The Impact of Stealthy Attacks on Smart Grid Performance: Tradeoffs and Implications

Yara Abdallah, *Student Member, IEEE,* Zizhan Zheng*, *Member, IEEE,* Ness B. Shroff, *Fellow, IEEE,* Hesham El Gamal, *Fellow, IEEE,* and Tarek M. El-Fouly, *Member, IEEE*

***Abstract*—The smart grid is envisioned to significantly enhance the efficiency of energy consumption, by utilizing two-way communication channels between consumers and operators. For example, operators can opportunistically leverage the delay tolerance of energy demands in order to balance the energy load over time, and hence, reduce the total operational cost. This opportunity, however, comes with security threats, as the grid becomes more vulnerable to cyber-attacks. In this paper, we study the impact of such malicious cyber-attacks on the energy efficiency of the grid in a simplified setup. More precisely, we consider a simple model where the energy demands of the smart grid consumers are intercepted and altered by an active attacker before they arrive at the operator, who is equipped with limited intrusion detection capabilities. We formulate the resulting optimization problems faced by the operator and the attacker and propose several scheduling and attack strategies for both parties. Interestingly, our results show that, as opposed to facilitating cost reduction in the smart grid, increasing the delay tolerance of the energy demands potentially allows the attacker to force increased costs on the system. This highlights the need for carefully constructed and robust intrusion detection mechanisms at the operator.**

I. Introduction

Over the past few years, the smart grid has received considerable momentum, exemplified in several regulatory and policy initiatives, and research efforts (see for example [2], [3] and the references therein). Such research efforts have addressed a wide range topics spanning energy generation, transportation and storage technologies, sensing, control and prediction, and cyber-security [4], [5], [6], [7], [8].

Demand response/load balancing and energy storage are two promising directions for enhancing the energy efficiency and reliability in the smart grid. Non-emergency demand response has the potential of lowering real-time electricity prices and reducing the need for additional energy sources. The basic idea is that, by utilizing two-way communication channels, the *emergency level* of each energy demand (at the end-users or central distribution stations) is sent to the grid operator that, in turn, *schedules* these demands in a way that *flattens* the load. Moreover, energy storage capabilities at the end-points offer more degrees of freedom to the operator, allowing for a higher efficiency gain. This potential gain, however, comes at the expense of the security threat posed by the vulnerability of the communication channels to interception and impersonation.

In this work, we study the impact of the vulnerability of two-way communications on the energy efficiency of the smart grid. More specifically, we propose a new type of data integrity attack towards Advanced Metering Infrastructures (AMI), that captures the above scenario in the presence of a single *stealthy* attacker. In an AMI system, a wide area network (WAN) connects utilities to a set of gateways, which are connected to electricity meters through neighborhood area networks (NANs). As observed in [9], neighborhood area networks is an attractive target of attacks, where a large number of devices are physically accessible with little security monitoring available. Moreover, since these derives are connected to networks, an attacker can potentially get access to a large amount of data by hacking into a few nodes or links in AMI [10]. As observed in [11], all the three major types of nodes in AMI, namely, smart meters, data concentrators, and the AMI headend, are subject to attacks, with different amount of data that can be utilized by the attacker.

In this work, we consider a simplified model of AMI, similar to [12], that includes a grid operator and n consumers that may be capable of energy storage, harnessing the potential cost savings in the smart grid. Our analysis covers two models of energy demands. In the first (total-energy model), each demand includes the total amount of energy to be served, the service start time, and the *deadline* by which the requested energy should be delivered. In the second (constant-power model), each demand similarly has an arrival time and deadline, but the consumers ask for energy to be distributed across a specified number of time slots (a service time), with a power requirement in each slot. In both models, the consumers send their demands over separate communication channels to the operator. The grid operator attempts to schedule these demands so as to balance the load across a finite period of time, and hence *minimize* the total cost paid to serve these demands.

In our model, we also assume the presence of a single attacker who is fully capable of intercepting and altering the consumer demands before they arrive at the operator (see Figure 1). The end goal of the attacker, as opposed to the operator, is to *maximize* the operational cost paid by the system for these demands, hence reducing the energy efficiency of

Y. Abdallah and H. E. Gamal are with the Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Ave., Columbus, OH 43210, USA. Email: yara.abdallah10@gmail.com; helgamal@ece.osu.edu.

Z. Zheng is with the Department of Computer Science, University of California, Davis, 1 Shields Ave., Davis, CA 95616, USA. Email: cszheng@ucdavis.edu.

N. B. Shroff is with the Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Ave., Columbus, OH 43210, USA. Email: shroff.11@osu.edu.

T. M. El-Fouly is with the Department of Computer Science and Engineering, Qatar University, PO Box 2713, Doha, Qatar. Email: tarekfouly@qu.edu.qa.

* Corresponding author.

This work was supported by QNRF fund NPRP 5-559-2-227 and ARO-W911NF-15-1-0277. A preliminary version of this work appeared in the proceedings of the IEEE Conference on Decision and Control, 2012 [1].

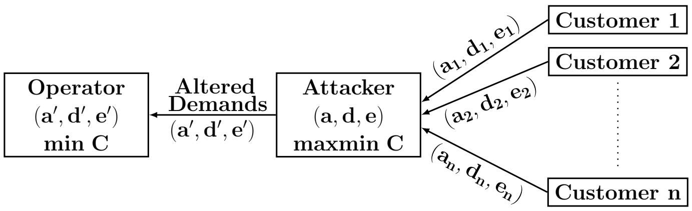


Fig. 1: A system model for a smart grid in the presence of a single attacker. The forward channels between the consumers and the grid operator are fully compromised by the attacker. (a, d, e) is the vector of the start times, deadlines and energy requirements of the consumer demands, respectively.

the system. We differentiate between two scenarios. The first corresponds to a naive operator who fully trusts the incoming energy demands, whereas in the second, a simple intrusion detection mechanism (to be discussed later) is assumed to be deployed by the operator. Rather intuitively, the attacker's desire to remain undetected imposes more limitations on its capabilities, and hence, reduces the potential harm. This desire can be justified, for example, by considering the long-term performance of the grid, where the total impact of successive attacks is more damaging when the attacker remains undetected.

Based on the aforementioned assumptions, we first formulate the optimization problems faced by the operator and the attacker. For the operator, when being oblivious to any attacks, a minimization problem needs to be solved. On the other hand, the attacker is aware of the optimal strategy employed by the operator, and hence, a maximin optimization problem needs to be solved. In our formulation, we limit the attack's strength by the number of energy demands the attacker is capable of altering. For the case when the attacker is capable of altering *all* of the energy demands (the attacks thus reach their full potential and force the system to operate at the maximum achievable total cost), we show that the maximin problem actually reduces to a maximization problem.

