A Multi-Sensor Energy Theft Detection Framework for Advanced Metering Infrastructures

Stephen McLaughlin, Brett Holbert, Ahmed Fawaz, Robin Berthier, and Saman Zonouz

***Abstract*—The advanced metering infrastructure (AMI) is a crucial component of the smart grid, replacing traditional analog devices with computerized smart meters. Smart meters have not only allowed for efficient management of many end-users, but also have made AMI an attractive target for remote exploits and local physical tampering with the end goal of stealing energy. While smart meters posses multiple sensors and data sources that can indicate energy theft, in practice, the individual methods exhibit many false positives. In this paper, we present AMIDS, an AMI intrusion detection system that uses information fusion to combine the sensors and consumption data from a smart meter to more accurately detect energy theft. AMIDS combines meter audit logs of physical and cyber events with consumption data to more accurately model and detect theft-related behavior. Our experimental results on normal and anomalous load profiles show that AMIDS can identify energy theft efforts with high accuracy. Furthermore, AMIDS correctly identified legitimate load profile changes that more elementary analyses classified as malicious.**

***Index Terms*—Power grid critical infrastructures, intrusion and energy theft detection, multi-sensor inference and information fusion, intrusion alert correlation, advanced metering infrastructures.**

I. Introduction

THE ADVANCED Metering Infrastructure (AMI) is changing the way electricity is measured, consumed, and even distributed. Digital smart meters remotely report not only fine-grained energy consumption data, but also logs of events indicating malfunctions, misconfigurations, and potential physical tampering. These monitoring capabilities, coupled with large-scale AMI data aggregation promise to significantly mitigate the problem of energy theft, an especially pervasive problem in developing countries.

However, the recent nation-wide AMI deployment effort has had quite an opposite effect by fueling concerns about new ways to steal power, e.g., through remote smart meter compromise. For instance, in 2009, the FBI reported a wide and organized energy theft attempt that may have cost up to 400 million dollars annually to a utility following an AMI deployment [1]. Indeed, AMI significantly increases the attack surface that utilities have to protect by introducing new cyber threats on physically-accessible devices [2]. Penetration testing efforts have shown vulnerabilities in smart meters that could lead to stealthy energy fraud. Additionally, remote meter reading eliminates the monthly visit by technicians to record consumptions and to visually inspect meters.

As a result, the need for an efficient monitoring solution to detect energy theft attempts in AMI has never been more critical. In this paper, we introduce AMIDS, an integrated cyber-physical intrusion detection system to identify malicious energy theft attempts. AMIDS differs from previous solutions by evaluating multiple AMI data sources under a combination of techniques to detect theft-related behavior while reducing false positives. In particular, AMIDS uses an attack graph based information fusion technique to conceptually combine collected evidences from three types of AMI-specific information sources: 1) cyber-side network- and host-based intrusion detection systems; 2) on-meter anti-tampering sensors; and 3) power measurement-based anomalous consumption detectors through nonintrusive load monitoring (NILM). The main contributions of this paper are as follows:

- We present an information fusion solution which makes use of an AMI-specific attack graph to identify energy theft attempts with minimum number of false positives.
- We leverage data mining techniques to identify energy theft through nonintrusive load monitoring. We designed two algorithms: a supervised approach that can identify individual appliance consumption and an unsupervised approach that learns by clustering load events.
- We build a realistic household load simulator that we used to evaluate the different individual detection techniques and the information fusion solution through the injection of realistic energy theft attacks.

II. Related Work

A variety of techniques have been discovered and performed to steal energy, starting from customer homes and up to the utility billing system. At the level of customer homes, the most common techniques are to tap energy from a neighbor or from a feeder or to tamper with meters so that consumption values are not properly recorded or not correctly reported. Tampering with meters includes applying magnets to slow down electromagnetic meters or to even pertubate measurements from solid state[1] meters, reversing or disconnecting meters, and hacking into the firmware of smart meters [4]. At the level of the grid, energy theft usually bypass meters by wiring heavy appliances (e.g., AC or heater unit) directly to the grid, or connecting their entire electric system to a feeder with a pirate transformer. Finally at the level of the utility, intentional or unintentional inaccuracy in the billing

Manuscript received October 8, 2012; revised March 15, 2013. This material is based upon work supported by the Department of Energy under Award Number DE-OE0000097.

S. McLaughlin and B. Holbert are with Computer Science and Engineering, Pennsylvania State University (e-mail: {smclaugh, bdh5027}@cse.psu.edu).

A. Fawaz and R. Berthier are with Information Trust Institute, University of Illinois (e-mail: {afawaz2, rgb}@illinois.edu).

X. Zonouz is with Electrical and Computer Engineering, University of Miami (e-mail: s.zonouz@miami.edu).

Digital Object Identifier 10.1109/JSAC.2013.130714.

[1]A meter in which the metrology is performed by solid-state components (as opposed to the mechanism used in more traditional electro-mechanical meters) [3].

system can cause important losses of energy revenue. Those inacurracies are either unintentional (e.g., incorrect meter multiplier value to compute overall energy consumption from sample recording) or intentional (e.g., customers switching their meter with a vacant premise or corrupted employees altering billing records). It is important to also note that the addition of smart communication infrastructure to the grid can increase attack vectors. For example, it can become trivial for a customer to jam the radio frequency communication so that automated meter alarms can never reach the utility database.

Energy theft has been a problem for utilities since the beginning of energy billing. Addressing this issue has been one of the motivation to invest in AMI [2], [5]. Indeed, smart meters have been designed to detect and report tampering attempts and the fact that they are solid-state eliminates some attack techniques that were popular with traditional analog meters. Alarms from smart meters have the potential to identify meters being tilted, disconnected, reversed or even hacked into. In addition to invidual meter alarms, utilities can detect energy theft with higher accuracy by leveraging the large scope and detailed resolution of AMI data to correlate events over time and across their entire customer base with additional information [6], [7]. For example, utilities can be alerted about typical symptoms of energy theft such as irregular outage notifications from a specific customer, or invalid consumption values from vacant premises. Moreover, detailed energy consumption profiles can be built over several months and change detection algorithms are applied to detect abnormal deviations (typically, a 20% threshold is used to trigger an alarm). Those profiles can be further normalized against customer profiles (e.g., residential or industrial) or geographical information system (GIS) so that outliers are easier to identify. Additionally, utilities can deploy meters on feeders or transformers to compare energy consumptions at the neighborhood level and at the level of individual meters. Mismatches between values reported that cannot be justified by technical losses are used to trigger alerts.

While the wide array of detection techniques brought by AMI seems to offer a comprehensive solution, the problem of energy theft remains a critical issue and utilities are now facing two new important challenges. First, smart meters are actually not tamper-proof and [4], [8] even demonstrate that the deployment of AMI introduces a significant set of new attack techniques to achieve energy theft. Those techniques include interrupting measurements, gaining privileged access to the meter firmware, tampering with the meter storage, and intercepting the meter communications to block or alter consumption values being reported. Second, alarms from the various energy theft detection techniques offered by AMI are highly proned to false positives (it is believed that up to 95% of tamper flags are erronoeous [9]) and utilities now have the difficult new task of dealing with a deluge of data from which identifying energy theft has become a sophisticated data mining challenge.

Solutions to energy theft have also been introduced from academia over the past few years [10]. A popular approach has been to apply support-vector machine (SVM) to energy consumption profiles [11], [12]. This approach consists in training a SVM from a historical dataset and then testing the SVM on a different dataset to find irregularities or deviations in the customer energy consumption profile. [11] reports an accuracy of 98.4% based on a training set of 440 instances and a testing set of 220 customers. The same authors extended their approach in [13] to leverage a hybrid neural-network model and encoding technique in order to automatically set the numerous parameters required by the SVM model. [14] studies a different method by focusing on identifying problematic metering installations (e.g., due to misconfiguration, energy theft or failure) through a central observer meter deployed at each neighborhood. This approach consists in comparing overall energy use with individual customer meters using a model of N linearly independent equations. This model is solved using matrix inversion and recursive statistical methods (i.e., least squares). The main limitations of this approach are to rely on a set of assumptions that often do not hold, such as the linear independence of equations or the zero resistance of energy cables. [15] takes a radically different approach by using a harmonic generator to actively deteriorate appliances of customers who steal energy. The concept is to monitor consumption values from smart meters, identify suspicious non-technical losses, disconnect genuine customers, operate the harmonic generator for few seconds and then reconnect everyone. An important limitation of this solution is to require smart meters to be instrumented with harmonic sensors so that genuine appliances remains protected from the active probes. Moreover, if such sensor fails, damage to genuine customers could make the cost of false positives prohibitively high.

