S2A: Secure Smart Household Appliances

Yuxin Chen
Department of Computer Science
Swiss Federal Institute of Technology (ETHZ)
Zurich, Switzerland
yuxin.chen@inf.ethz.ch

Bo Luo
Department of EECS
The University of Kansas
Lawrence, KS, USA
bluo@ku.edu

ABSTRACT

Security protection is an integral component for smart homes; however, smart appliances security has received little attention in the research community. Household appliances become very vulnerable if we introduce smart functions without proper security protection. In particular, smart access functions enable users to operate devices remotely. Meanwhile, smart devices are are also designed to support residential demand response, i.e. postpone non-urgent tasks to non-peak hours. However, remote adversaries could utilize such functions to manipulate smart appliances' operations without physically touching them. Such interferences, if not properly handled, could damage the smart devices, disturb owners' life or even harm the households' physical security.

In this paper, we present S2A, a security protection solution to be embedded in smart appliances. First, a SUP model is developed to quantify penalties from device security, usability and electricity price. We employ multi-criteria reinforcement learning to integrate the three factors to determine an optimal operation strategy. Next, to leverage the risk of forged control commands or pricing data, we present a realtime assessment mechanism based on Bayesian inference. Risk indices are further integrated into the SUP model to serve as weighting factors of corresponding decision criteria. Evaluation shows that S2A ensures appliances security while providing good usability and economical efficiency.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information systems**]: Security and Protection—*Physical Security*

General Terms

Security, Design

1. INTRODUCTION

As the next-generation standard for power generation and transmission, advanced computation and telecommunication capabilities are introduced into smart devices to constitute a large-scale smart grid network, and to support "smart" functions, such as large sale load balancing, dynamic pricing, smart consumption. Unfortunately, most of the advanced functions, especially those on the power consumption side (i.e., smart home side), are not yet implemented in the pilot projects. Security concern is one of the major obstacles that prevent broad industrial adoption of such smart functions.

Smart appliances are envisioned to receive control commands and electricity prices from the network. Embedded control systems have been installed in household appliances. Manufactures are starting to build appliances with remote access functions (a.k.a. smart access). For instance, LG products with THINQ technology were demonstrated at CES 2011. General Electronics (GE) has been working with Tendril to connect GE household products over Zigbee wireless networks. Such smart access capabilities enable owners to remotely monitor and operate their devices using phones, tablets, or through designated websites. On the other hand, smart meters are designed to receive realtime electricity pricing (RTP) and pass on to household devices [14], which optimizes energy consumptions based on RTP [34]. However, smart appliances are not yet equipped with *smart security protection mechanisms* to defend against cyber attacks. For instance, they follow remote control commands without verifying the authenticity of such commands. In this context, if we introduce "smart" functions to electrical appliances without proper security protection, they become more vulnerable than conventional devices. Adversaries could manipulate or intervene smart appliances' operations remotely, without physically touching them. More severely, when compromised devices are set to work in abnormal conditions for an extended period of time, they could be physically damaged, and even compromise environmental safety. For instance, overheating electric motors are shown to be a root cause of insulation failures, which is very dangerous to the users. In this paper, instead of focusing on the traditional security notions of *confidentiality*, *integrity* and *availability*, we focus on the *operational or physical safety* of smart devices. Therefore, the security goal of the S2A approach is to ensure the physical safety of the smart devices, preventing them from working in abnormal conditions, when the smart control environment becomes unreliable.

In this paper, we present S2A (secure smart appliances), a security protection mechanism for smart appliances. S2A is an embedded software solution, which employs machine learning technologies to provide smart and flexible protections for smart household appliances. First, for an individual smart

CODASPY'12, February 7–9, 2012, San Antonio, Texas, USA.
Copyright 2012 ACM 978-1-4503-1091-8/12/02 ...$10.00.

appliance, the S2A models heterogeneous notions of device security (S), usability (U), and electricity pricing (P) into homogeneous benefit (or penalty) functions. We then employ multi-criteria reinforcement learning (MCRL) to integrate all three factors to determine the optimal operation strategy, which aims to maximize usability and minimize both security penalties and electricity costs. Next, to leverage the risk of fake control commands or forged pricing data, we propose a real-time risk assessment and re-weighting mechanism. We invoke Bayesian inference approaches to evaluate the trustworthiness of input from each channel, and adjust the parameters for MCRL criteria accordingly. Through security analysis and simulation results, we show that our solution ensures appliance security, while maintaining usability and economical efficiency of power consumption.

Our contributions are: (1) we introduce a comprehensive security protection for smart household devices. Our solution integrates usability, electricity pricing, and device security to maximize the overall benefit (or minimize overall penalty). (2) By employing machine learning methods, S2A provides an effective and reliable security protection. Moreover, compared with the conventional security notion, which is black-or-white, S2A seamlessly integrates risk assessment into decision algorithms, without making a verdict of "safe" or "under attack". (3) We propose a flexible approach, in which degree of protection and quality of service is based on resources (e.g. historical data) and capabilities.

2. RELATED WORK

Smart grids are envisioned as the next generation power system [50, 16, 40, 48]. Some vision/introductory papers can be found at [25, 11, 35, 6]. Existing research projects mostly focus on the "power grid" side (i.e. the macro grid), for example, large scale dynamic load balancing, reliability and recovery, power market [45, 24, 38, 32]. On the other end of the spectrum, smart meters are being implemented [22], and smart meter communication systems are being deployed [41, 42, 1, 39]. Meanwhile, smart appliances are proposed to improve user experience and cost efficiency: realtime retail pricing (RTP) introduces dynamically changing electricity prices that reflect the realtime supply-vs-demand trend [4, 5]. RTP is delivered to smart meters and then household appliances. With the built-in intelligence, smart appliances could move non-urgent tasks to off-peak hours to enhance economic efficiency of power usage [10]. Recently, [34] introduces a reinforcement learning approach to identify a relatively optimal time to start tasks. Tasks from the queue are picked to execute based on RTP and the length of wait. On the other hand, systems have been designed to enable remote monitor and control for household appliances through smart meters [47, 20, 36, 46].

Security and privacy protection is an important and challenging component in smart grids [26, 19]. A comprehensive survey is provided at [2]. In particular, [30, 31] studied the security requirements in the overall smart grid framework and presented security technologies to fulfill such requirements. [44] presents a conceptual framework to protect power grid automation systems. [7] points out security requirements and threats related to smart meters. [3] analyzes external intrusions, and introduces specification-based detection approach as a potential solution. [27, 28] show that adversaries could attack the advanced metering infrastructure to manipulate power usage for energy theft.

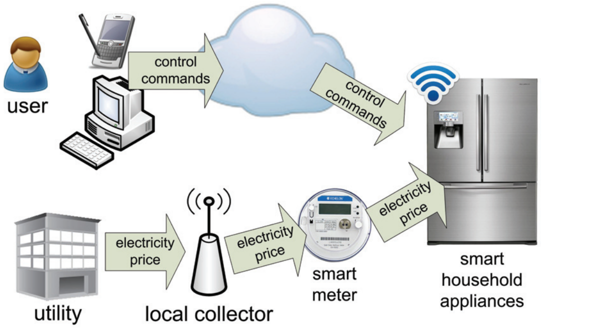

Figure 1: Smart appliances receive control commands and realtime electricity pricing from remote.

Most of the above-mentioned security protection approaches focus on the power grid, from generators to distributers to smart meters. Meanwhile, security issues related to household appliances have been lightly studied in the context of ubiquitous computing and home-area networks [23, 33]. They mostly concern about wireless communication security, authentication and privacy issues. For instance, [21] shows that in-home activities could be inferred from realtime energy consumption data. However, to our best knowledge, there has not been any work on protecting appliances' physical security, especially at the presence of untrustworthy external inputs (control commands and prices).

3. PROBLEM AND SOLUTION OVERVIEW

3.1 Smart Household Devices

Smart devices receive users' control commands and realtime pricing (RTP) from the network. As shown in Figure 1, utility distribution companies broadcast realtime electricity prices to households. Various proposals have been suggested in the literature. The more popular approach is to employ wired communication from utility companies to neighborhood collector devices, and wireless communications (e.g. wireless mesh) to further deliver to smart meters. Smart meters then send RTP information to compatible smart appliances via home-area WiFi. Meanwhile, manufactures such as LG and GE are starting to introduce remote control functions to smart appliances. In their design, users send control commands via a designated website or a mobile app. The commands go though the Internet to be delivered to the appliances, which connect to the Internet through household WiFi. There are also proposals that such commands could be delivered via smart meters.