To the best of our knowledge, however, the impact of stealthy attacks on the energy efficiency of the smart grid has not been studied before, and this paper is the first attempt to *explicitly* characterize such impact.

Our main contribution can be summarized as follows.

- We propose optimal offline strategies for both the operator and the unlimited attacker. The former gives the minimum energy cost when there is no attack, and the latter gives the maximum energy cost that can be enforced. The gap between the two indicates the maximum damage that can result from such attacks. We also provide efficient online strategies for both of them. These strategies are more practical in terms of operability and also indicate several bounds on the possible damage due to an unlimited attack.
- For more limited attacks, we provide a simple greedy offline algorithm to arrive at a lower bound, and a dynamic programming-based algorithm that computes an upper bound on the total cost achieved by such attacks. Moreover, efficient online attacks are provided.
- From our analysis and numerical results, we conclude that **in the absence of security threats** an increase in the delay tolerance of the energy demands increases the energy efficiency of the system, as expected, since the operator is offered more scheduling opportunities. On the other hand, a somewhat surprising observation is that, with a limited defense mechanism at the operator, this increase offers a similar opportunity to the attacker to force costs **even higher than those incurred by the regular grid**, transposing the purpose of the communication capabilities provided to the consumers.

The proposed framework enjoys several merits. Our analysis throughout the sequel does not assume any specific structure/distribution on the consumer demands and hence the derived results encompass a wide range of realistic scenarios. The attack bounds provided here are based on worst-case analyses and so provide strong guarantees on the impacts of different attacks. The main limitation of this work is the rather weak detection/defense mechanism at the operator. Our purpose here is to explore the attacker's side and arrive at performance bounds that motivate stronger defenses at the operator/consumers.

The remainder of this paper is organized as follows. After a brief overview of related work in Section II, we present our system model and the optimization problems at the operator and attacker sides in Section III. In Sections IV and V, we provide offline and online attacks for the total-energy model and the constant-power model, respectively. Numerical results are given in Section VI. We provide some suggestions to the operator in Section VII, whereas our conclusions are given in Section VIII. Extensions and missing proofs are provided in our online technical report [13].

II. Related Work

Cybersecurity is of critical importance to the secure and reliable operation of the smart grid, which is challenging to achieve due to the large scale and the decentralized nature of the grid, the heterogeneous requirements of the components, and the coupling of the cyber and physical systems. Various types of cyber attacks targeting the availability, integrity, and confidentiality of the smart grid have been studied, and both prevention and detection techniques have been proposed [14], [9].

Data integrality attack is considered as an important threat to the smart grid [14]. In particular, false data injection towards the SCADA systems has received a lot of attention recently [5], [6]. By injecting malicious data into a small set of controlled meters, this attack can bias the state estimation of the system while bypassing the bad data detection in the current SCADA systems. Since the seminal work of [5], much effort has been devoted to the problem of finding the minimum number of meters to be controlled to ensure undetectability [6], [8]. Although this sparsest unobservable data attack problem is NP-hard in general, a polynomial time solution is given in [15] for the case when the network is fully measured. Moreover, strategic defense techniques have been developed [8], [16] and the impact of data injection attack on real-time electricity market has been considered [7], [17]. When the attacker does not have enough number of controlled meters, a generalized likelihood ratio test is proposed to detect attacks [6]. In addition to data attacks, the sparsest unobservable attack problem has been studied in closely related power injection attacks [18].

In the context of AMI, various potential threats have been identified [11], [10], [14], including integrity attacks for the purposes such as energy theft and remote disconnection. Different intrusion detection systems (IDS) have been considered, including specification based [10], [19] and anomaly based approaches [20]. In particular, a set of data stream mining algorithms are evaluated and their feasibility for the different components in AMI is discussed in [20]. The information requirements for detecting various types of attacks in AMI are discussed in [9]. Although data integrity attacks are considered as a potential threat in AMI [14], its impact on the energy efficiency of the system has not been considered before, and proper intrusion detection schemes for the new type of attack that we consider remain open.

III. Problem Formulation

A. Demand Model

In this paper, we adopt the control and optimization framework first proposed in [12] for the demand side of the smart grid. This framework assumes a central operator and n energy consumers that send their energy service demands to the operator using perfect channels. Our model builds on this framework, adding to it a single active attacker. The attacker is capable of intercepting and altering the demand requests in order to maximize the total energy cost paid by the smart grid. We assume a time-slotted system and a finite time horizon $[0, T]$, and consider two types of demands:

1) Total-Energy requirements: each consumer has a total *energy* requirement that needs to be served before some deadline elapses. This, for instance, captures the scenario of having consumers with energy storage capabilities. Here, the energy demand of the j^{th} consumer, $1 \leq j \leq n$, is composed of the tuple (a_j, d_j, e_j), where $a_j, d_j \in \mathbb{N}^+, d_j \geq a_j, e_j \in \mathbb{R}^+$, indicating that demand j arrives at the beginning of time-slot a_j, and has to be served for a total amount of e_j, by the end of time-slot d_j.
2) Constant-Power requirements: each consumer has an instantaneous power requirement and specifies a service time duration to finish a given job [1], before a deadline elapses as well. The energy demand of the j^{th} consumer, $1 \leq j \leq n$, is composed of the tuple (a_j, d_j, s_j, p_j), where a_j and d_j are defined as in the above, $s_j \in \mathbb{N}^+$ is the job's duration time and $p_j \in \mathbb{R}^+$ is the instantaneous power requirement for this job. We note that in contrast to the total-energy model, the instantaneous power requirement p_j cannot be changed by either the operator or the attacker.

In both cases, we assume that the set of jobs can be scheduled preemptively, i.e., a job can be interrupted and resumed, so long as the deadline and energy/power requirements are met. Let $J = \{1, \ldots, n\}$, and the associated demands in the total-energy (constant-power) model in $\mathcal{J} = \{(a_1, d_1, e_1), \ldots, (a_n, d_n, e_n)\}$ ($\mathcal{J} = \{(a_1, d_1, s_1, p_1), \ldots, (a_n, d_n, s_n, p_n)\}$). The set of demands are sorted by their arrival times non-decreasingly. Each energy demand in $\mathcal{J}$ is sent to the operator over a perfect channel that is fully intercepted by the attacker. Hence the attacker could substitute the actual demand set $\mathcal{J}$ by a forged one, $\mathcal{J}'$, before it is received by the operator. An example for the total-energy case is shown in Figure 1. Similar to $\mathcal{J}$, the forged set is $\mathcal{J}' = \{(a'_1, d'_1, e'_1), \ldots, (a'_m, d'_m, e'_m)\}$ for the total-energy model and, for the constant-power model, is $\mathcal{J}' = \{(a'_1, d'_1, s'_1, p'_1), \ldots, (a'_m, d'_m, s'_m, p'_m)\}$. Let $J' = \{1, \ldots, m\}$ denote the indices of forged jobs. We note that $m \geq n$ in general as will be explained later in this section. For ease of notation, we define the vector $a = [a_1, \ldots, a_n]$, and define d, e, s, p similarly for the original vectors, and define a', d', e', s', p' for the corresponding forged vectors. For any job j, we define its *job allowance* to be $l_j = d_j - a_j + 1$. Let $l_{max} = \max_{j \in J} l_j$, $l_{min} = \min_{j \in J} l_j$. We similarly define $e_{max}, e_{min}, s_{max}, s_{min}, p_{max}, p_{min}$.