AMIDS. Besides theft detection techniques, AMIDS uses Non-intrusive Load Monitoring (NILM) to identify individual appliance behaviors from an aggregated net load profile. NILMs were first presented for residential use by Hart et al. [16]. This technique is essentially the same used by AMIDS. It uses transitions in the staedy state load as indications that an appliance has either turned on or off. This model makes the reasonable assumptiont that most residential appliances will have steady power conusmption during their operation, and that transient power signatures occuring when appliances first turn on are limited to brief instances. We make use of this technique by adopting another method used by Bergman et al. [17]. Here, a knapsack problem is solved to find the set of loads that most likely describes the current steady state load. We adapt this techniqe to instead, identify which loads may have contributed to a give edge transition. Additionally, we limit our search to at most three loads on a given transition. Additional NILM techniques that are useful for identifying space heaters [18] and energy saving appliances [19] may prove useful for future work.

III. Threat Model

There are a variety of known techniques for energy theft that we assume an adversary may attempt against an AMIDS-equipped AMI deployment. At the level of customer homes, the most common techniques involve either tapping an external source such as a neighbor or distribution feeder, or meter tampering to inhibit proper recording of consumption. The latter may be done by applying magnets to interfere with instruments such as electromechanical rotors or solid state current transformers, or by reversing or disconnecting meters

TABLE I
ATTACKS CLASSIFIED BY THREE DETECTION TECHNIQUES

Id	*Attack technique*
	Cyber
A_{c1}	Compromise meters through remote network exploit
A_{c2}	Modify the firmware/storage on meters
A_{c3}	Steal credentials to login to meters
A_{c4}	Exhaust CPU/memory
A_{c5}	Intercept/alter communications
A_{c6}	Flood the NAN bandwidth
	Physical
A_{p1}	Break into the meter
A_{p2}	Reverse the meter
A_{p3}	Disconnect the meter
A_{p4}	Physically extract the password
A_{p5}	Abuse optical port to gain access to meters
A_{p6}	Bypass meters to remove loads from measurement
	Data
A_{d1}	Stop reporting entire consumption
A_{d2}	Remove large applicances from measurement
A_{d3}	Cut the report by a given percentage
A_{d4}	Alter appliance load profile to hide large loads
A_{d5}	Report zero consumption
A_{d6}	Report negative consumption (act as a generator)

from their sockets. At the grid level, energy thieves usually bypass meters by wiring power hungry appliances directly to the grid, or connecting their entire electric system to a feeder with a pirate transformer.

Moreover, the addition of network communication and smart devices to the grid has also brought new attack vectors. For example, it is trivial for a customer to jam meter wireless communications to suppress physical tampering alarms. Cyber attack techniques against AMI have been recently studied [4], [8] and include interrupting measurements, gaining privileged access to the meter firmware, tampering with the meter storage, and intercepting the meter communications to block or alter consumption values being reported.

To summarize, we classify energy theft techniques into three categories: 1) physical attacks, 2) cyber attacks, and 3) data attacks having an impact on power measurements. Note that attacks in the third categories are made possible through attacks from the first and second categories. The different attacks are detailed in Table I. The table is used in the following sections as a guide to ensure a comprehensive coverage of the threats from the described detection solutions. Moreover, we draw from these attack techniques to simulate attack scenarios in order to evaluate AMIDS.

IV. INDIVIDUAL ENERGY THEFT DETECTION MECHANISMS

A. Physical Tampering Detection Solutions

Smart meters are already equipped with sensors to collect and log potential physical tampering events such as removal of the meter cover and physical bumping of the meter. However, a problem with some such alerts is the high rate of false positives. For example, a heavy truck passing near a meter can trigger the tilt alert [9]. Thus, we include such tamper detection sensors in our solution to detect physical attacks, but false positives are reduced by combining tampering alerts with additional data sources covered in the following two sections. In other words, the overall false positive rate reduction in AMIDS is mainly because the alerts are correlated using an attack graph model (Section V). As a clarifying example, consider the case where a sensor alert is triggered indicating a malicious event that, according to the attack graph, requires several prior exploitations, and none of those exploitations has been reported by any sensor in the past. Using the attack graph-based attack analysis, AMIDS will most likely mark the triggered alert as a false positive and will not report any intrusion. We have further clarified this through our evaluations in Section VI. From our threat models described in Table I, meter tamper alerts provide the following observations. (O_1 through O_7 are in the following section.)

- *Observation* O_8: anti-tampering alert to detect A_{p1}
- *Observation* O_9: reverse rotation alert to detect A_{p2}
- *Observation* O_{10}: disconnect alert to detect A_{p3}
- *Observation* O_{11}: anti-tampering alert to detect A_{p4}

B. Cyber Intrusion Detection Systems

To address the challenge of detecting cyber attacks introduced by AMI, AMIDS leverages two complementary intrusion detection systems that can be implemented and deployed via firmware upgrade: 1) a remote cumulative attestation kernel (CAK) in meters [20], and 2) a specification-based network intrusion detection systems deployed on access points or dedicated sensors in the local neighborhood area network [21], [22]. A CAK is a lightweight solution for embedded systems such as meters, which records an unbroken sequence of application firmware upgrades. This audit log can be remotely queried by a verifier to detect firmware tampering, e.g., due to remote exploitation. The specification-based network intrusion detection system complements the firmware attestation system by monitoring communications among meters and access points from a network perspective. This IDS is deployed at the head-end or on dedicated sensors and works by analyzing traffic from the network to the application layers to ensure that devices are running in a secure state and their operations respect a specified security policy. Unlike traditional signature-based IDSes that rely on a database of known attack signatures, the specification-based approach identifies malicious traffic by checking if communications respect a model of expected behavior. Such a model is defined according to the communication protocol (ANSI C12.22 and C12.19 in our case) and a set of constraints on the network operations authorized among meters and the collection engine. This white-listing approach leverages the deterministic nature of an AMI to provide highly accurate detection capabilities. Thus, if an adversary attempts to compromise a meter through the network, the malicious requests sent (e.g., a network scan or the delivery of a harmful payload) will deviate from the expected legitimate behavior and trigger an alert. A strength of this approach has also been to provide formal guarantees about the coverage of the detection. A mathematical prover was used to demonstrate that the intrusion detection checkers will detect any attack that violates the security policy. In practice, we developed a real prototype of the specification-based IDS and experimented in the TCIPG testbed with actual meters to make sure that legitimate traffic would not trigger false positive while attack attempts would be precisely identified.

This prototype consists of a C12.22 dissector and a stateful detection engine that applies pre-defined constraints (e.g., timing, frequency, ordered sequence of requests/replies, permissions on read/write operations) on packets being collected in the AMI.

In summary, the cyber intrusion detection system provides the following observation capabilities to cover attacks from the threat models described in Table I:

- *Observation* O_1: spec.-based network monitoring to detect A_{c1}
- *Observation* O_2: remote firmware attestation to detect A_{c2}
- *Observation* O_3: spec.-based monitoring and meter authentication logs to detect A_{c3}
- *Observation* O_4: spec.-based monitoring and meter responsiveness to detect A_{c4}
- *Observation* O_5: spec.-based monitoring to detect A_{c5}
- *Observation* O_6: spec.-based monitoring to detect A_{c6}
- *Observation* O_7: remote firmware attestation to detect A_{p5}

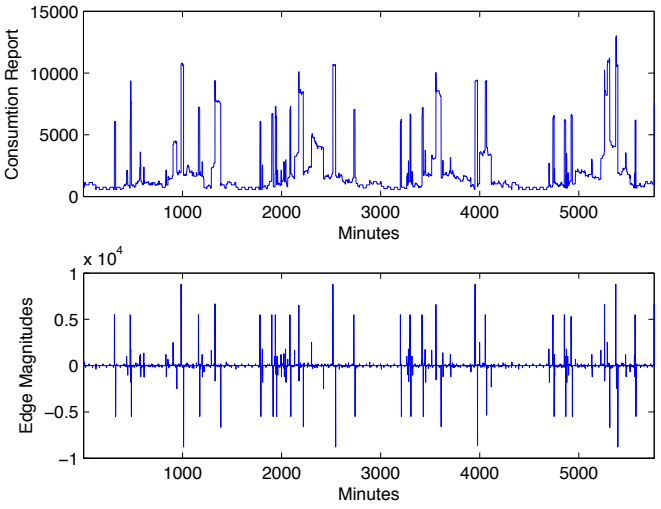


Fig. 1. A Sample 4-Day Load Profile and Corresponding Appliance ON/OFF Edges

C. Power Measurement-based Anomaly Detection

The third class of observations to detect theft-related behavior leverages the fine-grained load profile data available from smart meters. In particular, individual load profile events are analyzed to identify appliances being turned on and off. The results are used to create a usage *profile* for each household. These profiles will be used later to detect changes in household energy consumption patterns. In particular, we introduce two power measurement-based detection solutions based on supervised and unsupervised machine learning techniques. The algorithms are based on Naive Bayes learning that employs the method of maximum likelihood and is known to be one of the most effective and efficient classification algorithms in complex real-world situations.