3.2 The Threat Model

In a large scale open platform with many stakeholders, from the viewpoint of a smart device, it cannot assume absolute security of all the external peer(s). When adversaries penetrate into the control systems or temper with the communication channel, they could inject forged inputs (control commands and/or RTP data) into household smart devices, who may not be able to verify the authenticity and validity of such inputs. The interferences, if not properly handled, could affect the owner's regular lifestyle, or even cause serious physical damages. Let us look at some examples:

Example 1: Electric vehicles (EV) are designed to optimize the economical efficiency of power consumption, i.e., charge the battery when the electricity price is low, and (optionally) provide power to the household or the grid when the price is high (a.k.a. vehicle-to-grid [17, 18]). An intruder may send forged pricing data to trick EV to operate improperly to cause financial losses to the owner, to affect the load balancing of the power grid, or even mess up with the grid to achieve financial advantages. □

Example 2: Battery life of electric vehicles heavily rely on proper use and maintenance. An intruder may send forged fluctuant pricing to trick EV battery to constantly switch between charge and discharge for a relatively long period (e.g. start and stop charging 10 times per hour for 10 hours). This attack will seriously damage the battery, and even cause hazardous conditions when the battery gets overheated. □

Example 3: An adversary could penetrate into the remote control systems or home-area networks to obtain control of household appliances. Such interference could affect the owner's regular lifestyle, or even cause serious physical damages. When an adversary sets all the exothermic devices in a household to maximum heat level simultaneously, the room temperature rises significantly. More dangerously, the circuit gets overloaded, and the risk of fire increases. □

In this paper, we consider the situation where *smart devices (excluding smart meters) receive potentially harmful inputs from ostensibly legitimate sources.* We do not consider smart meters, since they are usually located outside of the household, and they are physically insecure. On the other hand, a mal-functioning smart meter will not directly threat household safety. In our settings, each smart appliance in the residence functions as an agent that has an embedded control unit to manage its own operations. We assume that smart devices are physically secure since they are usually located inside the household. We also assume that the embedded control systems are not compromised: the control logic is relatively less complicate; they only receive limited information (control and price) from designated sources; software updates usually require physically touching the device (e.g. using a USB drive). Therefore, it is not easy to hack into the kernels of the smart devices remotely.

The goal of the paper is to protect the operational security of household smart devices in the presence of potentially harmful inputs from information (and command) distribution channels. We also aim to maintain usability (QoS) and economical efficiency. In particular, we study two channels that may take suspicious inputs: *user control commands (UCC)* and *realtime pricing (RTP)*. Meanwhile, based on the duration and frequency of suspicious inputs, we consider two types of threats: *Threat 1. sporadic incidents* and *Threat 2. continuous attacks.* Continuous active attacks are potentially more damaging, and may not be handled by existing rule-based security protection mechanisms.

Please note that smart meters have essentially different capabilities and functionalities than household appliances. Our threat models and countermeasures are not applicable on smart meters. Some related works on smart meter security are summarized in Section 2.

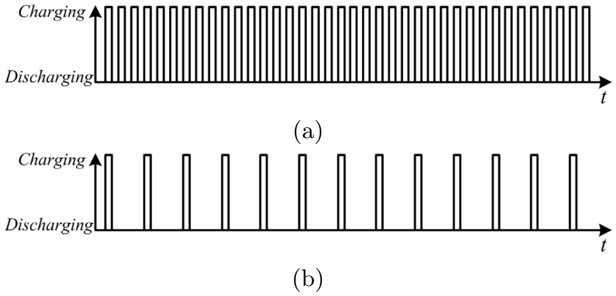

Figure 2: Battery charging system under active attack: (a) forged control commands that rapidly switch between charging and discharging; (b) chargers operations with rule-based protection.

3.3 Rule-based Security Protection

At present, most of the household appliances, including smart devices, are equipped with embedded security protection mechanisms that are usually rule-based. For instance, when an air conditioner is switched off, its internal security protection mechanism will keep it off for n minutes before it could be restarted. Similarly, when a smart car stops charging, it will mandatorily wait for m minutes to avoid immediate recharging to protect the battery. Some devices use sensors to obtain status information, and security rules are based on sensor inputs. For example: an electronic motor should stop for n minutes when the motor temperature is higher than x degrees. However, the rules are mostly designed to protect the device against users' misuse. They provide minimum protection, and do not consider future consequences. In particular, they can hardly protect the devices against active attacks, especially continuous attacks.

Example 4: Figure 2 gives an example of a battery charging system under active attack. Aimed to damage the battery, the attacker sends forged control commands that rapidly switch between charging and discharging. The embedded rule-based protection mechanism enforces an interval of t minutes between two charges, to protect the battery against transient power line faults. As shown in Figure 2 (b), for continuous attacks, such protection mechanism will only increase charging interval to t. However, without more complicate security protection mechanism, the battery is still damaged after an extended period of time. □

3.4 Solution Overview

In this paper, we propose the S2A framework, as an embedded software solution to protect operational security of smart household appliances against misuses and forged inputs. The goals are: (1) ensure appliances' security, (2) maintain usability, and (3) reduce energy costs. Typically, smart appliances need to make appropriate tradeoffs between ensuring usability (e.g. user wants to start the dishwasher) and minimizing energy cost (e.g. smart dishwasher wants to wait for low electricity price). Such requirements usually lead to a complicate optimization problem, which is difficult, if not impossible, to solve with rule-based methods.

We propose a two-phase solution, which enables smart devices to learn to protect themselves, without requesting any support from "supernodes" or smart meters. In the first phase, we aim to achieve an optimized and fine-grained operational strategy. The SUP model considers security penalties, usability penalties, as well as economical benefits. In the second phase, we assess the trustworthiness of each input channel by comparing the instant input with historical data. Since user commands and RTP demonstrate very different patterns in the regular working conditions, different intrusion detection mechanisms are invoked accordingly. We do not provide a verdict of "safe" or "under attack (forged input)". Instead, the security assessments are seamlessly feedback to the SUP model as weight factors of the corresponding penalty (or benefit) functions.

4. THE S2A FRAMEWORK

4.1 Overview of the S2A Framework.

Figure 3 demonstrates the S2A framework. As shown, our solution constitutes two major components (tiers): the SUP module and the realtime risk assessment module.

Tier 1: The SUP model. Tier 1 considers the basic scenario of S2A framework, in which an appliance is an independent device without any knowledge to external historical information or environmental information. Note that we assume a short operation log is available, which records a queue of user requests to use the device, and recent history of on-off operations. In the basic S2A solution, we define a SUP model to capture security, usability and electricity price. In SUP, a security function $s(t)$ is defined to model the operational penalty for the physical safety of smart electrical devices; a usability penalty function $u(t)$ is defined to model the frustration of users (similar to [34]) when they wait for the delayed operations; and finally real-time electricity price is received by smart pricing $p(t)$. When a user requests a S2A-enabled device to operate, SUP balances all three penalties to make a smart operation plan, so that: the device always works in a safe working mode; the user will not be very unhappy because of long wait; and the total cost of electricity to complete the task is relatively low. In our solution, we employ multi-criteria reinforcement learning (MCRL) to make real-time operational decisions based on three criteria: $s(t)$, $p(t)$, and $p(t)$.

Tier 2: Real-time risk assessment for SUP. In the second tier of the S2A framework, we consider the trustworthiness of the user requests and the electricity pricing information. To protect smart appliances in the presence of suspicious control commands or price data, we use Bayesian inference (RRA-RTP and RRA-UCC in Figure 2) to assess the credibility of the inputs, i.e. the likelihood of tampered control commands or forged electricity prices. Note that we only evaluate the validity of remote data, not physical operations on the device (e.g. pushing a button on the washer is always considered to be a valid control command). The Bayesian inference modules takes current inputs to compare with historical data, and generates two risk indexes R_p and R_u, which measure the trustworthiness of the control commands and electricity prices, respectively. Unlike conventional intrusion detection approaches, we do not provide a verdict on whether the system is under attack or not. Instead, the risk factors are seamlessly integrated into SUP. For instance, the risk index for user command (R_U) is sent back to the SUP model to serve as weighting factor for the usability penalty. In this way, when forged inputs are detected at tier 2, its risk factor increases, and the corresponding weight factor for the suspicious input channel decreases, to fade out the suspicious input.

Override Rules. To improve user experience and to give users better control, especially in unusual circumstances, the following override rules are enabled in S2A.