B. Simple Intrusion Detection

We put the following constraints on the attacker. First, when the attacker chooses a job to modify, he is limited to changing its arrival time or its deadline time, or, breaking the job into multiple separate jobs (that would appear to the scheduler as independent jobs), so long as the final schedule is *admissible*. That is, all of the original jobs are served exactly their energy requirement (or service time and power requirement) upon or after their arrival and before or upon their real deadlines. Note that the attacker could easily be detected by the consumers if the final schedule is **not** admissible.

Moreover, we assume that the operator adopts a simple statistical testing based intrusion detection scheme. For example, consider a statistical testing on the slackness of jobs. The slackness of a job j, denoted as x_j, is defined as the maximum time elasticity when serving the job. Formally, $x_j = l_i - 1$ for the total-energy model, and $x_j = l_j - s_j$ for the constant-power model. Assume that the slackness of demands are $i.i.d.$ samples of a known distribution with mean μ and variance σ^2. For a set of n demands received, the operator determines if it has been modified by using, for example, the one sample z-test with statistic $z = \frac{\overline{x} - \mu}{\sigma} \sqrt{n}$, with a significance level α, the probability threshold below which the operator decides the data has been modified.

Assume that the attacker knows (1) the distribution of demands, and (2) the statistical testing and α adopted by the operator. If the attacker also knows $\sum_j x_j$ for the set of demands in J, it can find the maximum amount of job slackness that can be reduced for demands in J, while still passing the z-test on slackness. When this knowledge is not available as in the more realistic online setting (to be precisely defined below), the attacker cannot ensure undetectablility. However, it can choose to modify a small number of jobs to ensure a small probability of detection, which is still useful to the attacker. We can similarly consider a statistical testing on the arrive times or other parameters of demands. Instead of working on a constraint that depends on the concrete statistical testing used, we consider a simple constraint on the fractional of energy demands that the attacker is capable of

[1] We use demand and job interchangeably in the paper.

altering, which can be derived from the statistical testing used. In addition to simplifying the optimization problems for the attacker, such a bound can also be interpreted as a resource constraint to the attacker. We will consider other types of constraints in our future work. Let $B = \lfloor \beta n \rfloor, \beta \in [0,1]$ denote the number of jobs that the attacker can modify.

We note that an accurate statistical modeling of electric demands with time elasticity is by itself a challenging problem especially when the demands are correlated, which provides further opportunity to the attacker. Although the operator can also consider more advanced intrusion detection schemes such as data mining based anomaly detection, the high dimension of the data stream (large number of demands with overlapping durations) is a big challenge to be addressed.

C. Optimization at the Operator and the Attacker

Upon receiving the m (altered) demands, $\mathcal{J}'$, an admissible schedule of these demands (jobs) is to be determined by the operator. A schedule is given by $\mathcal{S} = [\mathcal{S}]_{jt} \in {\mathbb{R}^+}^m \times {\mathbb{R}^+}^T$, where $\mathcal{S}_{jt}$ denotes the amount of energy allocated to job j in time-slot t. Let $E_{\mathcal{S}}(t)$ be the total energy consumed at time-slot $t \in [0,T]$ under schedule $\mathcal{S}$, i.e., $E_{\mathcal{S}}(t) = \sum_{j \in J'} \mathcal{S}_{jt}$. Let $C_t(E_{\mathcal{S}}(t))$ denote the cost incurred by the total power consumed at the time-slot t. We assume $C_t : \mathbb{R}^+ \to \mathbb{R}^+$ to be a general non-decreasing and convex function, as in [12]. The convexity assumption resembles the fact that, as the demand increases, the differential cost at the operator increases, i.e., serving each additional unit of energy to increasing demand becomes more expensive [12]. In our analysis and evaluation, we will consider the following commonly adopted power function as an example, where $C_t(E) = E^b, b \in \mathbb{R}, b \geq 1$, which allows for estimating the performance for a wide range of monotone increasing and convex functions. Moreover, for simplicity of exposition, we assume $C_t(\cdot)$ to be time invariant in the following and omit the subscript t. We show that most of our algorithms and analytic results can be extended to time-dependent cost functions in our technical report [13].

The operator attempts to balance the load by finding an admissible schedule (given the altered demands by the attacker) that minimizes the total cost over the interval $[0,T]$. The optimization problem at the operator side, for the total-energy model, is then defined as follows:

$$\begin{aligned} C_{min}(a',d',e') = \min_{\mathcal{S}} & \sum_{t=1}^{T} C(E_{\mathcal{S}}(t)) \\ \text{s.t.} \quad & \mathcal{S}_{jt} \geq 0, && \forall j \in J', \forall t \in [0,T], \\ & \sum_{t=a'_j}^{d'_j} \mathcal{S}_{jt} = e'_j, && \forall j \in J'. \end{aligned} \tag{PminE}$$

where we have dropped the constraint that no energy is served to a job j outside $[a_j, d_j]$ since $C(\cdot)$ is monotone increasing. Similarly, the problem for the constant-power model is

$$\begin{aligned} C_{min}(a',d',s',p') = \min_{\mathcal{S}} & \sum_{t=1}^{T} C(E_{\mathcal{S}}(t)) \\ \text{s.t.} \quad & \mathcal{S}_{jt} \in \{0, p'_j\}, && \forall j \in J', \forall t \in [0,T], \\ & \sum_{t=a'_j}^{d'_j} \mathbf{1}_{\mathcal{S}_{jt}=p'_j} = s'_j, && \forall j \in J'. \end{aligned} \tag{PminS}$$

where $\mathbf{1}_{\mathcal{S}_{jt}=p'_j} = 1$ if $\mathcal{S}_{jt} = p'_j$, and is 0 otherwise. The constraints in the both problems ensure the admissibility of the considered schedules.