We review the Naive Bayes algorithm briefly, and then discuss our two load-based energy theft detection solutions. Formally, the probability model for a classifier is a conditional model $Pr(C|F_1,F_2,\cdots,F_n)$ over a dependent class variable C that takes on a binary value, 0 (legitimate power consumer) and 1 (anomalous power measurements). F_i represents the i-th feature. Using a Bayes' theorem,

$$Pr(C|F_1,F_2,\cdots,F_n) = \frac{Pr(C)\cdot Pr(F_1,F_2,\cdots,F_n|C)}{P(F_1,F_2,\cdots,F_n)} \quad (1)$$

can be derived, and given the independence assumption,

$$Pr(C|F_1,F_2,\cdots,F_n) = \frac{1}{Z}P(C)\cdot\prod_{i=1}^{n}Pr(F_i|C), \quad (2)$$

where Z is a constant scaling factor representing the evidence. Given the above probability model, the Bayesian classifier combines this model with a decision rule. In particular, the hypothesis with the maximum a posteriori is picked:

$$C(f_1,f_2,\cdots,f_n) = \arg\max_{c\in\{0,1\}} P(C=c)\cdot\prod_{i=1}^{n}P(F_i=f_i|C=c). \quad (3)$$

Supervised Anomaly Detection. The supervised technique labels each on or off edge in the load profile according to its appliance of origin. The algorithm then determines which appliances $a \in A$ are missing from power measurements, i.e., if the mode of theft bypassed some appliances around the meter. The algorithm has two learning phases. First, a database of appliance signatures is created and stored for use by a Non-Intrusive Load Monitor (NILM). The NILM uses this database to identify appliance usage in the home over time. Second, AMIDS learns the daily usage frequencies of each individual appliance using appliance data provided by the NILM. More specifically, the power consumption time series are analyzed and the (*edges*) $E = (e_{t_0}, e_{t_1}, \cdots, e_{t_n})$ corresponding to on/off events are identified and recorded. Each edge magnitude represents one or more appliance events. Figure 1 shows 1) a sample power consumption time series of a single household generated by our implementation that simulated turn on/off incidents of 25 different home appliances, and 2) the identified edges within the same trace. The NILM works by solving the following binary integer programming problem to determine which devices contributed to a given edge.

$$\begin{array}{ll} \min & B^T x \\ \text{s.t.} & Qx \le e_{t_i} + \delta \\ & -Qx \le -e_{t_i} + \delta \\ & x \ge 0 \end{array} \quad (4)$$

where $B = [1,1,\cdots,1]_{2\cdot|A|\times 1}$; $Q = [Q_p; -Q_p]$, in which Q_p is an $|A|$-dimensional vector of power appliance consumption profiles, and $[a;b]$ represents the concatenation of the vectors a and b. This integer programming problem is solved to get the $2\cdot|A|$-dimensional binary vector x, where an element represents whether its corresponding appliance contributed to the edge e_{t_i}. Here, δ is a small threshold value to account for measurement noise. The objective of the optimization is to minimize number of incidents per edge. This is a reasonable assumption as many near-simultaneous appliance events are unlikely [16].

Once the set of appliances contribution to each edge is identified, AMIDS learns based on the daily frequency of each appliance f_a. Thus, over an n-day learning phase, a usage

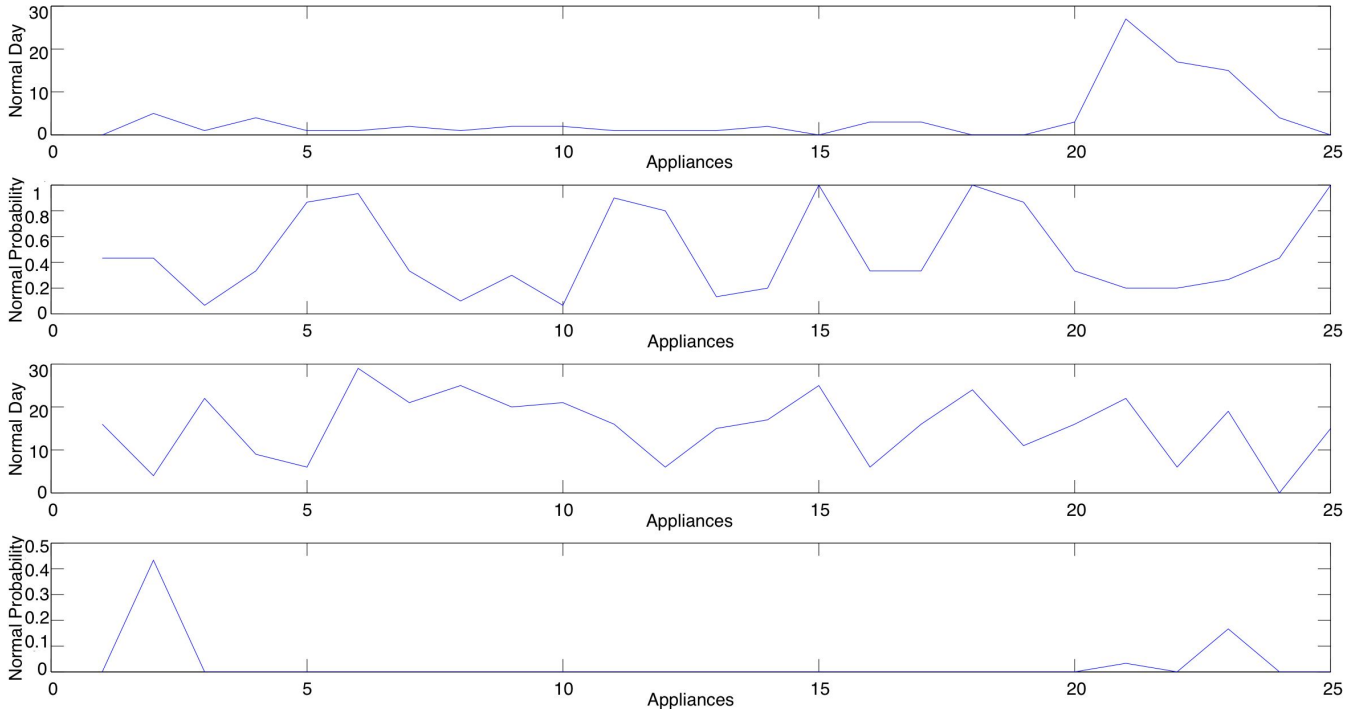


Fig. 2. Normal Day Profile and Classification Probability

profile matrix

$$U_{h_i} = \begin{pmatrix} f_{a_1,d_1} & f_{a_2,d_1} & \cdots & f_{a_{|A|},d_1} \\ f_{a_1,d_2} & f_{a_2,d_2} & \cdots & f_{a_{|A|},d_2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{a_1,d_n} & f_{a_2,d_n} & \cdots & f_{a_{|A|},d_n} \end{pmatrix} \quad (5)$$

per household h_i is saved. Each column is then used to calculate the probability mass distribution $P_{h_i,a_j}(v) \quad v \in \mathbb{Z}$ that appliance a_j is used v times per day in household h_i. This completes the profiling phase. Figure 2 shows our implementation results: 1) home appliance usage frequency reports of a single household over 20 days (each line represents a single day); and 2) the empirical probability mass distribution of the microwave usage frequency per day.

The calculated profiles (distributions) are used for anomaly detection purposes with the Bayesian classifier. The objective is to mark a given day-long smart meter measurements as normal or anomalous based on that household's profile. In particular, the prior class probability $P(C)$ in Equation (3) can be obtained from existing energy theft data [23], and the conditional distributions are obtained from the learned profiles. Here, a features F_i is the daily usage frequency of appliance i. Figure 2 shows our evaluation results for the supervised detection of anomalous power measurements. In particular, the first and second graphs show a normal trace for a single household over a day and the posterior distribution for individual appliances. As shown in the third and fourth graphs, a corrupted measurement trace leads to a significant reduction in the posterior distribution values (indicating that the reported measurements are less likely to be normal).

We note that the use of NILMs along side smart meters has raised privacy concerns [24]. Recent studies have shown that NILMs can reveal home occupant behaviors [25], [26]. While we defer the design of privacy-preserving protocols [27] for our scheme to future work, we mention a practical measure to mitigate leaks of most legitimate user's consumption patterns. Fine grained data for usage by load-based detection schemes can be released only after physical or cyber tampering alarms have been raised.

