the amount of energy that is charged in the low-price zone and then used in the high-price zone. Denote this amount of energy by U_s.
- Without the per-day usage flattening,
  - U_s = Ku for every day.
- With the per-day usage flattening, U_s becomes smaller than Ku, and varies according to the type of a day.
  - In a type 1 day, $U_s = K_v u - U_k$.
  - In a type 2 day, $U_s = K_v u$.
- When there are m days of type 1 within a P-day period, the per-day average of U_s is





$$\frac{1}{P}\left(m(K_v u - U_k^{max}) + (P-m)K_v u\right) \;=\; K_v u - \frac{m}{P}U_k^{max}$$

assuming $U_k = U_k^{max}$ for each day of type 1

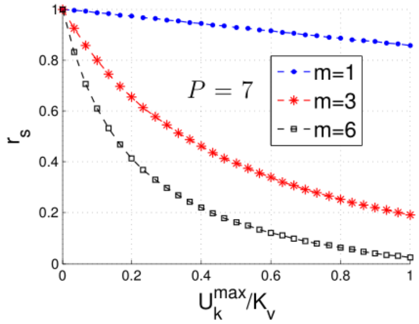
- The worst-case average for the maximum possible cost saving per day is

$$(R_H - R_L)(K_v u - \frac{m}{P}U_k^{max})/u$$

Influence on Cost Saving (2/2)

$$r_s = \frac{K_v u - \frac{m}{P}U_k^{max}}{K_v u + (m-1)U_k^{max}}$$

the ratio of the maximum cost saving w/ the per-day usage flattening to the maximum cost saving w/o the per-day usage flattening



- As a larger amount of energy is kept for future use, the cost saving is further reduced.

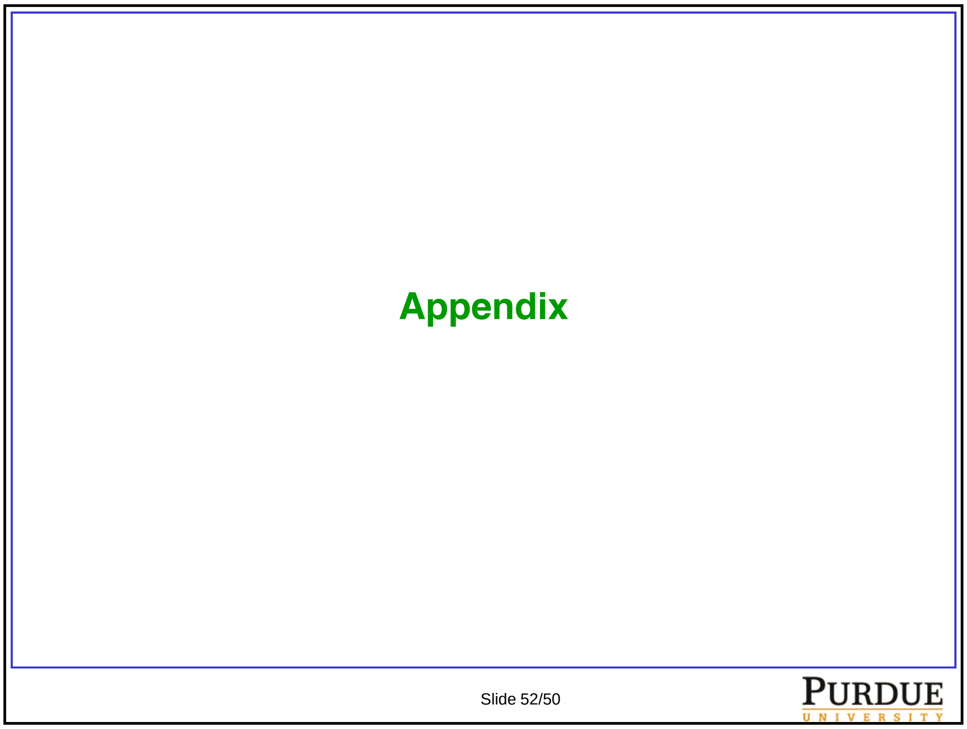
Summary of Part 2

- PRIVATUS can flatten the energy use across days in the average sense.
- The privacy-protection mechanism within a day window is the same as before.
  - Information leakage level remains similar.
- The flat per-day usage comes at the expense of a reduced cost saving.
  - Due to the amount of energy to be kept for future use, the cost saving is reduced.

Conclusion

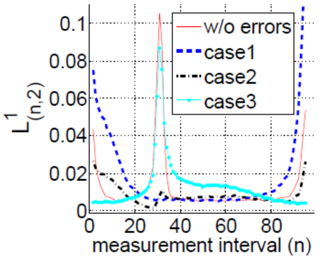
- PRIVATUS de-couples the meter readings and the actual user behaviors.
  - The meter readings reported to the utility are randomized, and also achieve the optimal cost saving.
- PRIVATUS can flatten per-day energy usage to hide whether you are home or not.
  - At the expense of reduced cost saving.

Thank you!

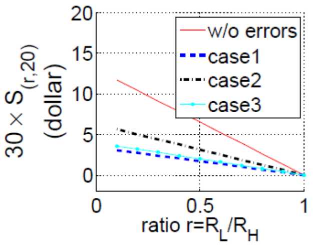


Appendix

Effects of the Estimation Error for $P_x(n)$



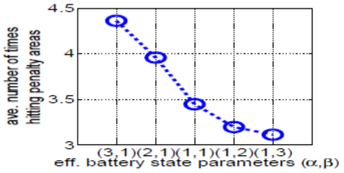
(a) Information leakage when $m = 2$, and $s = 1$.



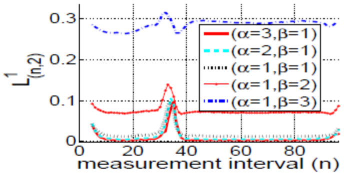
(b) Cost saving when u = 0.2143kWh, and $R_H = \$0.03/u$.

- Our estimation: $P_X(n)$ = [0.5, 0.2, 0.2, 0.1] in the low-price zone and P_X (n) = [0.1, 0.3, 0.4, 0.2] in the high-price zone
- However, X(n) is generated by different distributions P_X (n) = [0.1, 0.2, 0.3, 0.4] (`case1'), $P_X$ (n) = [0.25, 0.25, 0.25, 0.25] (`case2'), and P_X(n) = [0.4, 0.3, 0.2, 0.1] (`case3').
- Although there exists an estimation error in P_X(n), it does not affect the information leakage much.
- The estimation error influences the cost saving more significantly: the magnitude of the saving goes down with the estimation error.

Effects of Different Values of α and β



(a) The number of times hitting the penalty areas in a day according to α and β.



(b) Information leakage according to α and β.

- When the ratio of α to β goes down, the frequency to hit the penalty areas also decreases.
- We see a negative effect in terms of information leakage, when the ratio of α to β is too low.
  - In that case, the actual battery state wants to stay in the middle of the two penalty area thresholds T_H and T_L to avoid getting a penalty score.
  - This makes the compensation effect larger.