On the other hand, the attacker attempts to find appropriate values of a', d', e' (or a', d', s', p') in $\mathcal{J}'$ such that the cost achieved by the operator is maximized, subject to the number of demands that can be modified. Let b_j be the collection of the (sub)jobs that the attacker generates out of job j, $1 \leq j \leq n$. Each (sub)job is, again, a tuple of the form (a', d', e') or (a', d', s', p'). To guarantee an admissible final schedule, each set b_j should satisfy the following conditions:

In the total-energy model, for each job j:
For $1 \leq k \leq |b_j|$

$$a'_k, d'_k \in \mathbb{N}^+, e'_k \geq 0, \tag{1a}$$

$$a_j \leq a'_k \leq d'_k \leq d_j, \tag{1b}$$

$$\sum_{1 \leq k \leq |b_j|} e'_k = e_j. \tag{1c}$$

In the constant-power model, for each job j:
For $1 \leq k \leq |b_j|$

$$a'_k, d'_k, s'_k \in \mathbb{N}^+, \tag{2a}$$

$$a_j \leq a'_k \leq d'_k \leq d_j, p'_k = p_j, \tag{2b}$$

$$\sum_{1 \leq k \leq |b_j|} s'_k = s_j. \tag{2c}$$

$$[a'_k, d'_k] \cap [a'_l, d'_l] = \emptyset, \quad \forall k, l, k \neq l, 1 \leq k, l \leq |b_j|. \tag{2d}$$

The sets b_j are then collected in the forged demand vector, i.e., $\mathcal{J}' : = \bigcup_{1 \leq j \leq n} b_j$. Under this setting, the attacker solves the following optimization problems. For the total-energy model:

$$\begin{aligned} C_{maxmin}(a,d,e,\beta) = & \max_{a',d',e',J^*} C_{min}(a',d',e') \\ \text{s.t.} \quad & \text{Eqs (1a) - (1c)}, \\ & |J^*| \leq \beta n, \end{aligned} \tag{PmaxminE}$$

where $\beta \in \mathbb{R}, 0 \leq \beta \leq 1$, and

$$J^* = \{ j \in J \colon b_j \neq \{(a_j, d_j, e_j)\} \}. \tag{3}$$

Here J^* denotes the set of consumer job indices that were **modified** by the attacker. In a similar fashion, we define the attacker's optimization problem for the constant-power model:

$$\begin{aligned} C_{maxmin}(a,d,s,p,\beta) = & \max_{a',d',s',p',J^*} C_{min}(a',d',s',p') \\ \text{s.t.} \quad & \text{Eqs (2a) - (2d)}, \\ & |J^*| \leq \beta n, \end{aligned} \tag{PmaxminS}$$

where $\beta \in \mathbb{R}, 0 \leq \beta \leq 1$, and

$$J^* = \{j \in J \colon b_j \neq \{(a_j, d_j, s_j, p_j)\}\}. \tag{4}$$

We provide efficient *offline* and *online* solutions to the problems formulated above. Offline solutions not only give us performance bounds on the extreme case when there is no uncertainty on energy demands, but also provide useful insights for the design of online solutions. On the other hand, in the more realistic online setting, a demand is revealed only on its *actual* arrival.

- Offline setting: In the offline setting, we assume that the attacker knows all the true demands $\mathcal{J}$ at time 0, while the operator knows all the forged demands $\mathcal{J}'$ at time 0, and obtains no further information during $[0, T]$.
- Online setting: In the online setting, at any time t, the attacker only knows the set of true demands with $a_j \leq t$, while the operator only knows the set of unmodified demands with $a_j \leq t$, and the set of forged demands with $a'_j \leq t$. In addition, the number of demands n is the common knowledge.

Note that in the online setting, if $a'_j = a_j$, demand j should be forwarded to the operator without delay. On the other hand, if $a'_j > a_j$, the attacker should hold demand j until a'_j so that the operator does not get extra information.

For comparison purposes, we also consider the following inelastic scheduling policy for the operator as a baseline strategy. In the total-energy model, this strategy serves each job its energy demand, entirely and immediately upon its arrival. The associated baseline cost, $C_{base}(a, d, e)$, can be found as:

$$C_{base}(a, d, e) = \sum_{t \in [0,T]} C\left(\sum_{j \in J : a_j = t} e_j\right). \tag{5}$$

The counterpart quantity in the constant-power model is:

$$C_{base}(a, d, s, p) = \sum_{t \in [0,T]} C\left(\sum_{j \in J : t \in [a_j, a_j + s_j - 1]} p_j\right). \tag{6}$$

This strategy represents the case when the delay tolerance of the jobs is not exploited. Therefore, we treat this quantity as the cost paid in the *current regular gird*, where no two-way communication channels are established, and accordingly, the system is not vulnerable to the cyber-attacks discussed in this paper.

As a first attempt towards understanding the impact of stealthy attacks on smart-grid demand-response, we have made several simplifications in this work. In our technical report [13], we provide a discussion on the rationale behind our model and outline several extensions including how to conduct the impact analysis under congested power systems.

IV. TOTAL-ENERGY DEMANDS: SCHEDULING AND ATTACK STRATEGIES

In this section, we focus on the total-energy demand model. We first find the optimal scheduling strategy for the operator in Section IV-A. We next propose full attack strategies in Section IV-B including both offline and online attacks. Finally, in Section IV-C, we propose limited attacks and study the impact of such attacks. We note that the offline attacks we discuss below have a time complexity of $O(n^3)$. On the other hand, all the online attacks have a time complexity of $O(n)$ and are therefore more scalable to large systems.

A. Scheduling at the Operator

The optimization problem at the operator (PminE) can be directly mapped to the "minimum-energy CPU scheduling problem" studied in [21]. Our discussion below is an adapted discrete-time version of the classical YDS algorithm [21].

For every pair $(k, l), k \leq l$, let $\mathcal{I}_J(k, l)$ be the set of all job indices whose intervals are entirely contained in $[k, l]$, that is, $\mathcal{I}_J(k, l) = \{j \in J \colon a_j \geq k, d_j \leq l\}$. For the received (forged) demands $\mathcal{J}'$, define the *energy intensity* on $\mathcal{I}_{J'}(k, l)$ to be

$$g(\mathcal{I}_{J'}(k, l)) = \frac{\sum_{j \in \mathcal{I}_{J'}(k,l)} e'_j}{l - k + 1}, \tag{7}$$

Note that if we only consider the set of jobs in $\mathcal{I}_{J'}(k, l)$, a schedule that serves $g(\mathcal{I}_{J'}(k, l))$ amount of electricity in each time slot in the interval $[k, l]$ minimizes the energy cost (assuming it is admissible). We further define $(k^*, l^*) = \text{argmax}_{(k,l):k \leq l}\, g(\mathcal{I}_{J'}(k, l))$, that is, $\mathcal{I}_{J'}(k^*, l^*)$ is the set of jobs with the maximum energy intensity among all $\mathcal{I}_{J'}(k, l)$ for any k, l with $k \leq l$.