Unsupervised Anomaly Detection. The unsupervised detector groups individual load events into clusters based on their real-power magnitude. Knowledge of the exact number or type of appliances is not assumed. This should not be confused with isolating each appliance to its own cluster. Instead, each cluster represents an equivalence class of appliances that all have similar real-power consumption. Thus, appliances with similar load sizes will be placed in the same cluster. As an example to give some intuition behind this approach, consider maliciously bypassing an HVAC unit around the meter. If only the net load is monitored, then this may not produce a significant enough change in load consumption to be noticeable. However, if the HVAC is in a cluster with only a few other large appliances, e.g., a furnace, then the net consumption and number of edge transitions in the load profile for this individual cluster will substantially differ if the HVAC is suddenly removed, i.e., it is indefinitely in the OFF state.

The unsupervised learning algorithm proceeds as follows. Edge detection is first used to extract a set of events $\{f_1, f_2, \ldots, f_n\}$ (positive or negative edges) from the load profile. As with the supervised algorithm, each edge represents an appliance on or off event. However, because no labels are given, it is not known which appliance the edge corresponds to. K-means clustering is then done based on individual event magnitudes, resulting in a set of clusters C with each individual cluster $c = \{f_1, f_2, ..., f_{|c|}\} \forall c \in C$. Note that $\bigcup_{c \in C} \; c = \{f_1, f_2, \ldots, f_n\}$. Of course, the number of clusters $|C|$ must be appropriate for the set of appliances in the given home. For example, a large number of clusters might

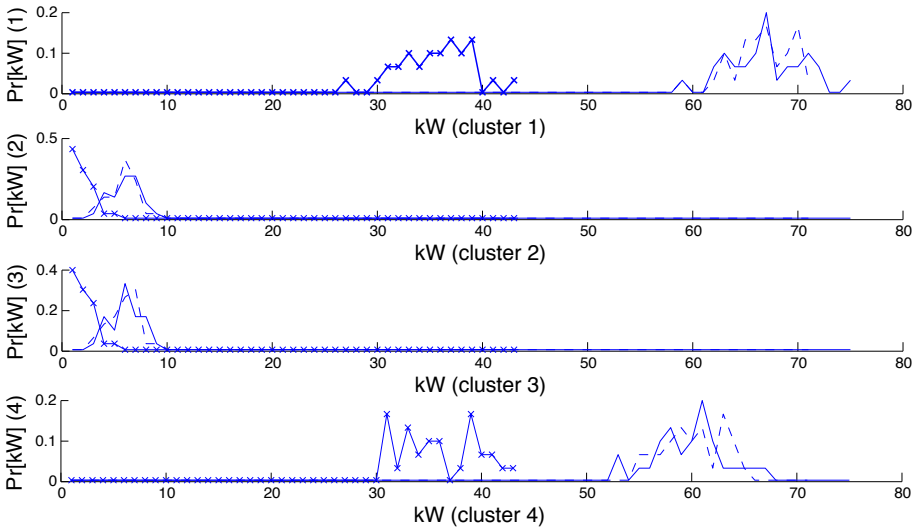


Fig. 3. Unsupervised Learning of Basline (solid), Legitimate (dashed), and Malicious $\times$ Profiles.

be inappropriate for an apartment with many small appliances with similar power consumptions. Thus, for a given set of input data, we repeatedly cluster the data using a different number of clusters each time. We then select the number of clusters $|C|$, that maximizes the average silhouette value s, defined as follows.

$$s = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|c|} \sum_{f \in c} \frac{b(f) - a(f)}{max\{b(f), a(f)\}} \tag{6}$$

Here, $b(f)$ is the Euclidean distance between f and all events in other clusters, and $a(f)$ is the distance between f and all events in its own cluster. Given an optimal clustering, the upper and lower bounds on each cluster are found and used to bucket events during normal operation. Bayesian classification is then done over the distribution of bucketed data against the clustering of the training data.

An example clustering of three datasets is shown in Figure 3 with four clusters formed from each dataset. The three datasets are as follows. (1) The solid line shows the probability density function (pdf) of events per day in each of four clusters from training data. (2) The dashed line is the pdf of events in a clustering of the same scenario with an HVAC system that is 30% more efficient than the baseline. (3) The line with $\times$ marks has the HVAC bypassing the meter. As can be seen, the clustering of the malicious test case differs significantly from the baseline and legitimate test cases.

In summary, the power-measurement monitoring system provides the following observation capabilities to cover attacks from our threat models described in Table I:

- *Observation* O_{12}: supervised and unsupervised anomaly systems to detect A_{p6}
- *Observation* O_{13}: utility-side report freq. checkers to detect A_{d1}
- *Observation* O_{14}: supervised anomaly system to detect A_{d2}
- *Observation* O_{15}: aggregated monthly changes to detect A_{d3}
- *Observation* O_{16}: supervised anomaly system to detect A_{d4}
- *Observation* O_{17}: unsupervised anomaly system to detect A_{d5}
- *Observation* O_{18}: utility-side negative consumption alert to detect A_{d6}

V. Multi-Source Information Fusion

Alerts from each of the security sensors discussed in Section IV indicate individual attack steps against AMI. However, as proved in practice, these sensors report fairly large numbers of false positives and sometimes miss intrusions; therefore, reporting energy theft solely based on individual alerts will result in many costly physical inspections. To improve the overall accuracy, AMIDS makes use of a novel model-based solution to correlate alerts and provides operators with contextual information. In particular, AMIDS leverages a set of common energy theft attack paths, i.e., the different ways that an energy theft attack could occur, to reduce false positives due to individual false alarms.

AMIDS uses an attack graph-based information fusion algorithm to combine evidence of on-going attacks from multiple sources. Figure 4 shows a simplified energy theft attack graph for a smart meter. The attack graph is a state-based directed graph which models various attack paths starting from the initial state s_0 and continues until the goal of theft (state s_g) is reached. At each node, the security state of the smart meter is identified by the following two binary values. 1) The attacker's current *privilege* in the meter: this captures what the attacker can do in the future, and is either none $\varnothing$ or the administrator privilege M. 2) The security *consequences* of attacker actions: this captures the set of actions the attacker has accomplished such as a modified meter firmware or exhausted CPU on the meter.

As shown in the figure, there are specific alerts and intrusion detection methods to identify each malicious action needed to proceed through the graph. Because these individual alerts are subject to false positives, AMIDS makes use of the attack graph to detect energy theft efforts by correlating alert sequences denoting a complete energy theft attack, i.e., a path from s_0 to s_g.

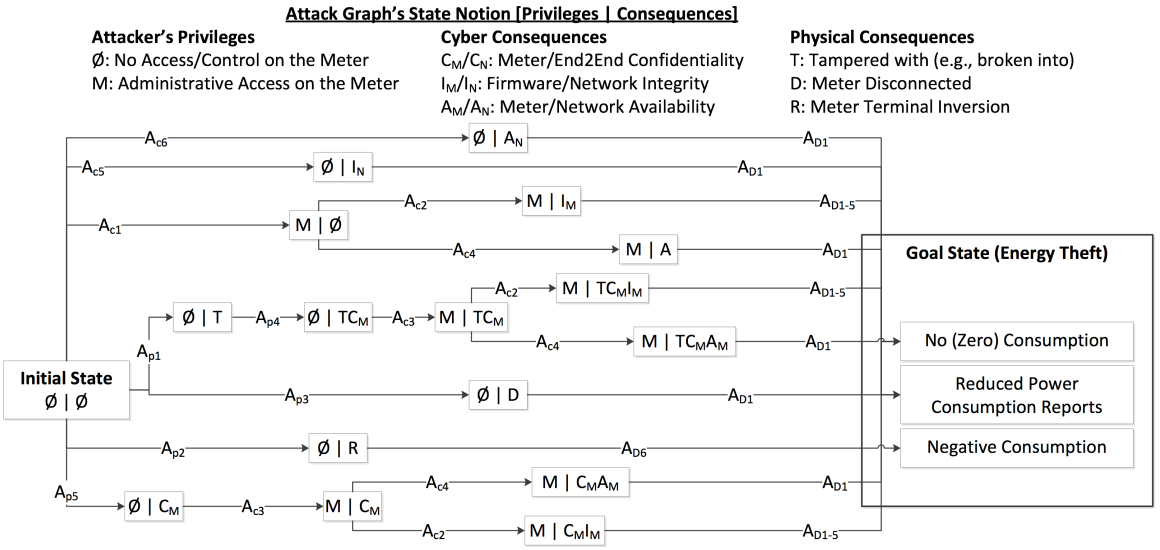


Fig. 4. A Simplified Cyber-Physical Attack Graph for AMI

To perform information fusion online, AMIDS considers the attack graph as a hidden Markov model (HMM) [28] and the alerts triggered by different detection techniques as observables $o_i \in \mathbf{O}$. Formally, AMIDS considers each attack path as a discrete-time hidden Markov process, i.e., event sequence $Y = (y_0, y_1, \cdots, y_{n-1})$ of arbitrary lengths. $y_i = (s_i, o_i)$, where s_i is an HMM state at the ith step of the attack and is unobserved, and the observation o_i is the set of triggered intrusion detection alerts at that step. AMIDS's main responsibility is to compute $Pr(s_t \mid o_{0:t})$, that is, the probability distribution over hidden states at each time instant, given the HMM model and the past IDS alerts $o_{0:t} = (o_0, \cdots, o_t)$. In particular, AMIDS makes use of the forward-backward smoothing algorithm [28], which, in the first pass, calculates the probability of ending up in any particular HMM state given the first k alerts in the sequence $Pr(s_k \mid o_{0:k})$. In the second pass, the algorithm computes a set of backward probabilities that provide the probability of receiving the remaining observations given any starting point k, i.e., $Pr(o_{k+1:t} \mid s_k)$. The two probability distributions can then be combined to obtain the distribution over states at any specific point in time given the entire observation sequence:

$$Pr(s_t \mid o_{0:t}) = Pr(s_k \mid o_{1:k}, o_{k+1:t}) \propto Pr(o_{k+1:t} \mid s_k) \cdot Pr(s_k \mid o_{1:k}), \tag{7}$$

where the last step follows from an application of Bayes's rule and the conditional independence of $o_{k+1:t}$ and $o_{1:k}$ given s_k. Having solved the HMM's smoothing problem for $Pr(s_t \mid o_{0:t})$, AMIDS probabilistically knows about the current state. Consequently, AMIDS picks the state with highest probability using the Most Likely State (MLS) technique [29] $s^* = \arg\max_s Pr(s_t \mid o_{0:t})$ and triggers the energy theft alert if $s^* = s_g$.

VI. Experimental Evaluations

A. Testbed

In order to realistically evaluate AMIDS, we leveraged our access to the smart meter testbed deployed by the Trustworthy Cyber Infrastructure for the Power Grid (TCIPG) center at the University of Illinois at Urbana-Champaign. Indeed, testing the framework requires a working AMI network with access to frequent meter readings and C12.22 traffic sent during network operation. The meter reading is used for the energy data mining. While the C12.22 traffic is used for the specification-based IDS. The smart meter testbed deployed in the TCIPG center consists of Itron equipment. In particular, we used Itrons OpenWay CENTRON architecture with 22 Centron smart meters, 4 cell relays, and a collection engine. The collection engine connects to an Oracle database installed on a dedicated host.

Meters form a wireless mesh network to connect to the cell relay. The mesh network is established using a proprietary communication scheme as part of the OpenWay system. The meters are connected to two separate neighborhood networks. Each network has a cell relay connected over a gigabit network to the collection engine. Each meter is connected to a load box enclosure to which we can easily plug loads and supplies.

We are currently collecting meter readings every 5 minutes, and we programmed the meters to send alerts for every possible tampering event. We are sniffing the alerts at the collection engine and storing them with a timestamp. Moreover, we connected the specification-based IDS using a network tap between the collection engine and the cell relays.

B. Load Profile Datasets

1) Baseline: We generate realistic load profiles based on simulated residents and their electric device usage. Each scenario is assigned a device profile consisting of a set of appliances, electronic devices, lighting, and other household items drawing power. Profiles are then created for individual occupant types, e.g., that cook, do other chores or are nocturnal. These occupant types can be combined to simulate the usage patterns of common household arrangements.

Each device consists of a usage profile with the device's power consumption as obtained from common device vendor websites. Each user profile then contains the times of day, number of uses, and durations of uses of each device. The

TABLE II
MULTI-SENSOR ENERGY-THEFT DETECTION USING THE AMIDS FRAMEWORK: PROBABILITIES OF DETECTION FOR SEQUENTIAL STEPS OF THREE DIFFERENT ATTACKS

		Attack Graph States ([Privilege|Consequence], as defined in Figure 4)																		
	Step $\mapsto$ Obs.	$\varnothing\|\varnothing$	$\varnothing\|T$	$\varnothing\|TC_M$	$M\|TC_M$	$M\|TC_MI_M$	$M\|C_MI_M$	$M\|TC_MA_M$	$\varnothing\|A_N$	$\varnothing\|I_N$	$M\|\varnothing$	$M\|I_M$	$M\|A$	$\varnothing\|D$	$\varnothing\|R$	$\varnothing\|C_M$	$M\|C_M$	$M\|C_MA_M$	Goal state
Attack 1	$A_{p5} \mapsto O_7$	0	0.65	0	0	0	0	0	0.06	0.06	0.06	0	0	0.06	0.06	0.05	0	0	0
	$A_{p3} \mapsto O_{10}$	0	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.03
	$A_{c3} \mapsto O_3$	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	$A_{c2} \mapsto O_2$	0	0	0	0	0.92	0	0.08	0	0	0	0	0	0	0	0	0	0	0
	$A_{d2} \mapsto O_{14}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Attack 2	$A_{p3} \mapsto O_{10}$	0	0.14	0	0	0	0	0	0.14	0.14	0.14	0	0	0.14	0.14	0.14	0	0	0
	$A_{c2} \mapsto O_2$	0	0	0.14	0	0	0	0	0	0	0	0.07	0.07	0	0	0	0.14	0	0.57
	$A_{d2} \mapsto O_{14}$	0	0	0	0.08	0	0.03	0	0	0	0	0	0	0	0	0	0	0.04	0.85
Attack 3	$A_{p4} \mapsto O_{11}$	0	0.08	0	0	0	0	0	0.08	0.08	0.08	0	0	0.08	0.08	0.5	0	0	0
	$A_{p5} \mapsto O_7$	0	0	0.08	0	0	0	0	0	0	0	0.04	0.04	0	0	0	0.5	0	0.33
	$A_{c3} \mapsto O_3$	0	0	0	0.13	0	0.38	0	0	0	0	0	0	0	0	0	0	0.38	0.13

time of day and duration fields each have a time granularity of one minute, giving us minute level load profiles. Previous work has shown that refrigerators loads follow roughly a 70 minute cycle and power is only drawn for half of that duration [30]. The simulated refrigerators are assigned a cycle between 60 and 70 minutes to introduce some variation into the model.

Power usage for the water heater is generated as a simplified version of the model used in [31]. Heat loss due to hot water use for showering, miscellaneous hot water usage, and ambient temperature difference is considered to decrease the water temperature at a constant rate. This results in only negligible variations in the power usage compared to the previous model. The HVAC system is simulated using a pre-calculated load curve for a given temperate pattern. The compressor is then simulated to approximately meet the load curve.

2) Legitimate Changes: Two modifications are made to the baseline load profiles: *legitimate*, and *malicious*. In the legitimate load, the traces are perturbed probabilistically according to the legitimate usage profile model to reflect legitimate deviations from the baseline. It is noteworthy that AMIDS creates/learn legitimate profiles for every household. Ideally, AMIDS will not raise any alerts for legitimate traces. Those traces are: (*Legit-Replace*) the replacement of a large appliance with a version 30% more efficient, (*Legit-Season*) reduced usage of heating or cooling appliances due to seasonal changes, and (*Legit-Occupant*) modified use of all appliances due to occupancy change.

3) Malicious Changes: In the malicious scenarios, the traces are perturbed to reflect load changes caused by common energy theft scenarios. Ideally, AMIDS will raise an alert for each malicious trace. The three malicious cases are: (*Mal-Bypass*) the bypassing of a large appliance, e.g., HVAC, around the meter, (*Mal-Disconnect*) periodic disconnection of the meter resulting in zero usage, and (*Mal-Reduction*) a constant reduction in measured power, e.g., due to magnets or meter hacking.

We will evaluate accuracy of the individual proposed detection solutions and the integrated AMIDS approach on various normal (baseline) and anomalous usage profiles.