(1). In S2A, the user may force the task to be conducted without any delay, i.e. force usability functions to override smart-pricing functions. As a reference, in [34], user could press "start" button twice to instantly start the operation, without waiting for low electricity price. However, security penalty is still in place to ensure device security.

(2). For security purposes, we assume that the device could be turned off at anytime. That is, there is no security penalty if the user intends to turn the devices off.

(3). When users request legitimate but unusual operations from remote, the operation could appear to be highly suspicious to the realtime risk analysis module. To prevent any legitimate requests from being denied or deferred, we introduce an additional task verification process, which is independent from the routine verification. Risk assessment could be overridden by additional validation, so that critical (and irregular) task will not be delayed. Technically, we enforce an extra authentication to verify the identity of the requestor. For verified tasks, we increase the weight for usability and decrease the weight for smart pricing. Once again, security penalty is still in place.

4.2 The SUP Model

Overview. The core of the S2A framework is an SUP model. For a smart appliance without long time memory, we first identify its operational states Ω (i.e., the total reward/penalty gained by leaving the previous states), and model state transitions as a set of actions A. For simplicity of description, we only consider the case that appliances are either ON or OFF. Hence, we have four types of actions: $A = \{\langle \text{OFF}\rightarrow\text{OFF}\rangle, \langle \text{OFF}\rightarrow\text{ON}\rangle, \langle \text{ON}\rightarrow\text{ON}\rangle, \langle \text{ON}\rightarrow\text{OFF}\rangle\}$. We model the process as a multi-objective Markov decision process (MMDP) that considers the following three criteria, and define penalty functions for each action w.r.t. each criterion. Note that we can easily add more modes (e.g., use four modes: high, mid, low, off) by adding penalty functions. Our learning algorithms takes general MMDP, with no restrictions on number of states or actions.

Security Criterion. A *security penalty function* $s(A, t) \in R^+$ is defined to denote the penalty of performing action A at time t. At high level, security penalty $s()$ quantifies the potential of damaging the device (e.g. overheat the battery) and/or harming the environment (e.g. burn down the house). We only enforce penalties for turning on the device ($\langle \text{OFF}\rightarrow\text{ON}\rangle$) or keeping on the device ($\langle \text{ON}\rightarrow\text{ON}\rangle$). $s(\langle \text{ON} \rightarrow \text{ON}\rangle, t)$ and $s(\langle \text{OFF} \rightarrow \text{ON}\rangle, t)$ cannot coexist since the current state is either ON or OFF. For simplicity of description, we use $s(t)$ when there is no confusion. A larger $s(t)$ indicates that the current working condition is not desirable, and pushes the decision against turning or keeping the device on.

The parameters of the penalty function are defined by the manufacture of the appliance, based on the operational and security characteristics of the device. In our model, each device is equipped a built-in function generator $G_s(oper)$,

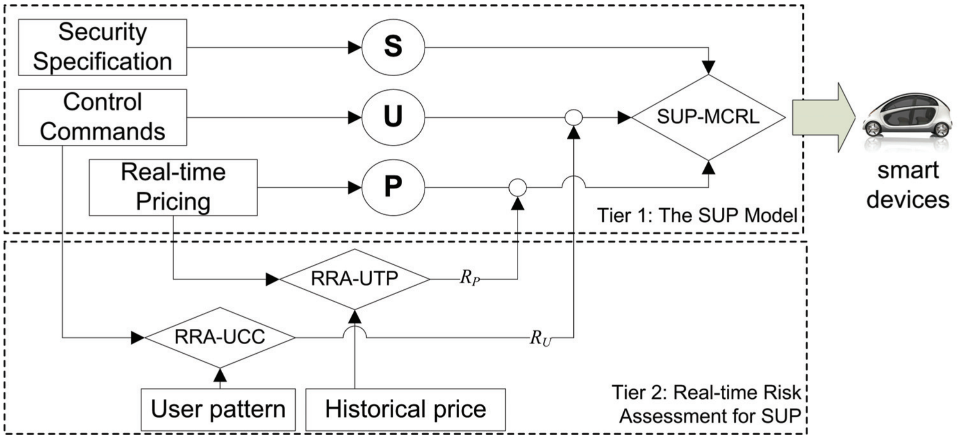

Figure 3: The S2A Framework: SUP-MCRL: the security-usability-pricing model with multi-criteria reinforcement learning; RRA-RTP: realtime risk analysis on realtime electricity pricing; RRA-UCC: realtime risk analysis on users' control commands; R_P and R_U: risk factors.

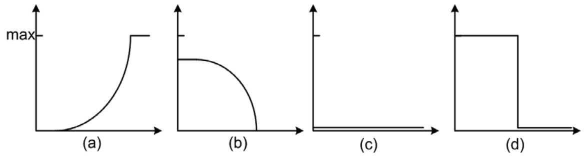

Figure 4: Examples of security penalty functions

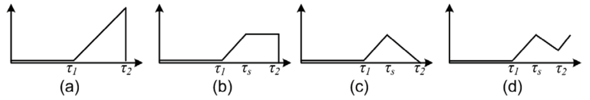

Figure 5: Examples of usability penalty functions

which constructs penalty functions based on pre-defined rules, device states and recent operations. $G_s(oper)$ is triggered to refresh $s(t)$ whenever an operation is performed (i.e. at state change: $\langle$OFF$\rightarrow$ON$rlangle$ or $\langle$ON$\rightarrow$OFF$\rangle$). For "smarter" appliances, the security penalty is generated on-the-fly from sensor inputs (e.g. heat, environmental temperature, etc). When the security penalty reaches MAX, it cannot be surpassed by other penalty functions – the device should remain off until security penalty drops.

Example 5: Some devices cannot operate for more than a pre-defined period of time – they need to stop and cool down. When the appliance is first switched on, $G_s(oper)$ generates a security penalty function for keeping the device on ($\langle$ON$\rightarrow$ON$\rangle$). In this case, $s(t)$ demonstrates an increasing pattern (Figure 4 (a)). When it is switched off before reaching maximum penalty, $G_s(oper)$ refreshes $s(t)$ to $s(t, \langle$OFF$\rightarrow$ON$\rangle)$, which requires the device to keep off for a while, and then starts to decrease (Figure 4 (b)). On the other hand, some devices (e.g. batteries) cannot switch between on and off frequently. There could be no security penalty for keep charging (Figure 4 (c)), but the penalty function for $\langle$OFF$\rightarrow$ON$\rangle$ reaches maximum value once the device is turned off, hence preventing it from being switched on until a waiting period (Figure 4 (d)). □

Usability Criterion. A smart appliance receives user request $c(t) \in R^+$ indicating his/her desire to run the devices at time t. Such a control command, however, does not necessarily start the device instantly, but rather specifies a reservation with the S2A system to run the device at an optimal (possibly later) time to balance user utility with other factors, such as economical efficiency and system security. With the user reservation at time t_0, a certain quantity $c(t_0) = e_0$ of electricity is requested for the operation, otherwise $c(t_0) = 0$, indicating that there is no reserved energy use at t_0. User reservations are stored in a FIFO *pending energy queue* $\mathbf{q}_i = \{\langle t_0, e_0\rangle, \langle t_1, e_1\rangle, \ldots\}$.

In the SUP model, we capture *usability penalty* with a penalty function $u(A, t) \in R^+$, which denotes the expected usability penalty when we take action A at time t. When the requested task is delayed ($\langle$ON$\rightarrow$OFF$\rangle$ or $\langle$OFF$\rightarrow$OFF$\rangle$) due to high electricity price or active security protection, usability penalty ($u(t)$) starts to increase. Meanwhile, there is no usability penalty for turning on or keeping on the device. In practice, $u(t)$ cannot be detected on-the-fly, rather, it is calculated from a pre-defined usability penalty model, which is based on characteristics of the appliance's usage and user-centric analysis results. Note that $s(t)$ represents the appliance's perception (guess) of users' dissatisfaction. When the operation pauses at time t_0 ($\langle$ON$\rightarrow$OFF$\rangle$), the appliance immediately knows that task completion will be postponed. Hence, the penalty $s(t)$ starts to increase at t_0.