It is shown in [21] that, for strictly convex $C(\cdot)$, the optimal strategy schedules a total energy of $g(\mathcal{I}(k^*, l^*))$ in each time slot in $[k^*, l^*]$. That is, the interval with the maximum energy intensity must maintain this intensity in the optimal schedule. This also implies that no jobs out of $\mathcal{I}(k^*, l^*)$ are scheduled with those in $\mathcal{I}(k^*, l^*)$. Hence a greedy algorithm that searches for $\mathcal{I}(k^*, l^*)$, schedules the jobs in $\mathcal{I}(k^*, l^*)$ and then removes those jobs (and the corresponding interval) from the problem instance, can be used to solve Problem (PminE). The corresponding algorithm is outlined below (see [21] for the details).

Algorithm 1 Offline Scheduling at the Operator

```
while $J' \neq \emptyset$ do
    $\mathcal{I}_{J'}(k^*, l^*) \leftarrow$ an interval with the highest energy intensity;
    Schedule the jobs in $\mathcal{I}_{J'}(k^*, l^*)$ according to the Earliest Deadline First (EDF) policy, such that $E_{\mathcal{S}}(t) = g(\mathcal{I}_{J'}(k^*, l^*))$, for all $t \in [k^*, l^*]$;
    Delete the jobs in $\mathcal{I}_{J'}(k^*, l^*)$ from $J'$ and modify the problem to reflect the deletion of jobs.
```

The above algorithm arrives at the optimal schedule with complexity $O(n^3)$ since it suffices to consider intervals whose two endpoints are either arrival times or deadlines of some jobs. Let C_{min} denote the optimal minimum cost achieved (when there is no attack). A simple online algorithm for Problem (PminE) was also given in [21] (the Average Rate Heuristic, AVR). This online scheme distributes the energy requirement of each job *evenly* on its service interval, ignoring further information on how the jobs intersect. The performance of this simple heuristic is studied in [21] when the cost mapping is a power function, and the following bounds are proven: For $C(E) = E^b, b \in \mathbb{R}, b \geq 2$, this online heuristic achieves a total cost $\overline{C}_{min} \leq r_b C_{min}$, where $b^b \leq r_b \leq 2^{b-1} b^b$. Since each demand is processed once, this algorithm has an $O(n)$ complexity.

B. Full Attack Strategies and Performance Bounds

We now turn our attention to the attacker's selection of $\mathcal{J}'$. We note that the special case ($\beta = 1$) is of special interest to us, as it resembles a *full* attack, i.e., the attacker is capable of modifying *all* of the consumer demands (e.g., when there is no intrusion detection at the operator). We first address this case. The more general attacks for $\beta < 1$ will be considered in Section IV-C.

1) An Optimal Offline Full Attack: We first show that, in the case $\beta = 1$, the Problem (PmaxminE) can be transformed into a *maximization* problem. To see this, consider any undetectable strategy followed by the attacker such that, for each demand $(a_j, d_j, e_j) \in \mathcal{J}$, there exists exactly one corresponding forged demand, $(a'_j, d'_j, e'_j) \in \mathcal{J}'$, with $a'_j = d'_j = t_j$ for some $t_j \in [a_j, d_j]$, and $e'_j = e_j$. All such strategies are always feasible to the attacker by our assumption of $\beta = 1$ and, if employed by the attacker, leave no degrees of freedom to the operator. Moreover, due to the monotonicity and convexity of $C(\cdot)$, it suffices for the attacker to consider only this set of strategies as shown in the following lemma (see [13] for the proof).

Lemma 1: When $\beta = 1$, there is an optimal attack where for any job j, $a'_j = d'_j = t_j$ for some $t_j \in [a_j, d_j]$.
Based on this observation, Problem (PmaxminE) under $\beta = 1$ reduces to a maximization problem, which, for a given job instance, looks for an optimal strategy that serves each job in a *single* feasible time slot. Formally, the attacker solves the following problem:

$$C_{max}(a, d, e) = \max_{\mathcal{S}} \sum_{t=1}^{T} C(E_{\mathcal{S}}(t)) \tag{Pmax}$$
$$\text{s.t.} \quad \mathcal{S}_{jt} = 0, \quad \forall j \in J, \forall t \in [0, T], t \neq t_j,$$
$$\mathcal{S}_{jt_j} = e_j, t_j \in [a_j, d_j], \qquad \forall j \in J.$$

Hence, in the above formulation, the attacker needs to decide only on t_j for each $j \in J$. Given a set of jobs J, define a *clique* of J as a subset of jobs in J whose job intervals intersect with each other, and a *clique partition* of J as a partitioning of set J into disjoint subsets where each subset forms a clique of J. We then have the following observation (see [13] for the proof).

Lemma 2: Each clique partition of J corresponds to a feasible solution to Problem (Pmax) and vice versa.

Moreover, we observe that to find the optimal attack, it is sufficient to consider *locally maximal cliques* defined as follows. For any time slot t, let K^t denote the set of jobs whose job interval contains t. A clique is called *locally maximal* if it equals K^t for some t. The following result is key to derive the optimal attack (see [13] for the proof):

Lemma 3: There is an optimal clique partition solving (Pmax) that contains a locally maximal clique [2].

Let $\overline{C}(k, l)$ be the maximum feasible cost that could be achieved by solely scheduling the jobs in $\mathcal{I}_J(k, l)$. Given any time-slot z contained in $[k, l]$, let $K^z_{k,l}$ be the locally maximal clique at z for jobs restricted to $\mathcal{I}_J(k, l)$. We then have the following recursion.

Theorem 1:

$$\overline{C}(k, l) = \max_{z \in [k,l]} \left[C\left(\sum_{j \in K^z_{k,l}} e_j \right) + \overline{C}(k, z-1) + \overline{C}(z+1, l) \right]. \tag{8}$$

Proof: Consider the set of jobs in $\mathcal{I}_J(k, l))$. Lemma 3 implies that $\overline{C}(k, l)$ is achieved by a partitioning that contains a locally maximal clique for jobs in $\mathcal{I}_J(k, l))$. Each such clique *separates* the optimization problem into two subproblems for smaller intervals. By searching over all the locally maximal cliques over the interval $[k, l]$, $\overline{C}(k, l)$ can be achieved. ■

Accordingly, we can apply the dynamic programming algorithm in [22] to our problem as in Algorithm 2 (a formal description appears in [13]). The optimal cost is then $\overline{C}(1, T)$, which is computed in the final step together with the optimal clique partition. From the obtained clique partition, one can easily compute a set of time slots, $t_j, j \in J$ and set $a'_j = d'_j = t_j$, solving Problem (Pmax). The obtained schedule leaves no degrees of freedom to the operator as, after the attacker's modifications, all jobs become virtually urgent to operator and must be scheduled immediately upon their arrival. It is also clear that, as the job allowance of jobs increases, the attacker is capable of forming larger cliques and hence imposing higher costs on the operator. When we study online attacks, one of our goals is to formalize this observation.