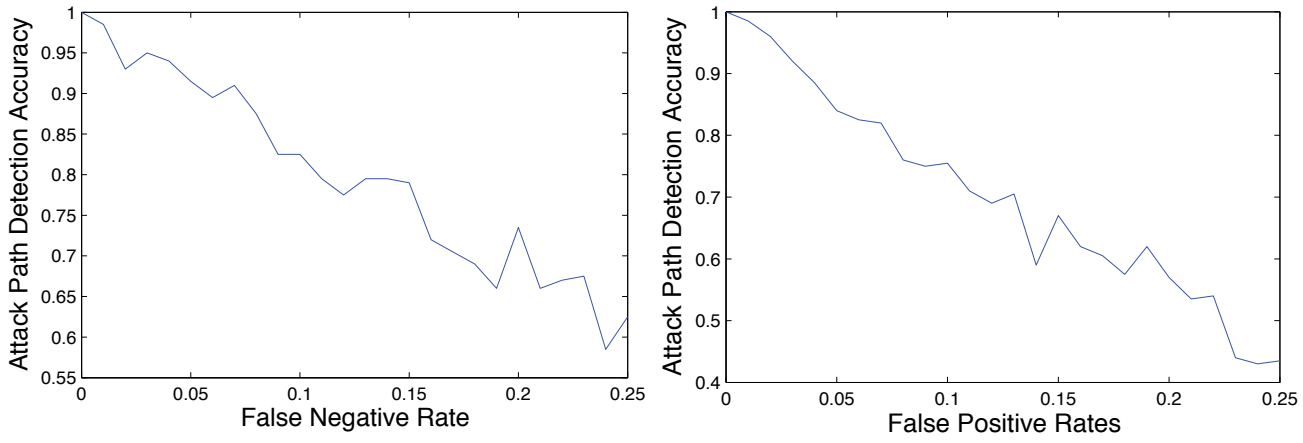
C. Integrated Intrusion Detection

We implemented the proposed HMM-based solution for the integrated energy theft detection, and evaluated its overall detection capability in dealing with sensor inaccuracies. In particular, AMIDS was tested against three complete and incomplete energy theft attack attempts (see Table II). The first attack was a 5-step energy theft attack which was reported by the intrusion detection sensors accurately (each step was reported by the corresponding sensor). Each row in Table II shows the posterior distribution over the attack graph's state space. As expected, AMIDS can detect the energy theft attempt accurately, i.e., $P(s_g|observations) = 1$. During the second attack scenario (identical steps), some alerts were not triggered by the sensors, and hence AMIDS had to infer the steps based on the attack graph structure. As shown in the table, after the last step, AMIDS reports the energy theft attempt with 85% confidence. The last incomplete attack scenario which actually does not result in the goal state is not reported as a successful energy theft attempt with 87% confidence.

D. Accuracy

We now evaluate the accuracy of AMIDS under a number of attacks on a load profile for a single occupant apartment. We are particularly interested in the accuracy gains that can be made through information fusion of (*i*) cyber IDS alerts, (*ii*) physical tampering alerts, and (*iii*) load-based IDS alerts, as compared to the accuracy of the individual methods. Table III shows the results of running the individual IDSs as well as the combined HMM approach on a single-occupant dwelling. A check mark means that the correct action was taken, and an $\times$ indicates a false positive or false negative. A dash indicates that the experiment did not apply. As can be seen, the combined approach eliminates the false positives of the individual approaches. Alerting capabilities for the cyber and physical IDSes were validated experimentally on real meters in the TCIPG testbed [32]. In particular, we disconnected and reversed meters and checked that alerts were generated. We also collected a week of meter traffic in a mesh network of nine meters and made connection attempts towards meters using a rogue software client in order to test our implementation of the ANSI C12.22 specification-based IDS.

Of particular instances are the three **Legit** cases designed to cause false positives in the load-based approaches. Indeed, the unsupervised learning algorithm identified two as malicious behavior. The lack of any cyber or physical IDS alerts in these cases resolved these false positives in the combined approach.



(a) False Negative Rates

(b) False Positive Rates

Fig. 5. HMM-based Attack-Path Detection Accuracy for Various Sensor False Positive/Negative Rates

TABLE III
EMPIRICAL DETECTION RESULTS FOR FOUR INDIVIDUAL SENSORS AND THE COMBINED AMIDS FOR A VARIETY OF ATTACK TECHNIQUES

	Cyber			Physical			Data Modification			
Detection	**Network Exploit (A_{c1})**	**Intercept Comm. (A_{c5})**	**Mal-Login (A_{c3})**	**Meter Breakin (A_{p1})**	**Meter Disconnect (A_{p3})**	**Appliance Bypass (A_{p6})**	**Legit-Replace**	**Legit-Season**	**Legit-Occupant**	**Mal-Reduction (A_{d3})**
Cyber IDSs	✓	✓	✓	-	-	-	-	-	-	-
Physical IDSs	-	-	-	✓	✓	✓	-	-	-	-
Supervised	-	-	-	-	✓	×	✓	✓	✓	✓
Unsupervised	-	-	-	-	✓	✓	✓	×	×	✓
AMIDS (HMM)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

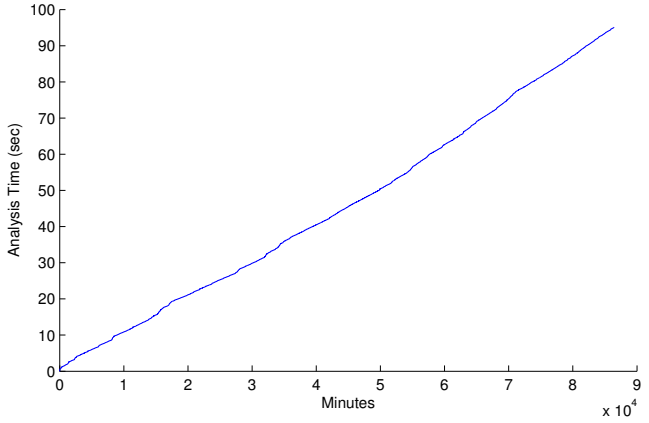


Fig. 6. Analysis Time Required (Y axis, in seconds) by the HMM-based Detection Approach to Process a Given Dataset Time Length (X axis, in 10^4 minutes)

An additional false negative by the supervised approach was also resolved. While additional field testing is necessary, these results show that the HMM approach used by AMIDS is an effective solution for combining smart meter data sources to identify energy theft behaviors.

We also evaluated the energy theft detection accuracy of the AMIDS's HMM-based information fusion component using a Monte Carlo emulation of attacks. In particular, our implementations generated numerous attack paths that either accomplish successful energy theft or fail in completing the intrusion. We measured the accuracy of the "energy theft" alerts triggered by the HMM inference engine. Figure 5(a) shows the intrusion detection accuracy results for various false negative rate values of individual physical, cyber, or power sensors. For instance, as shown in the figure, when sensors individually miss every malicious incident with 0.25 probability, the energy theft attack paths, reported by AMIDS, match the actual paths traversed by the attacker 62% of the cases. Figure 5(b) shows similar results for sensors with various false positive rates. The results on both of the figures are averaged over 20 runs.

E. Performance

We also conducted performance evaluation of how long two major AMIDS analysis phases take to complete. First, we measured the time requirements for learning phases for profiling different households' electricity consumption patterns given the collected dataset of the smart meter measurements. Figure 6 illustrates the results for smart meter measurement datasets of different time interval lengths. For instance, if the dataset stores the reported power measurements for a month (43200 minutes), AMIDS takes approximately 44 seconds to complete the dataset parsing, analysis, and household consumption profiling procedures. As expected, the analysis time grows linearly with the meter measurement dataset size. Although the learning phase is performed as an offline onetime effort, it is still important to complete the learning phase sufficiently fast for each household especially if a single power utility server is in charge of performing the analyses for many meters. Second, we evaluated the run time operation of the HMM-based energy theft detection component in AMIDS. In particular, we generated random attack graphs with different sizes (number of vertices) and a single attack path for each of them. Each attack path's length was equal to the graph's size. Then, we measured how long the HMM-based inference algorithm takes to start and complete the analyses, i.e., to report the best attack path estimate given the sensor alert sequence for the corresponding attack path. Figure 7 shows

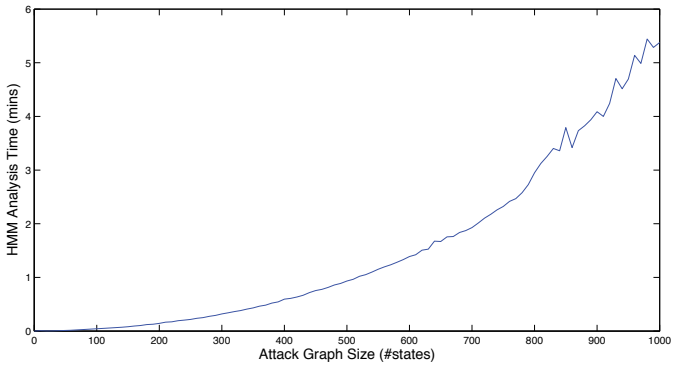


Fig. 7. Analysis Time Required (Y axis, in minutes) for the HMM-based Detection Approach to Process a Given Graph Size (X axis, in number of states)

the results in minutes. It is important to note that, in practice, the energy theft detection analysis and current HMM state estimation are performed step by step every time a sensor alert is triggered, whereas the results in the figure show the time needed for analysis of the whole alert sequence for an attack path.