Example 6: Figure 5 shows some simple examples of usability penalty function $u(t)$. In Figure 5 (a), the task is paused at τ_1, where the penalty function for $\langle$OFF$\rightarrow$OFF$\rangle$ starts to increase linearly. In Figure 5 (b), the delayed task is restarted at time τ_s, the expected completion time stops changing. Hence, usability penalty (for $\langle$ON$\rightarrow$OFF$\rangle$) keeps static, until the task is completed at τ_2. Meanwhile, if the user is aware of the task progress, the dissatisfactory level could decrease when s/he knows that the task is in progress and is expected to finish soon (Figure 5(c)). Last, as shown in Figure 5 (d), user frustration may increase again when the task is paused at τ_2, before its completion. □

In S2A, the usability model is pre-built in the smart device by its manufacturer. $u(t)$ is generated when a new task

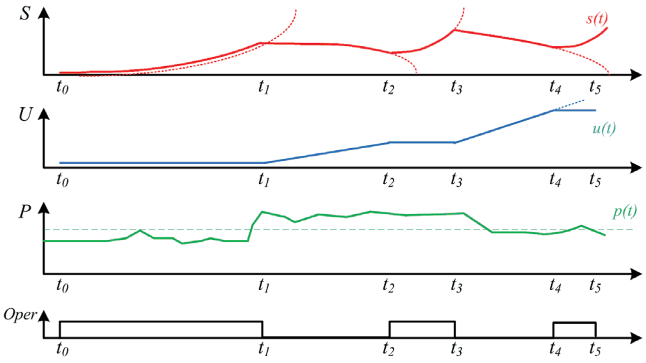

Figure 6: Example of a smart appliance operating under SUP model.

is picked from the task queue. It is refreshed when the operation of the appliance changes. In general, $u(t)$ increases (usually nonlinearly) for ⟨OFF → OFF⟩, and stays stable or decreases when the task is progressing (⟨ON → OFF⟩). The model also takes in the recent working history and environmental parameters, so that $u(t)$ is adjusted to users' everyday life. Different appliances will also have different patterns for usability penalty in different conditions. For instance, users are less concerned when a smart car is being charged at night; but s/he may want the task to complete soon if the car is plugged-in in the morning. In this paper, we model $u(t)$ as an abstract function. Usability and user behavior modeling problems are studied in the human factors research community. Usability in the context of dynamical electricity pricing has been studied in the context of residential demand response (RDS), e.g. [34, 15, 12].

Smart Pricing Criterion. Smart appliances are designed to receive realtime retail electricity prices (RTP) from the distributor. In the SUP model, realtime price is provided by a function $p(t) \in R^+$. Data from world-wide pilot projects have shown different patterns of electricity pricing. Most of them demonstrates a daily revolving pattern, which peaks in early evening, decreases later into the night, and increases in the morning. Currently, our model only considers electricity cost. With reasonable modifications, it could be expanded to include more complicate cost models, which consider costs from multiple sources.

The SUP Model. The SUP model integrates all the criteria described above to minimize three factors: security penalty, usability penalty, and total expense for the task. Before discussing the detailed learning algorithm, we show an intuitive example on how the SUP model works.

Example 7: As shown in Figure 6, the user submits a request at time t_0 to an S2A embedded device. Since the electricity price $p(t)$ is low, the appliance starts instantly (note that the dashed line in P plot represents average electricity price, not a decision threshold – there is no preset decision threshold for each penalty function). We assume that this device cannot continuously operate for a very long period of time. Security penalty starts to increase gradually. At t_1, there is a sharp raise of electricity price. Meanwhile, $u(t)$ is very low at t_1 – there has not been any delay until t_1. At t_1, the SUP model decides to pause the job. Starting from t_1, usability penalty starts to grow since the completion time is expected to be postponed (we use a linear function to model usability penalty in this example, however, real-world usability model is usually non-linear). Security penalty reduces as the device is off. At t_2, the SUP model decides to switch the device on, based on the increasing usability penalty and decreasing security penalty. At t_3, due to very high security penalty (e.g. the motor it very hot), the device is turned off again. The device cools down until t_4, when it is restarted to get the task done at t_6. □

5. THE ALGORITHMS

In this section, we describe the core algorithms to support smart protection in the S2A framework. First, we introduce multi-criteria reinforcement learning (MCRL) to determine the optimal operational strategy for the SUP model. Next, we introduce Bayesian-based realtime risk assessment, and seamlessly integrates risk indices into the SUP model.

5.1 MCRL for SUP

The core problem in the SUP model is to learn an optimal operational behavior for a smart appliance in the presence of dynamic preferences/penalties introduced by multiple objectives. In SUP, a device is an independent agent, which learns an approximately optimal strategy through trail and error interactions with the environmental variables. The *pending energy* is defined as the amount of energy that is required to finish the task. When the input power to the device is stationary, its pending energy is directly proportional to the remaining time to finish the task. The *environment* of a smart appliance is described by a deterministic multi-objective Markov decision process (MMDP) $\langle \Omega, A, f, \vec{\rho} \rangle$, where Ω is the finite set of discrete states, A is the set of actions, $f : \Omega \times A \to \Omega$ is the state transition function, and $\vec{\rho}$ is the vector-based penalty function $\vec{\rho} : \Omega \times A \to \mathbb{R}^n$. The state signal $x_k \in \Omega$ describes the environment at each discrete time-step k. In SUP, x_k encodes the device's current working status (i.e., whether the device is on or off), the current pending energy, the pricing information, the cumulative delay of the task, and the duration since the last operation (i.e., how long has the device been on or off), etc. The learning agent can alter the state at each time step by taking actions $a_k \in A$ of keeping on/off or turning off/on a device accordingly. As a result of the action a_k, the environment changes its state from x_k to $x_{k+1} \in \Omega$ according to the state transition rules given by $f : x_{k+1} = f(x_k, a_k)$. The agent then receives immediate vector-valued penalties of taking the action a_k on the basis of multiple evaluating objectives, which is completely determined by the current state and action: $\vec{\phi}_{k+1} = \vec{\rho}(x_k, a_k)$.

In the SUP settings, each of the penalty criterions is associated with a weight in accordance with its reliability. Given a weight vector $\vec{w} = (w_1, \ldots, w_n)$ and an MMDP, a new MDP with vector-valued penalty functions is created when multiplying each penalty $\rho_i(x, a)$ of type i with w_i. For a constant weight vector $\vec{w}$, the goal of the learning agent is to minimize the expected discounted penalty:

$$\Phi_k = E\{\sum_{j=0}^{\infty} \gamma^j \vec{\phi}_{k+j+1} \cdot \vec{w}\} \qquad (1)$$

where $\gamma \in [0, 1)$ is the discount factor. It can be regarded as encoding increasing uncertainty about the penalties that will be received in the future. Such discounted penalty compactly represents the penalty accumulated in the long run, and measures a policy's long-term performance.

For deterministic SUP models, the behavior of an agent is described by its policy $\pi : \Omega \rightarrow A$, which specifies how the agent chooses its actions given the state. The vector-based *action-value function*, $\vec{Q}^{\pi} : \Omega \times A \rightarrow \mathbb{R}^n$, is the expected return of a state-action pair given the policy π: $\vec{Q}^{\pi}(x,a) = E\{\sum_{j=0}^{\infty} \gamma^j \vec{\phi}_{k+j+1} .* \vec{w} | x_k = x, a_k = a, \pi\}$, and the optimal Q-function is defined as $\vec{Q}^{*}(x,a) = \min_{\pi} \vec{Q}^{\pi}(x,a)$. It satisfies the Bellman optimality equation

$$\vec{Q}^{*}(x,a) = \vec{\rho}(x,a) + \gamma \min_{a'} \vec{Q}^{*}(\pi(x,a),a'), \ \forall a' \in A \qquad (2)$$

where $a' = \mathrm{argmin}_{a'}[\vec{w} \cdot \vec{Q}^{*}(\pi(x,a),a')]$.

The formula is derived from the original Q-Learning[43], with vector-based representation of the immediate and expected discounted penalty function. The current estimate of $\vec{Q}^{*}$ is updated using estimated samples of the right-hand side of Equation 2. These samples are computed using actual experience with the task, in the form of weighted penalty vectors and pairs of subsequent states x_k, x_{k+1}:

$$\begin{aligned} &\vec{Q}_{k+1}(x_k,a_k) = \vec{Q}_k(x_k,a_k) + \\ &\alpha_k[\vec{\phi}_{k+1} + \gamma \vec{Q}_k(x_{k+1},a') - \vec{Q}_k(x_k,a_k)] \end{aligned} \qquad (3)$$

where $a' = \mathrm{argmin}_{a'}[\vec{w} \cdot \vec{Q}_k(x_{k+1},a')]$.