Algorithm 2 Offline Full Attack

```
Iterate over all intervals $[k, l], k \leq l, k, l \in [0, T]$, with increasing interval length.
In each iteration, compute $\overline{C}(k, l)$ using Eq. (8), where the last two terms are obtained from previous iterations.
```

The algorithm has $O(n^2)$ iterations since it suffices to consider intervals whose two endpoints are either arrival times or deadlines of some jobs, where in each iteration, it takes $O(n)$ time to find $\overline{C}(k, l)$. Therefore, the algorithm has a total complexity of $O(n^3)$.

2) Online Full Attacks: In this section, we investigate the case where the attacker processes the arriving jobs in an online fashion, where at any time-slot t, the attacker possesses knowledge about the demands that have arrived by t. We propose a simple online attack where the jobs in J are partitioned into cliques according to an EDF policy. The attacker maintains a set of active jobs, that is, the set of demands that have arrived but not scheduled yet. Let A denote the set of active jobs. In any time-slot t, if t is the deadline for a demand $j \in A$, then all the active demands in A are grouped in a single clique, by setting their arrival times and deadlines to t. These demands are then forwarded to the operator, and A is set to the empty set. Note that the algorithm ensures that the operator only learns a demand j at a'_j. The intuition of using an EDF policy is to delay the decision as far as possible so that more demands can be compressed together to generate a large clique.

[2] A similar fact is proved in [22], where the authors consider clique partitioning so as to *minimize* a submodular cost function on the cliques, and shows the existence of a (globally) maximal clique in the optimal partition. We introduce the notion of locally maximal clique so that our results can be extended to time-dependent cost functions as we discuss in [13].

Algorithm 3 Online Full Attack

```
$A \leftarrow \emptyset$. In any time-slot $t$,
$A \leftarrow A \cup \{j : a_j = t\}$;
if $d_j = t$ for some job $j \in A$ then
    For each job $k$ in $A$, $a'_k \leftarrow t, d'_k \leftarrow t$;
    Forward the set of (forged) jobs in $A$ to the operator;
    $A \leftarrow \emptyset$
```

Since each job is processed once, the algorithm has a complexity of $O(n)$. We denote the resulting cliques by $K_1, \ldots, K_m$. The resulting cost is computed as

$$\underline{C}_{max} = \sum_{i=1}^{m} C\Bigg(\sum_{j \in K_i} e_j\Bigg). \tag{9}$$

Our next result shows that, despite its simplicity and online operation, Algorithm 3 could achieve a significant loss in the system's efficiency. We first make the following observation (see [13] for the proof).

Lemma 4: Consider any clique X in an optimal (offline) solution that achieves C_{max}. Then $X = \bigcup_i (X \cap K_i)$, where $X \cap K_i$ and $X \cap K_j$ are disjoint for $i \neq j$, and $X \cap K_i$ is non-empty for at most r_1 different K_i, where $r_1 = \left\lceil \frac{l_{\max}}{l_{\min}} \right\rceil + 1$.

This observation leads to the following bound for the online attack (see [13] for the proof).

Theorem 2: For $C(E) = E^b, b \in \mathbb{R}, b \geq 1$,

$$\underline{C}_{max} \geq \frac{1}{{r_1}^{b-1}} C_{max}, \text{ where } r_1 = \left\lceil \frac{l_{\max}}{l_{\min}} \right\rceil + 1 \tag{10}$$

When $C(.)$ is a power function of the form $C(E) = E^b, b \in \mathbb{R}, b \geq 1$, the simple structure of the online solution further delivers an explicit lower bound for the maximum achievable cost by the attacker (see [13] for the proof):

Theorem 3: For $C(E) = E^b, b \in \mathbb{R}, b \geq 1$,

$$C_{max}(a, d, e) \geq \left(\frac{l_{min} \sum_{j \in J} e_j}{2l_{min} + a_n - a_1} \right)^b. \tag{11}$$

The above result formalizes our intuition that the harm done by a cyber attack grows with the scheduling leverage given to the grid's operator. When all other parameters are fixed, we use this result to specifically estimate the growth of C_{max} with l_{min}. For instance, in Figure 2, we plot our bound versus an increasing l_{min}, fixing the average energy demand and the average inter-arrival time. In this instance, C_{max} grows at least linearly with l_{min}, and the rate of growth increases as the sample size n increases. More numerical results are reported in Section VI.

C. Limited Attacks and Performance Bounds

We now focus on the case when the attacker is limited by the number of jobs he is capable of modifying without being detected, i.e., the attacker can alter only $B = \lfloor \beta n \rfloor$ jobs, where $0 < \beta < 1$. We again divide our study in two cases, the offline setting and the online setting. In both cases, we derive bounds with respect to the corresponding full attacks. Therefore, these bounds are independent of the online scheduling algorithms used by the operator.

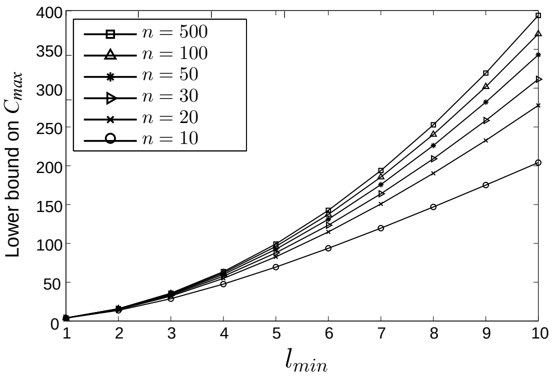


Fig. 2: A lower bound on C_{max} plotted for various values of n and l_{min} under a quadratic cost function (i.e., $b = 2$). The average energy demand is 10 while the average inter-arrival time is 5.

1) Offline Limited Attacks: For limited attacks, we are not able to find an optimal offline solution as we do for full attacks. To understand the impact of stealthy attacks in this more general setting, we propose two polynomial time offline algorithms that render a lower and an upper bound, respectively, on the performance of *optimal* limited attacks, and evaluate their performance in simulations. We show that even in the more challenging limited attack regime where the attacker may not be able to find the optimal attack, it is still possible to enforce significant amount of damage using a simple attack strategy.