F. Real-world Testbed Deployment

We evaluated the testbed data using the unsupervised learning algorithm to check for tamper detection. Initially the first 12 hours of data shown in Figure 8 were used to build the clustering, making the large anomalous outages in the subsequent 12 hours detectable. To test the flexibility of the clustering algorithm to learn not just specific devices, but classes of devices, we used an additional week of data for one fridge and one computer to construct the clusters, and then cross checked them against outages in different devices of the same class. Once again, the outages were detected as the lack of events during the outage period caused the distribution of events over clusters to change substantially. Figure 9 shows the sequence of events in same time series. The energy theft detection algorithms further refined the event time series (Figure 9) to ignore the negligible events which are caused mainly by sensor noise.

We deployed the HMM-based information fusion and energy theft detection framework in our AMI testbed. The results are shown in Figure 10. In particular, the attack was composed of five individual steps, namely $s_1 = (\emptyset|\emptyset)$, $s_{15} = (\emptyset|C_M)$, $s_{16} = (M|C_M)$, $s_{17} = (M|C_{M,M})$, $s_{18} =$ *Energy-Theft*, according to our attack graph (Figure 4). During the experiments, we also got a false positive indication a physical tampering with the device, i.e., the state transition from the initial state to the state $s_2 = (\emptyset|T)$. Figure 10 shows how the HMM-based state estimation results evolve after each individual alert gets triggered. For instance, the first graph on the top illustrates the AMIDS's belief about the system's current state before the attack started, i.e., the system is in the initial state currently with probability 1 and any other state with probability 0. The alert regarding the state s_2 was triggered which resulted in the second graph from the top. Clearly, the probability of being in state s_2 increased. Once we received the state s_{15}'s alert, AMIDS updates the state estimation result accordingly, giving less weight to state s_2. Once the alerts indicating states s_{16}, s_{17}, s_{18} are triggered, AMIDS uses the HMM's smoothing implementation to infer the current state of the system probabilisticaly and accurately. As a case in point, after the energy theft attack was complete, AMIDS calculated the probability of being in state s_{18} to be 0.87.

VII. Discussions

AMIDS employs alerts triggered by different types of AMI sensors as well as Markovian information fusion techniques to identify malicious energy theft efforts effectively. However, AMIDS's large-scale deployment requires a few other capabilities and solutions to be in place. In the following, we review these requirements and limitations briefly. The in-depth analysis of those requirements and their corresponding solutions are out of this paper's scope.

As one of the energy theft detection algorithms, AMIDS employs the power measurements to perform a non-intrusive load monitoring and obtain information about what home appliances are being used in a particular household. Traditionally, usage of the NILM techniques (the supervised learning-based detection in Section IV-C) in AMI infrastructures raise the concerns regarding customer privacy violations. The privacy violation concern in AMIDS can be addressed through two major techniques potentially. First, AMIDS can employ only the unsupervised learning-based techniques (Section IV-C) that do not distinguish individual home appliances by fingerprinting their electricity consumption signatures. Clearly, ignoring the extra information from the supervised solution will affect the energy theft detection accuracy of the AMIDS framework. Alternatively, as the more technical solution, AMIDS can make use of cryptographic privacy-preserving solutions using secure computation and homomorphic encryption techniques that are proposed in the recent AMI security literature, e.g., [33]. However, deployment of the cryptographic solutions using the existing algorithms would require strong computation capabilities.

Accurate detection of malicious energy theft efforts by misbehaving customers by AMIDS requires precise construction of the intrusion detection models and estimation of the involved parameter values. In AMIDS, the models are created either manually by the expert people or automatically by the machine learning solutions. Traditionally, the drawbacks and advantages of both of those solutions are known. Briefly, manual model creation will result in correct and accurate models (if a sufficient amount of time is spent); however, other than their scalability concerns, the manual techniques are not usually fast enough for change management and reconfigurations. On the other hand, automated machine learning-based model creation and parameter estimation methods, that is used in our current implementations, be design, require an attack-free environment during the learning phase when the values are learned. To address the abovementioned issue with the automated techniques, more frequent manual inspections by the power system engineers are needed to ensure that the learned profiles reflect the reality precisely.

Finally, before the AMIDS's large-scale deployment, the following questions must be answered: where should the

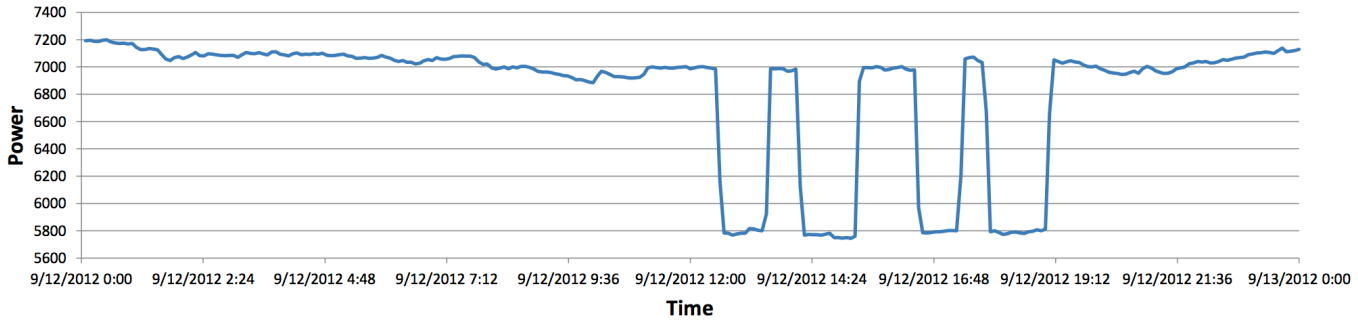


Fig. 8. Power Consumption Time Series Reported by a Smart Meter Under Attack

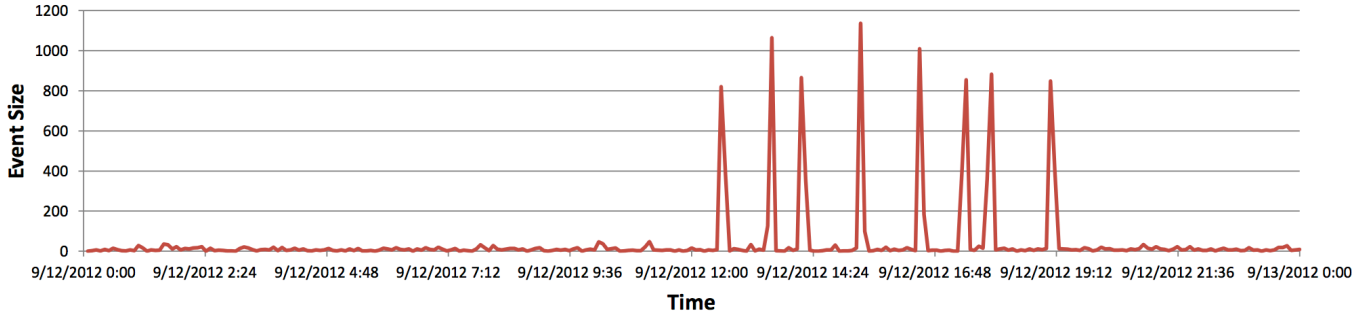


Fig. 9. Substantial Changes in the Distribution of On/Off Events for Appliances During a Successfully-detected Power Outage

AMIDS's intrusion detection analysis engine(s) be deployed? The main three promising possibilities are at the smart meter level, at the neighborhood feeder level, and at the power utility level. Decision upon the deployment points require an in-depth trust and scalability analysis. In particular, from the power utility's viewpoint, the last option, where the intrusion detection analyses are performed by the utility's local servers, is the most trustworthy option; however, it can not scale up for large-scale deployments where analyses for many meters must be computed in a centralized manner. Deployment of the AMIDS engines in individual smart meters eliminates the scalability concern; however, smart meters reside in the potentially untrusted zone, i.e., within the households. Furthermore, on-meter AMIDS deployment will require improvements in the meters' computational capabilities, and consequently their replacements, that may be costly in practice. Feeder-level deployment of the AMIDS engines provide a decent compromise for the abovementioned scalability-trust tradeoff, because it results in distributed intrusion detection analysis, and in the meanwhile, potentially malicious household owners can not tamper with the AMIDS analyses and energy theft detection reports.