In variable-penalty settings, we employ an efficient variable-transfer algorithm derived from [29]. Since the immediate penalty at each time step is a linear combination of different penalty factors (e.g., usability penalty, electricity cost), and the Q-value function (long term penalty) is based on the sums of the immediate penalties, we can infer that the *expected discounted penalty* of policy π starting from state x: $\vec{w} \cdot \vec{Q}^{\pi}(x,a)$ is also linear in penalty weights.

In variable-penalty reinforcement learning, each weight vector corresponds to an individual Markov decision process. All the MDPs share the same transition dynamics (i.e., same states, same actions, same transition function, etc. One example is that delaying a task will always increase user frustration), but are linear in a set of penalty features. Thus, given a new weight vector $\vec{w}_{new}$ and a starting state x_k, one can approximate the optimal policy π_{new} for the new weight vector based on the already learned policy set $\mathcal{C}$, simply by selecting the one with minimum expected discounted penalty $\pi_{new} = \mathrm{argmin}_{\pi \in \mathcal{C}} Q^{\pi}(x_k,a')$, where $a' = \mathrm{argmin}_{a'}[\vec{w}_{new} \cdot \vec{Q}^{\pi}(x_k,a')]$.

SUP-MCRL is presented in Algorithm 1. In step 11, the agent tests all actions in all states with nonzero probability, which is an exploration-exploitation tradeoff problem. The agent uses the Boltzmann exploration strategy, which in state x selects action a with probability

$$Probability(x,a) = \frac{e^{1/(\tau \vec{w} \cdot \vec{Q}(x,a))}}{\sum_{a'} e^{1/(\tau \vec{w} \cdot \vec{Q}(x,a'))}} \qquad (4)$$

where $\tau > 0$ controls the randomness of the exploration. When $\tau \rightarrow 0$, this is equivalent with greedy action selection. When $\tau \rightarrow \infty$, actions are random. When $\tau \in (0, \infty)$, actions with lower penalties are more likely to be selected.

5.2 Real-time Risk Assessment (RRA) for SUP

The above model assumes that all the inputs are valid. However, the control commands and price data could be fake since the input channels from remote sources are vulnerable. Assuming (trusted) historical data is available, we can further evaluate the trustworthiness of current inputs by comparing them with the reference data. Due to the different characteristics of smart pricing signals and remote user control commands, we evaluate different input channels with different models. In particular, real-time pricing signals mostly show a periodical pattern that repeats daily; while the user control commands are more likely to be scattered over a certain period of a day and usually conform to diversiform distributions. The RRA scheme estimates anomalies when the new patterns are not in accordance with a historic norm, and generates two risk indexes, indicating the belief (for RTP) and the confidence (for remote use control commands) that the input sources are trustworthy, respectively.

Algorithm 1 Multi-Criteria Reinforcement Learning for SUP

```
$i \leftarrow 1$
$c \leftarrow 0$
$\mathcal{C} \leftarrow \emptyset$
$\pi_{init} \leftarrow \emptyset$
repeat
  Obtain the current weight vector $\vec{w}$ and the starting state $x_k$
  if $\mathcal{C} \neq \emptyset$ then
    Compute $\pi_{init} \leftarrow \vec{w} \cdot \vec{Q}^{\pi}(x_k,a')$
    Initialize the $Q$-function vectors of the states
  end if
  Learn the new policy $\pi'$ through vector $Q$-Learning
  if $(\mathcal{C} = \emptyset)$ or $\vec{w} \cdot Q^{\pi_{init}}(x_k,a') - (\vec{w} \cdot Q^{\pi'}(x_k,a'') > \gamma)$ then
    $\mathcal{C} \leftarrow \mathcal{C} \cup \pi'$
    $c \leftarrow 0$
    $i \leftarrow i+1$
  else
    $c \leftarrow c+1$
  end if
until $c \geq \frac{1}{\epsilon} \ln \frac{(i+1)^2}{\delta}$
return $\mathcal{C}$
```

5.2.1 RRA-RTP

The smart pricing signal p in S2A is represented in terms of stochastic variables that are time indexed. Suppose RTP circulates in periods of T. Rather than serializing real-time pricing data continuously over time, we model the current pricing by exploiting the periodical structure of historic pricing information, and extracting each RTP p_t^d at time t $(t = 1, ..., T)$ as a distinct stochastic process, which evolves over the index of changing cycles d. For instance, if electricity pricing data changes/evolves daily, the pricing sequence at midnight could be modeled as random variables that are indexed with dates, as these pricing variables are more closely correlated and easier to be inferred.

For real-time risk assessment of smart pricing inputs, we choose a Hidden Markov Model (HMM), where the hidden states correspond to the working conditions $z_{1:T}$ of an appliance (i.e., time-indexed states indicating whether the appliance is under attack), and the observable states correspond to the real-time pricing states $p_{1:T}$ (and any other states that we could measure). Such Dynamic Bayesian Network (DBN) encodes the joint probability distribution over those stochastic variables that capture the evolution of the dynamic working conditions. In particular, we adopt the following state transition model P_t and observation model P_o:

$$z_t \sim P_t(\hat{z}_t|z_{t-1})$$
$$p_t \sim P_o(\hat{p}_t|\mathbf{p}_t^{hist}, z_t)$$

where $\hat{z}_t$ and $\hat{p}_t$ are the predicted states, $\mathbf{p}_t^{hist}$ denotes the historic pricing vector at time t, $p_t \in \mathbb{R}^+$ is the real-time pricing signal, and $z_t \in \{True, False\}$ denotes the unknown hidden states. The parameters of the conditional probability functions are known matrices that could be obtained or learned from the S2A system. Although we only consider smart pricing signals as observable states for discussion simplicity, it is worth mentioning that our RRA-RTP algorithm is also applicable for DBNs with multi-dimensional observation states with minor modification.

The aim of the analysis is to compute the posterior distribution of the hidden states $P(z_{0:t}|p_{1:t})$. Since the observation model could be non-Gaussian distribution (i.e., daily electricity pricing may change significantly with seasons), we employ a *particle filtering* (PF) algorithm [37] to approximate the probability distribution of the hidden variables. The basic idea is to establish a posterior probability distribution of the hidden variable by utilizing a large number of random samples. The samples are propagated over time in a sequential importance sampling step and a subsequent resampling step: (1) The SIS step generates samples from a specific probability distribution and computes their associate weight. (2) The resampling step then multiplies and/or discards these samples to automatically concentrate them in regions of interest of the state-space of the hidden variables.

Given N particles $\{z_{0:t-1}^{(i)}\}_{i=1}^N$ at time $t-1$ approximately distributed according to the distribution $P(z_{0:t-1}^{(i)}|p_{1:t-1})$, particle filters enable us to compute N particles $\{z_{0:t}^{(i)}\}_{i=1}^N$ approximately distributed according to the posterior $P(z_{0:t}^{(i)}|p_{1:t})$ at time t. As we cannot sample from the posterior directly, the PF update process is achieved by an appropriate importance proposal distribution $Q(z_{0:t})$, from which we can generate samples:

$$Q(\hat{z}_{0:t}|p_{1:t}) = Q(\hat{z}_t|z_{0:t-1}, p_t)P(z_{0:t-1}|p_{1:t-1})$$

The samples from $Q(\cdot)$ must be weighted by the importance weights

$$w_t = \frac{P(\hat{z}_{0:t}|p_{1:t})}{Q(\hat{z}_{0:t}|p_{1:t})} \propto \frac{P_o(p_t|\mathbf{p}_t^{hist}, \hat{z}_t)P_t(\hat{z}_t|z_{0:t-1})}{Q_t(\hat{z}_t|z_{0:t-1}, p_{1:t})} \qquad (5)$$

where $Q_t(\cdot|\cdot)$ denotes the choice of proposal distribution. To simplify the calculation, one can adopt the transition prior as proposal distribution (i.e., $Q_t(\cdot|\cdot) = P_t(\cdot|\cdot)$) [13]. In this case, the weights are given by the likelihood function

$$w_t = P_o(p_t|\mathbf{p}_t^{hist}, \hat{z}_t)$$

The detailed algorithm is shown in 2.

5.2.2 RRA-UCC

The patterns of user control commands are highly user-dependent, and may be non-revolving. Such characteristics make it infeasible to construct a probabilistic graphical model as we did in RRA-RTP for anomaly inference. Instead, we formulate a *frequentist approach* to assess the reliability of remote control signals, using observed frequencies and statistical hypothesis testing. With historic data, any given (daily) UCC input can be considered as one of an infinite sequence of possible repetitions of the same experiment, each capable of producing statistically independent results.