Similar to our argument in the full attack case, the attacker could only consider the following simple strategy: Choose a set of job indices $J^* \subset J$ such that $|J^*| = \beta n$, and set $a'_j = d'_j = t^*_j$ for all the jobs $j \in J^*$. Leave all the remaining jobs $(J \backslash J^*)$ unaltered. We adopt this approach in our proposed offline attacks in this section. We let $C_{max} = C_{max}(a, d, e)$ and $C_{maxmin}(\beta) = C_{maxmin}(a, d, e, \beta)$, whenever clear from the context.

We first propose a simple variant that is tailored to our problem (see Algorithm 4). For any β, the algorithm finds a feasible limited attack, the cost of which provides a lower bound on the cost resulting from the optimal limited attack. We further establish an explicit performance bound for this algorithm in Theorem 4.

Our algorithm is inspired by the standard greedy algorithm for the fractional knapsack problem [23]. In the classical fractional knapsack problem, m items are given, each with a weight w_i and a value v_i. We need to find a set of items such that their total value is maximized subject to a budget on their total weight, say, $\beta_0 \sum_i w_i, 0 \leq \beta_0 \leq 1$. A fraction of any item might be collected, and the corresponding value is scaled according to its chosen weight. The greedy algorithm below solves this problem.

1) Sort (v_i, w_i) according to v_i/w_i non-increasingly.
2) Choose the first k pairs, $(v_1, w_1), \ldots, (v_k, w_k)$ such that

$$\sum_{i=1}^{k} w_i \leq \beta_0 \sum_{i=1}^{m} w_i \quad \text{and} \quad \sum_{i=1}^{k+1} w_i > \beta_0 \sum_{i=1}^{m} w_i. \tag{12}$$

The optimal choice is given by the k items collected in step (2), and a fraction of the $(k+1)^{th}$ item as the weight budget allows. Moreover, if we let the remaining weight

Algorithm 4 Offline Limited Attack

```
1: Find the optimal clique partitioning using Algorithm 2, assuming a full budget;
2: Sort the set of cliques found by the ratio of clique cost over clique size non-increasingly;
3: Greedily choose a set of cliques with total size bounded by $\beta n$; Let $K$ denote the first unchosen clique on the list;
4: Greedily choose $\min(\beta n, |K|)$ jobs of highest energy requirements in $K$ to compress;
5: Among the above two choices, the one that results in a higher cost is adopted.
```

budget after selecting the first k pairs be parameterized by $\beta_1 = \left(\beta_0 \sum_{i=1}^{m} w_i - \sum_{i=1}^{k} w_i\right)/w_{k+1}$, by the greedy selection, we must have

$$\sum_{i=1}^{k} v_i + \beta_1 v_{k+1} \geq \beta_0 \sum_{i=1}^{m} v_i. \tag{13}$$

The proposed attack strategy builds on the aforementioned algorithm:

Since finding the optimal clique partitioning is the most time-consuming step, the algorithm has a complexity of $O(n^3)$. Let $C^1_{maxmin}(\beta)$ denote the total cost enforced by this attack. It is clear that $C^1_{maxmin}(\beta) \leq C_{maxmin}(\beta)$. To get insights on the performance of this attack, we consider two special cases. Suppose that, under no budget constraints, the optimal clique partition (obtained from Algorithm 2) is composed of cliques of size one, i.e., each job forms a separate clique. In this case, our greedy attack will choose to fully compress $B = \beta n$ jobs, and those will be of the highest energy demands according to step (3) above. By the greedy selection, this clearly guarantees that $C^1_{maxmin}(\beta) \geq \beta C_{max}$. Another extreme case is when the optimal clique partition is composed of one single clique containing all of the n jobs. In this case, it is again clear by the greedy selection, in step (4), that $C^1_{maxmin}(\beta) \geq C\left(\beta \sum_{j \in J} e_j\right)$. When $C(.)$ is a power function of the form $C(E) = E^b, b \in \mathbb{R}, b \geq 1$, we get $C^1_{maxmin}(\beta) \geq \beta^b C_{max}$. For cases between those two extremes, we make use of the aforementioned insights to arrive at the following lower bound (see [13] for the proof).

Theorem 4: For $\beta \in [0, 1], C(E) = E^b, b \in \mathbb{R}, b \geq 1$,

$$C^1_{maxmin}(\beta) \geq \frac{\beta^b}{2} C_{max}. \tag{14}$$

We have further developed an algorithm that gives us an upper bound on the energy cost that can be achieved by any offline limited attacks. The details are given in our technical report [13].

2) Online Limited Attacks: To derive an efficient online limited attack, we consider the following simple strategy that mimics the behavior of Algorithm 3 while taking the budget constraint into account. As in Algorithm 3, the attacker maintains the set of active jobs in A. It also maintains the total number of jobs that have been modified in N, and the number of future jobs in R (recall that the attacker knows n). At any time t, the set of jobs that arrive at t are added to A. The main idea of the algorithm is to modify each job with probability β, or forward it to the operator directly with probability $1-\beta$, independent of other jobs. Note that this decision has to made at the arrival time of a job. Let $A' \subseteq A$ denote the set of active jobs to be modified. If there is a job j in A with $d_j = t$, then all the jobs in A' are compressed to the single time slot t. These jobs are then forwarded to the attacker, and both A and A' are set to the empty set. To make sure that all the budget is used and no more, the algorithm checks two boundary conditions. First, it stops sampling if all the budget has been used (lines 3-4). Second, when $R + N \leq B$, all the future jobs can be modified (line 6).

Algorithm 5 Online Limited Attack

```
$B \leftarrow \lfloor \beta n \rfloor, A \leftarrow \emptyset, A' \leftarrow \emptyset, N \leftarrow 0, R \leftarrow n.$
In any time-slot $t$,
1:  $A \leftarrow A \cup \{j : a_j = t\}$;
2:  for each job $j$ with $a_j = t$ do
3:      if $N = B$ then
4:          break;
5:      Sample $r$ from the uniform distribution in $[0, 1]$;
6:      if $r \leq \beta$ or $R + N \leq B$ then
7:          $A' \leftarrow A' \cup \{j\}$;
8:      else
9:          forward $j$ to the operator;
10:     $N \leftarrow N + 1, R \leftarrow R - 1$;
11: if $d_j = t$ for some job $j \in A$ then
12:     For each job $k$ in $A'$, $a'_k \leftarrow t, d'_k \leftarrow t$;
13:     Forward the set of forged jobs to the operator;
14:     $A \leftarrow \emptyset, A' \leftarrow \emptyset$
```

Since a separate decision is made for each demand on its arrival, and each demand to be modified is then processed once, this algorithm has a complexity of $O(n)$. Note that we have intentionally choose to generate the set of cliques at the earliest deadlines of jobs in A, not in A', so that this algorithm closely simulates the behavior of Algorihtm 3. In particular, consider an input sequence, and any clique K' generated by Algorithm 5, and the corresponding clique K generated by Algorithm 3 at the same time slot. Then $K' \subseteq K$. Moreover, for a set of *i.i.d.* demands, when n becomes large, for most cliques K, the corresponding K' has an expected size of $\beta|K|$. Although there is no guarantee on the worst-case performance, we expect that the algorithm achieves an expected cost that is at least a constant fraction of $\left(\beta \frac{e_{min}}{e_{max}}\right)^b \underline{C}_{max}$ for *i.i.d.* demands.