VIII. Conclusions

In this paper, we presented AMIDS, an integrated intrusion detection solution to identify malicious energy theft attempts in advanced metering infrastructures. AMIDS makes use of different information sources to gather sufficient amount of evidence about an on-going attack before marking an activity as a malicious energy theft. Our experimental results show that through an effective information fusion and using the correlation among the triggered alerts, AMIDS can detect various types of energy theft attempts accurately using individually inaccurate sensors.

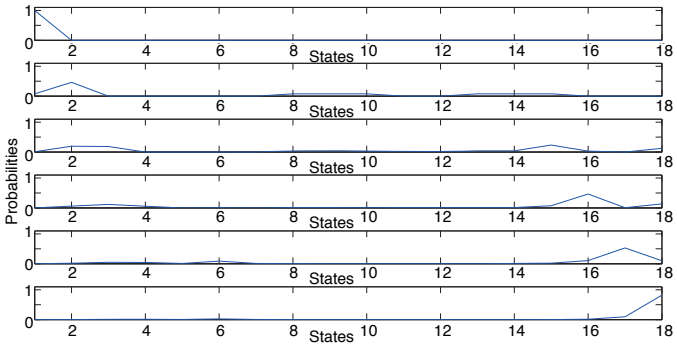


Fig. 10. Evolution of Probabilities of Being in a Given State by the HMM Approach During the Testbed Energy Theft Attack

References

[1] FBI: Smart Meter Hacks Likely to Spread. Available at http://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/.

[2] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy theft in the advanced metering infrastructure," in *Proc. international conference on Critical information infrastructures security*. Springer-Verlag, 2010, pp. 176–187.

[3] 2008, aMI Industry Glossary. Available at http://www.aclaratech.com.

[4] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel, "Multi-vendor penetration testing in the advanced metering infrastructure," in *Proc. Annual Computer Security Applications Conference*. ACM, 2010, pp. 107–116.

[5] S. Vukmirovic, A. Erdeljan, F. Kulic, and S. Lukovic, "Software architecture for smart metering systems with virtual power plant," in *MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical Conference*. IEEE, 2010, pp. 448–451.

[6] Z. Xiao, Y. Xiao, and D. H.-C. Du, "Non-repudiation in neighborhood area networks for smart grid," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 18–26, 2013.

[7] M. VEILLETTE, "Process for detecting energy theft," Patent Application US 2012/0 062 210 A1, 03 15, 2012. [Online]. Available: http://www.patentlens.net/patentlens/patent/US_2012_0062210_A1/en/

[8] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy theft in the

advanced metering infrastructure," *Critical Information Infrastructures Security*, pp. 176–187, 2010.
[9] B. Loeff, "Deputizing data: Using ami for revenue protection," *Utility Automation and Engineering*, 2008.
[10] S. Depuru, L. Wang, V. Devabhaktuni, and N. Gudi, "Measures and setbacks for controlling electricity theft," in *IEEE North American Power Symposium*, 2010, pp. 1–8.
[11] S. Depuru, L. Wang, and V. Devabhaktuni, "Support vector machine based data classification for detection of electricity theft," in *IEEE/PES Power Systems Conference and Exposition*, 2011, pp. 1–8.
[12] J. Nagi, K. Yap, S. Tiong, S. Ahmed, and A. Mohammad, "Detection of abnormalities and electricity theft using genetic support vector machines," in *IEEE TENCON Region 10 Conference*, 2008, pp. 1–6.
[13] S. Depuru, L. Wang, V. Devabhaktuni, and P. Nelapati, "A hybrid neural network model and encoding technique for enhanced classification of energy consumption data," in *IEEE Power and Energy Society General Meeting*, 2011, pp. 1–8.
[14] C. Bandim, J. Alves Jr, A. Pinto Jr, F. Souza, M. Loureiro, C. Magalhaes, and F. Galvez-Durand, "Identification of energy theft and tampered meters using a central observer meter: a mathematical approach," in *IEEE/PES Transmission and Distribution Conference and Exposition*, vol. 1, 2003, pp. 163–168.
[15] S. Depuru, L. Wang, and V. Devabhaktuni, "A conceptual design using harmonics to reduce pilfering of electricity," in *IEEE Power and Energy Society General Meeting*, 2010, pp. 1–7.
[16] G. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
[17] D. Bergman, D. Jin, J. Juen, N. Tanaka, C. Gunter, and A. Wright, "Nonintrusive Load-Shed Verification," *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 49–57, jan.-march 2011.
[18] M. El Guedri, G. D'Urso, C. Lajaunie, G. Fleury *et al.*, "Time-frequency characterisation for electric load monitoring," 2009.
[19] H. Murata and T. Onoda, "Applying kernel based subspace classification to a non-intrusive monitoring for household electric appliances," in *Proc. 11th International Conference on Artificial Neural Networks*, 2001.
[20] M. LeMay and C. Gunter, "Cumulative attestation kernels for embedded systems," *Computer Security–ESORICS 2009*, pp. 655–670, 2009.
[21] R. Berthier and W. Sanders, "Specification-based intrusion detection for advanced metering infrastructures," in *IEEE Pacific Rim International Symposium on Dependable Computing*, 2011, pp. 184–193.
[22] P. Jokar, H. Nicanfar, and V. Leung, "Specification-based intrusion detection for home area networks in smart grids," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, Oct. 2011, pp. 208 –213.
[23] Electricity thefts surge in bad times. Available at http://www.usatoday.com/money/industries/energy/2009-03-16-electricity-thefts_N.htm.
[24] E. Quinn, "Smart metering and privacy: Existing law and competing policies," A report for the Colorado Public Utilities Commission, 2009.
[25] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 2010, pp. 61–66.
[26] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security and Privacy*, vol. 8, no. 1, pp. 11–20, 2010.
[27] A. Rial and G. Danezis, "Privacy-Preserving Smart Metering," Microsoft Research, Tech. Rep. MSR-TR-2010-150, 2010.
[28] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
[29] A. Cassandra, "Exact and approximate algorithms for partially observable markov decision processes," Ph.D. dissertation, Brown University, 1998.
[30] M. Armstrong, M. Swinton, H. Ribberink, I. Beausoleil-Morrison, and J. Millette, "Sythetically derived profiles for representing occupant-driven electric loads in canadian housing," *J. Building Performance Simulation*, vol. 2, no. 1, pp. 15–30, 2009.
[31] C. Goh and J. Apt, "Consumer strategies for controlling electric water heaters under dynamic pricing," *Carnegie Mellon Electricity Industry Center Working Paper*, 2004.
[32] T. Yardley, "Keynote address: Developing a testbed for the smart grid," in *IEEE Conference on Local Computer Networks*, 2010, pp. 1–1.
[33] C. Lee, H. Yang, B. Lee, and D. Won, "A novel privacy-enhanced ami system using searchable and homomorphic encryption techniques," in *Convergence and Hybrid Information Technology*. Springer Berlin / Heidelberg, 2012, vol. 7425, pp. 608–617.

Stephen McLaughlin is a Ph.D. candidate in computer science and engineering at the Pennsylvania State University under the advising of Patrick McDaniel. His dissertation is on automation and detection of attacks against closed-loop control systems used in critical infrastructure. He has also done security assessments of real-world Advanced Metering Infrastructure systems.

Brett Holbert received his B.S. in Computer Science from University of Maryland, College Park in 2010. He is currently a Ph.D. candidate in Computer Science and Engineering at the Pennsylvania State University. As a member of the Networking and Security Research Center he is exploring topology inference in partially responding computer networks. Additional research interests include network fault diagnosis and failure recovery.

Ahmed Fawaz is a graduate student at the Coordinated Science Laboratory, University of Illinois at Urbana Champaign. He received his B.E. in electrical and computer engineering in 2011 from the American University of Beirut. Currently, he is working on intrusion resilience in the future smart grid through automated response and recovery using techniques from control theory, game theory, hybrid systems and machine learning.

Robin Berthier is a research scientist at the University of Illinois at Urbana-Champaign, working with Prof. William H. Sanders. Robin graduated from the Reliability Engineering Department at the University of Maryland in 2009. His doctoral dissertation with Prof. Michel Cukier, and focused on the issue of honeypot sensors deployed on large networks. He introduced a new architecture to increase the scalability of high-interaction honeypots, and combined network datasets of different granularities to offer unique attack forensics capabilities. His current research interests include advanced intrusion detection systems and the security of critical infrastructures.

Saman Zonouz is an Assistant Professor in the Electrical and Computer Engineering Department at the University of Miami. He received his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2011. He has worked on intrusion response and recovery, information flow-based security metrics for power-grid critical infrastructures, online digital forensics analysis and monitorless recoverable applications. His research interests include: computer security and survivable systems, control/game theory, intrusion response and recovery systems, automated intrusion forensics analysis, and information flow analysis-based security metrics.