Algorithm 2 RRA-RTP with Particle Filtering

```
for $t = 1$ to $T$ do
    For $i = 1, ..., N$, sample from the transition priors $\hat{z}_t^{(i)} \propto P_t(z_t|z_{t-1}^{(i)})$, and set
        $\hat{z}_{0:t}^{(i)} \leftarrow (\hat{z}_t^{(i)}, z_{0:t-1}^{(i)})$
    For $i = 1, ..., N$, evaluate and normalize the importance weights
        $w_t^{(i)} \propto P_o(p_t|\mathbf{p}_t^{hist}, z_t^{(i)})$
    Multiply/Discard particles with respect to high/low importance weights $w_t^{(i)}$ to obtain $N$ particles $\{z_{0:t}^{(i)})\}_{i=1}^N$.
end for
```

In RRA-UCC, we integrate two complementary risk assessment schemes to detect anomalies in task starting time (e.g. remotely start the bread maker at 1am) and anomalies in task frequency (e.g. switch smart car charging on and off 10 times in a minute), respectively. In our settings, smart appliances have pre-built default distribution patterns of starting times, whose parameters are learned from the usage in the household. Normally, the UCC distribution prototype of appliances is a sum of N Gaussians in the form

$$f(x) = \sum_i a_i \exp(-\frac{(x-\mu_i)^2}{2\sigma_i^2}).$$

For instance, a smart dishwasher is embedded with a UCC prior in the form of three Gaussian distributions. In a household where residents do not eat breakfast, the first Gaussian will show a weak (or none) peak. Given the number of Gaussians in the prior distribution, we can easily obtain the parameters of each component distribution by multi-Gaussian fitting techniques (e.g.,[49]). The mean values μ_i are clustering centers of the user control commands. The confidence level α_t of an incoming command $c(t)$ appearing at time t is then evaluated according to the i^{th} Gaussian with the nearest mean value: $\alpha_t = f_i\{(t-\mu_i)/\sigma_i^2\}$, where $i = \text{argmin}_i\,(t-\mu_i)$. Next, to detect operation frequency anomalies, we explore the periodical control command interval distributions in a household. The basic idea is that, the intervals between adjacent operations within a certain period should conform to the historic norm. Suppose that a significant repeating cycle of an appliance's behavior is T (generally T should be $n \in \mathbb{N}^+$ days) . At time t, appliance A receives a remote control command $c(t)$. We obtain all the intervals between UCCs in $[t-T, t]$, and compare its distribution with the distributions of intervals in time slices $[t-2T, t-T], ..., [t-mT, t-(m-1)T]$ from historic clean data using non-parametric statistical testing approaches (i.e., Kolmogorov-Smirnov Test [9]). Then we select the historic time slice(s) where UCC intervals are most similarly distributed with the current time window $[t-T, t]$, and retrieve the statistical test results α_f (i.e., the p-value of K-S test) as a measurement of trustworthiness of the current operation frequency.

The frequentist approach gives a confidence level with a frequency probability interpretation and/or a pre-experiment interpretation. Such probabilities are combined as the risk

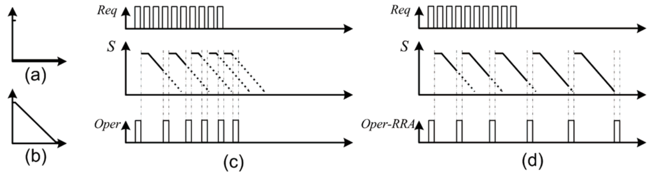

Figure 7: Real-time risk assessment for UCC of smart car charging system.

assessment of the user control command:

$$R_u = f(\alpha_t) + g(\alpha_f) \quad (6)$$

where $f(\cdot)$ and $g(\cdot)$ are monotone increasing functions.

Example 8: We consider a smart car, which could be remotely controlled to start and stop charging. In Figure 7, (a) and (b) are security penalty functions for the charging system: users can start charging at anytime, but need to wait for a while to restart charging after stopping it. An adversary, taking over the control, can send many consecutive charging requests. In the basic SUP model w/o risk detection, usability penalty increases as the later tasks are being hold by the security function. The increasing $u(t)$ penalty will force charging to restart soon after the security function drops below MAX. Restart interval will decrease with higher usability penalty. On the other hand, with RRA, it is detected that the requests are unusual. With more requests received, weight for $u(t)$ will decrease significantly, so that usability will have very small impact in operational decision making. Therefore, recharging interval will increase to a level that will not hurt physical security of the battery. □

6. SECURITY ANALYSIS

Objective. From security perspective, the goal of the S2A approach is to ensure that: (1) the smart device shall not work in extreme state; and (2) the smart device shall not work in abnormal state for a long period. It is acceptable that a device may need to work in abnormal mode for a short while in special circumstances, or while the risk assessment components are in the process of detecting an intrusion.

Threat model. We assume that smart devices are physically secure since they are usually located inside the house. We also assume that their control systems are not compromised – the control logic is relatively less complicate, and they only receive limited information (control and price) from designated sources; therefore, it is not easy to hack into the kernels of smart devices. Devices are under two types of threats: (a) improper operational requests from legitimate users; and (b) faked operational requests or electricity prices from attackers. Threat (a) is usually once-only, while threat (b) could be continuous and more risky.

Baseline security. In response to threat (a), physical security of each individual appliance is protected by security penalty function $s(t)$ in S2A. $s(t)$ defines the penalty of turning on or keeping on the appliance at time t. It cannot be overridden by other factors. However, it could be suppressed when the usability penalty is high (e.g. the task had been held for a long time), so that the appliance may work at non-favorable mode for a short period of time. Both $s(t)$ and $u(t)$ are generated by mechanisms embedded in smart appliances. Manufactures should set a very high security penalty (e.g. infinite) when the device is approaching extreme status. Moreover, to ensure security objective (1) described above, $s(t)$ cannot be surpassed when it reaches max – the device must be switched off. Therefore, with a properly designed security function, the device is guaranteed not to work in extreme state. The baseline security assurance applies for both threat (a) and (b).

Response to continuous attacks. Tier 2 of the S2A framework is to identify abnormal inputs, especially continuous abnormal inputs. Forged pricing data (or legitimate but unstable data) is detected by the RRA-RTP component. The RRA-RTP model employs HMM, so that the current risk assessment will affect the next assessment; therefore, the risk index will propagate continuously. When pricing information demonstrates unusual patterns for an extended period of time, RRA-RTP will detect increasing risk, and the weight for $p(t)$ will continuously decrease. In this way, price factor will become too weak to disturb the normal operations of the device. On the other hand, fake user input will be detected by RRA-UCC. A one-time fake command may not be detected if the command history does not demonstrate a strong pattern, or the fake command falls in the pattern. Meanwhile, when the attacker sends multiple commands in a short period of time (e.g. "start battery charging" - "stop charging" - "start charging" - etc.), the high frequency abnormalities are always accurately detected. The weight for usability factor decreases accordingly, and the system sees less need to fulfill such requests. S2A ensures that the smart device will not work in extreme mode in any condition; and also ensures that the smart device will not work in abnormal mode for a long period, with the presence of continuous attacks (faked operational requests or electricity prices).

False positives. Traditional intrusion detection systems (IDS) strive to reduce false positives and false negatives. Conceptually, *false negatives* are undetected anomalies. As we have shown, since we do not label the input data with a binary decision (safe or abnormal), unusual inputs will always be penalized in the second tier of the SUP model. On the other hand, *false positives* are normal inputs (that appears to be suspicious) that are mistakenly labeled as anomalies. Again, since we do not enforce a decision boundary, such inputs are not classified as anomalies. As they carry patterns that are different from regular ones, they will be somehow penalized (i.e. weights will be reduced) in the SUP model. However, the degree of the penalties are lower than the "true negatives". More importantly, the existence of the usability criterion effectively balances the (wrong) penalties, so that users will not become extremely dissatisfied.

Comparison with rule-based security protection. In some appliances, security protection is provided by rule-based decision (e.g. the motor should stop after continuous operation for 5 minutes, or the device has to remain off for 3 minutes before turned on again). Compared with rule-based decision method, we provide fine-grained security protection – SUP starts protection before reaching the extremely critical point (i.e., the rule-based decision boundary), but also allows a certain level of compromise at the strong demands from other factors. Moreover, in the presence of continuous active attacks, we provide better security by dropping attacker inputs, instead of working at minimum-protection

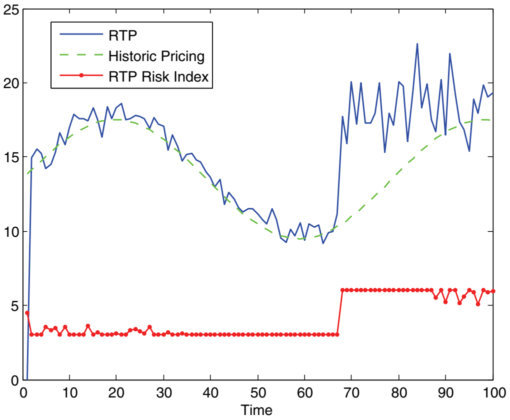

Figure 8: Real-time risk assessment for realtime pricing: RRA-RTP

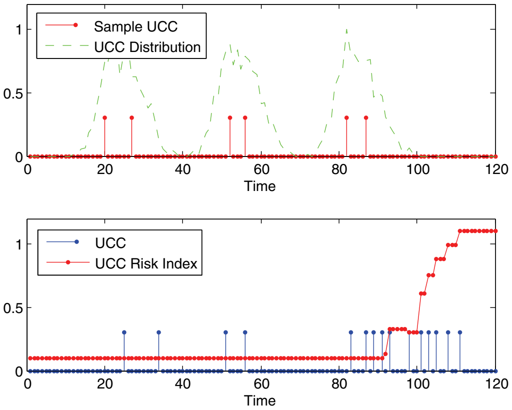

Figure 9: Real-time risk assessment for user control commands: RRA-UCC

conditions (as shown in Example 4 in Section 3). On the other hand, when we take smart pricing and remote control into consideration, the system becomes too complicated to be handled by rule-based models.

7. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the S2A approach, we first generated synthetic usage and pricing data based on heuristic assumptions, and tested S2A with these data. Note that our S2A framework could take arbitrary form of UCC and RTP inputs; as well as arbitrary form of security and usability penalties. In our simulation, the Q-value of a state-action pair converges after approximately 150 learning steps. In real world applications, however, the Q-table of a smart appliance is usually pre-trained by manufacturer, so that it would adapt to new conditions faster and more accurate.

For RRA-RTP, we implement Algorithm 2 with 1000 particles. The observation model P_o is set to be the weighted sum of historic mean RTP and white noises, where the weights are derived from the current states z_t in the HMM. As is shown in Figure 8, historical pricing (dashed line) follows a periodic pattern that revolves daily. The solid blue line denotes realtime pricing, and the red dots indicates the risk indexes (R_p) generated by RRA-RTP. As shown, RTP deviates away from the historic distribution starting from time point 68. The anomalies are detected and R_p increases correspondingly. On the other hand, Figure 9 shows the real-time risk assessments of user control commands. The upper plot shows the UCC distribution pattern, which is learned from historical control commands. The lower figure denotes real-time user control commands and the corresponding risk indexes generated by our algorithm. As we can see, slight offsets of request time will not immediately affect the risk assessment. However, clear unusual patterns (starting at time 90) are effectively detected. The risk index increases when we have higher confidence that the received control commands demonstrate an abnormal pattern.

We have tested S2A for different alternations of user commands, electricity pricing and security penalty patterns. Figure 10 demonstrates part of the experiment, which contains a complete use case. In this experiment, we adopt a scenario that the device cannot work for a long time (e.g. a motor). As shown in the first plot, a request is made at time point 4829 (middle of X-axis). It is put on hold due to high RTP (plot 4), and usability penalty starts to increase (plot 3). At approximately time 4840, usability penalty surpasses RTP penalty, the job starts to be processed, and the security penalty increases. S and P together stop the operation at time 4842, and waited until time 4848, when RTP drops to very low. From 4848 to the end of the task, the security penalty has stopped the operation twice, to force the motor to cool off. Overall, the task was completed with balanced considerations of S, U, and P.

8. CONCLUSION & FUTURE WORK

In this paper, we present S2A, a two-stage security protection framework for smart household appliances. We first introduce a Security–User–Price (SUP) model to capture three key factors, and present a multi-criteria reinforcement learning (MCRL) approach to integrate all three factors to dynamically determine an optimal operational strategy for the smart device. Furthermore, we present two risk assessment approaches based on statistical inferences. They evaluate the trustworthiness of users' remote control commands as well as the pricing information received through smart grid communication systems. The realtime risk indices are seamlessly incorporated into the SUP model to serve as weighting factors of the corresponding penalty functions, therefore ensures device security under active attacks. Through security analysis and experimental results, we show that S2A protects the device security of smart appliances, while maintaining usability and economic efficiency.

We have presented the S2A model in the paper, however, deploying the model on smart appliances still requires a lot of research and engineering efforts. First, it is nontrivial to define security functions for different types of smart devices. For appliances with sensors to monitor device status, it is also challenging to quantify (usually non-linear) sensor inputs and assess risks. On the other hand, it requires intensive human and behavior studies to observe usage habits of different devices and construct usability functions from the observed patterns. That is, the model still needs to be equipped with application-specific parameters to demonstrate best performance. Finally, it is important and effective to enable collaborations between smart devices for situational awareness and better risk assessment.

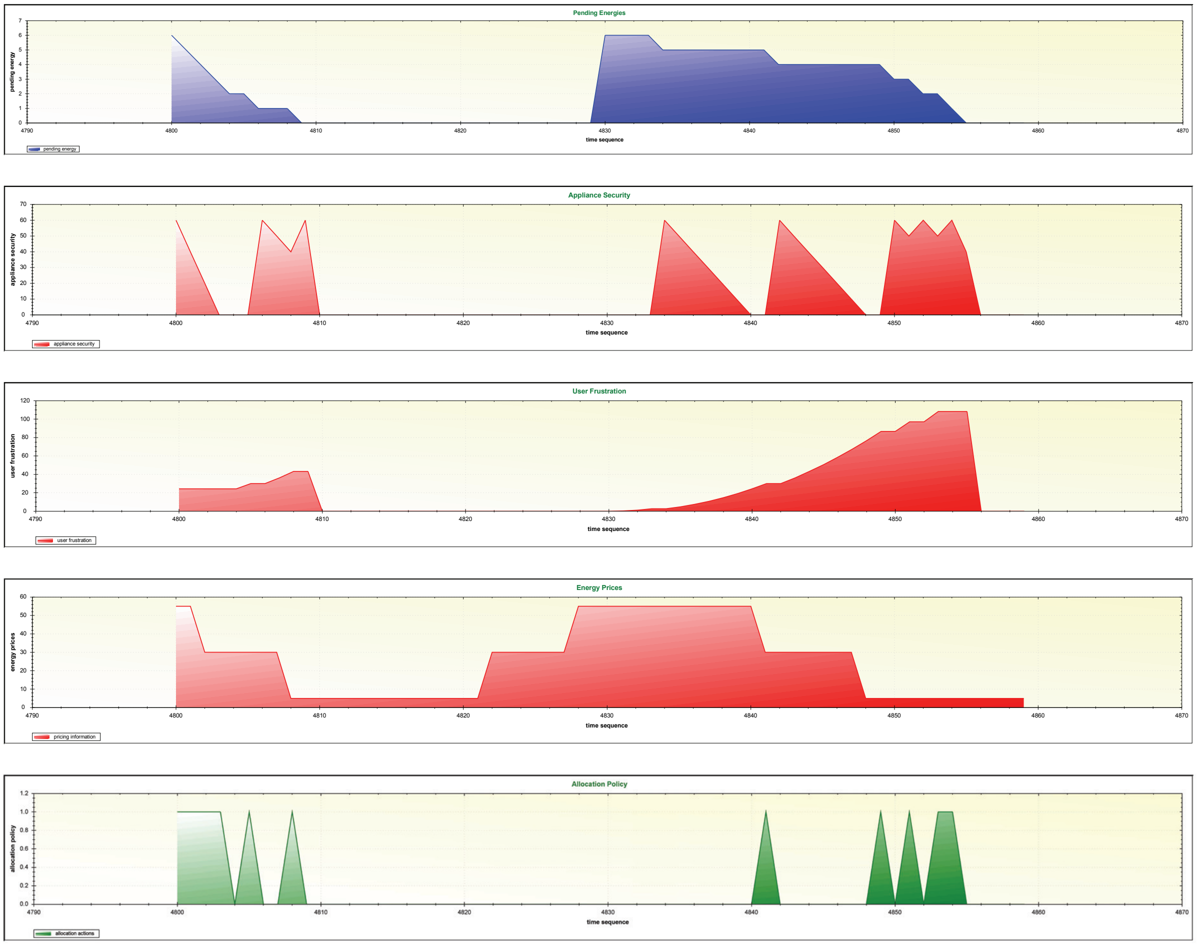

Figure 10: Sample results for S2A. From top to bottom: pending energy, appliance security penalty, usability penalty, smart pricing, and energy allocation actions.

9. ACKNOWLEDGEMENTS

Bo Luo was partially supported by NSF OIA-1028098 and University of Kansas General Research Fund 2301420. The authors would like to thank the anonymous reviewers for their constructive suggestions to improve the manuscript.

10. REFERENCES

[1] A. Aggarwal, S. Kunta, and P. Verma. A proposed communications infrastructure for the smart grid. In *Innovative Smart Grid Technologies (ISGT)*, 2010.

[2] T. Baumeister. Literature review on smart grid cyber security. Technical Report CSDL-10-10, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, Dec. 2010.

[3] R. Berthier, W. Sanders, and H. Khurana. Intrusion detection for advanced metering infrastructures: Requirements and architectural directions. In *IEEE SmartGridComm*, pages 350 –355, oct. 2010.

[4] S. Borenstein. The long-run efficiency of real-time electricity pricing. *The Energy Journal*, 26(3), 2005.

[5] S. Borenstein. The redistributional impact of non-linear electricity pricing. Working paper 602, Regulation2point0, 2010.

[6] A. Bose. Smart transmission grid applications and their supporting infrastructure. *Smart Grid, IEEE Transactions on*, 1(1):11 –19, june 2010.

[7] F. Cleveland. Cyber security issues for advanced metering infrasttructure (ami). In *IEEE Power and Energy Society General Meeting*, pages 1 –5, july 2008.

[8] F. Cohen. The smarter grid. *Security Privacy, IEEE*, 8(1):60 –63, jan.-feb. 2010.

[9] W. J. Conover. *Practical Nonparametric Statistics.* John Wiley & Sons, December 1998.

[10] Q. Dam, S. Mohagheghi, and J. Stoupis. Intelligent demand response scheme for customer side load management. In *IEEE ENERGY 2008*, 2008.

[11] H. Farhangi. The path of the smart grid. *Power and Energy Magazine, IEEE*, 8(1):18 –28, 2010.

[12] A. Faruqui and S. George. Quantifying customer response to dynamic pricing. *The Electricity Journal*, 18(4):53 – 63, 2005.

[13] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107 –113, apr 1993.

[14] A. B. Haney, T. Jamasb, and M. G. Pollitt. Smart metering and electricity demand: Technology,

economics and international experience. Technical report, Faculty of Economics, University of Cambridge, 2009.

[15] K. Herter, P. McAuliffe, and A. Rosenfeld. An exploratory analysis of california residential customer response to critical peak pricing of electricity. *Energy*, 32(1):25 – 34, 2007.

[16] A. Johnson. The history of the smart grid evolution at southern california edison. In *Innovative Smart Grid Technologies (ISGT)*, pages 1 –3, jan. 2010.

[17] W. Kempton and J. Tomic. Vehicle-to-grid power fundamentals: Calculating capacity and net revenue. *Journal of Power Sources*, 144(1):268 – 279, 2005.

[18] W. Kempton and J. Tomic. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of Power Sources*, 144(1):280 – 294, 2005.

[19] H. Khurana, M. Hadley, N. Lu, and D. Frincke. Smart-grid security issues. *Security Privacy, IEEE*, 8(1):81 –85, jan.-feb. 2010.

[20] Y. Kim, T. Schmid, Z. M. Charbiwala, and M. B. Srivastava. Viridiscope: design and implementation of a fine grained power monitoring system for homes. In *Ubicomp '09*, 2009.

[21] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker. Inferring personal information from demand-response systems. *IEEE Security and Privacy*, 8:11–20, 2010.

[22] S.-W. Luan, J.-H. Teng, S.-Y. Chan, and L.-C. Hwang. Development of a smart power meter for ami based on zigbee communication. In *PEDS*, 2009.

[23] J. marc Seigneur, C. D. Jensen, S. Farrell, E. Gray, and Y. Chen. Towards security auto-configuration for smart appliances. In *in Proceedings of the Smart Objects Conference*, pages 03–45, 2003.

[24] M. Masoum, P. Moses, and S. Deilami. Load management in smart grids considering harmonic distortion and transformer derating. In *Innovative Smart Grid Technologies (ISGT)*, 19-21 2010.

[25] S. Massoud Amin and B. Wollenberg. Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3(5):34 – 41, sept.-oct. 2005.

[26] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3):75 –77, may-june 2009.

[27] S. McLaughlin, D. Podkuiko, and P. McDaniel. Energy theft in the advanced metering infrastructure. In *Critical Information Infrastructures Security*. 2010.

[28] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In *ACSAC*, 2010.

[29] N. Mehta, S. Natarajan, P. Tadepalli, and A. Fern. Transfer in variable-reward hierarchical reinforcement learning. *Machine Learning*, 73(3):289–312, 2008.

[30] A. Metke and R. Ekl. Smart grid security technology. In *Innovative Smart Grid Technologies*, 2010.

[31] A. R. Metke and R. L. Ekl. Security technology for smart grid networks. *Smart Grid, IEEE Transactions on*, 1(1), june 2010.

[32] K. Moslehi and R. Kumar. Smart grid - a reliability perspective. In *Innovative Smart Grid Technologies (ISGT)*, pages 1 –8, 19-21 2010.

[33] H. Nakakita, K. Yamaguchi, M. Hashimoto, T. Saito, and M. Sakurai. A study on secure wireless networks consisting of home appliances. *Consumer Electronics, IEEE Transactions on*, 49(2):375 – 381, may 2003.

[34] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra. Residential demand response using reinforcement learning. In *IEEE SmartGridComm*, 2010.

[35] F. Orecchini and A. Santiangeli. Beyond smart grids - the need of intelligent energy networks for a higher global efficiency through energy vectors integration. *International Journal of Hydrogen Energy*, 2011.

[36] D. Petersen, J. Steele, and J. Wilkerson. Wattbot: a residential electricity monitoring and feedback system. In *CHI*, 2009.

[37] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[38] B. D. Russell and C. L. Benner. Intelligent systems for improved reliability and failure diagnosis in distribution systems. *Smart Grid, IEEE Transactions on*, 1(1):48 –56, june 2010.

[39] T. Sauter and M. Lobashov. End-to-end communication architecture for smart grids. *IEEE Transactions on Industrial Electronics*, 58(4), 2011.

[40] S.-Y. Son and B.-J. Chung. A korean smart grid architecture design for a field test based on power it. In *Transmission Distribution Conference Exposition: Asia and Pacific, 2009*, pages 1 –4, oct. 2009.

[41] V. Sood, D. Fischer, J. Eklund, and T. Brown. Developing a communication infrastructure for the smart grid. In *Electrical Power Energy Conference (EPEC), 2009 IEEE*, pages 1 –7, oct. 2009.

[42] G. Srinivasa Prasanna, A. Lakshmi, S. Sumanth, V. Simha, J. Bapat, and G. Koomullil. Data communication over the smart grid. In *IEEE International Symposium on Power Line Communications and Its Applications*, 2009.

[43] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[44] D. Wei, Y. Lu, M. Jafari, P. Skare, and K. Rohde. An integrated security system of protecting smart grid against cyber attacks. In *Innovative Smart Grid Technologies, 2010*, pages 1–7. IEEE, 2010.

[45] X. Wei, Z. Yu-hui, and Z. Jie-lin. Energy-efficient distribution in smart grid. In *SUPERGEN*, 2009.

[46] M. Weiss and D. Guinard. Increasing energy awareness through web-enabled power outlets. In *MUM*, 2010.

[47] M. Weiss, F. Mattern, T. Graml, T. Staake, and E. Fleisch. Handy feedback: connecting smart meters with mobile phones. In *Ubicomp '09*, 2009.

[48] P. Wolfs and S. Isalm. Potential barriers to smart grid technology in australia. In *Australasian Universities Power Engineering Conference*, 2009.

[49] D. Xu, L. Yang, and Z. He. Overcomplete time delay estimation using multi-gaussian fitting method. In *IEEE VLSI Design and Video Technology*, 2005.

[50] Z. Zhang. Smart grid in america and europe: Similar desires, different approaches (part 1). *Public Utilities Fortnightly*, 149(1), 2011.