V. CONSTANT-POWER DEMANDS: SCHEDULING AND ATTACK STRATEGIES

Our previous scheduling and attack policies were derived solely for the total-energy demand model. In this section, we extend these results to demands that have service time and constant power requirements instead. We first provide an overview for the scheduling problem solutions at the operator in Section V-A. We then derive new full and limited attacks via simple modifications over the previously derived ones and analyze their performance in Sections V-B and V-C, respectively.

A. Scheduling at the Operator

When all of the consumers require the same amount of power per time slot (i.e., $p_j = p$ for all $j \in J$), the Problem (PminS) belongs to a class of "load balancing" problems that are studied in detail in [24]. In this work, the author shows that the problem of finding the optimal schedule is equivalent to a network flow problem with convex cost. An optimal solution can be obtained by an iterative algorithm followed by a rounding step [24]. For arbitrary power requirements, however, the integral nature of the problem renders it strongly NP-hard (see [13] for a proof).

Theorem 5: For the constant-power model, Problem (PminS) is strongly NP-hard.

In our simulations, we report the relaxed continuous-version solution (as given in [24]) as a lower bound to the achieved cost by the optimal scheduler. In this relaxed version, instead of a constant power p_j, job j can be served by an amount $p_{jt} \in [0, p_j]$ for any time-slot t such that $\sum_{t \in [a_j, d_j]} p_{jt} = s_j p_j$. We note that the continuous solution thus obtained can be furthered rounded to a feasible integral solution to the original problem. The main challenge, however, is to design the rounding process to achieve a low approximation factor, which remains open.

As for online algorithms for the operator, solutions with performance guarantee are unknown for preemptive demands. Two scheduling policies were provided for non-preemptive demands in [12]. We choose the Controlled Release (CR) policy in our simulations, which is shown to be asymptotically optimal as average deadline duration approaches infinity [12]. In the CR policy, an active demand is served if the instantaneous power consumption in the current time slot is below a threshold or if it cannot be further delayed. Since each demand j is processed at most l_j times, independent of other demands, the algorithm has a complexity of $O(n)$. Note that the online solution is always feasible and provides an upper bound to the offline optimal solution that is computationally hard to find.

B. Full Attack Strategies

1) Optimal Offline Full Attacks: In the case of full attacks ($\beta = 1$), the total-energy demand model allowed the attacker to collapse the allowance of each job into a single time slot, while in this model, a job j must be served in exactly s_j time slots. However, we can still make use of the results developed earlier as follows. We break each job j into s_j separate sub-jobs, each having the same arrival time, deadline and the power requirement as those of j and each should be served in exactly one time slot. With an entirely forced schedule on the operator, Problem (PmaxminS) is thus turned into a maximization problem as before. In essence, to find the cost-maximizing schedule of those new (smaller) jobs, we are still attempting to form a clique partition of the resulting set of jobs only with the additional constraint that no two subjobs resulting from a job j can be scheduled in the same clique.

Let $\tilde{J} = \{(1,1), \ldots, (1, s_1), \ldots, (n, 1), \ldots, (n, s_n)\}$ be the extended set of job indices, where (j, k) denotes the kth subjob of the original job $j \in J$. Our clique partition is now over $\tilde{J}$. For any clique K, let $J_K = \{j \in J \colon (j,k) \in K, \text{ for some } k\}$, i.e., the set of jobs that originated the subjobs in K. For any time-slot t, we define a locally maximal clique, K^t, in this new setting as the set of subjobs that intersect at t, where at most one subjob from any job can be included. Following this definition, it is clear that the optimal solution indeed contains a locally maximal clique of subjobs, and, this also holds for any set of subjobs entirely contained within an interval.

Let $\overline{C}(k, l, \{m_j\}_{j \in J})$ denote the maximum achievable cost by solely scheduling $m_j \leq s_j$ subjobs of job j within interval $[k, l]$, which is defined to be 0 if for some j, $m_j > l - k + 1$, or $m_j > 0$ and $[k, l] \subsetneq [a_j, d_j]$. Our objective is to find $\overline{C}(1, T, \{s_j\}_{j \in J})$. Similar to Algorithm 2, we can construct a recursion that computes $\overline{C}(k, l, \{m_j\}_{j \in J})$ by parsing for locally maximal cliques in each time-slot $z \in [k, l]$. However, we observe that, unlike our previous model, a locally maximal clique in our extended set of jobs *does not* divide a problem instance into a unique pair of smaller problems. Instead, all the potential subproblem-pairs resulting from a given locally maximal clique should be considered. We then have:

$$\overline{C}(k, l, \{m_j\}_{j \in J}) = \max_{z \in [k,l], m'_j \in [0, m_j - 1] \forall j} \Bigg[C\Bigg(\sum_{j \in K^z_{k,l}} p_j \Bigg) + \overline{C}(k, z-1, \{m'_j\}_{j \in J}) + \overline{C}(z+1, l, \{m_j - 1 - m'_j\}_{j \in J}) \Bigg]. \tag{15}$$

We note that the complexity of this algorithm grows exponentially with the maximum clique size for a given problem instance, which indicates that the strategy can be computationally expensive for the attacker to use in practice. Due to the high complexity of the proposed attack, we have considered a relatively small scale setting in our simulations on offline attacks (see Figure 3). An interesting open problem is to design a more efficient attack strategy that is close to optimal or rigorously prove that such an attack is hard to find.

2) Online Full Attacks: In the online case, we consider an attack similar to Algorithm 3. The attacker again maintains a set of active jobs in A. In any time-slot t, the attacker checks if there is a job j such that $d_j = t + s_j - 1$. Note that to satisfy its service time requirement, such a job j cannot be further delayed. If this is the case, all the jobs in A are modified so that they will be scheduled for a consecutive number of time slots starting from t until their service time requirements are satisfied. These jobs are then forwarded to the operator, and A is the set to the empty set. It is important to notice that, similar to Algorithm 3, if we only consider the set of jobs in A, then this strategy enforces the highest possible cost for those jobs.

Algorithm 6 Online Full Attack (constant-power model)

```
$A \leftarrow \emptyset$. In any time-slot $t$,
$A \leftarrow A \cup \{j : a_j = t\}$;
if $d_j = t + s_j - 1$ for some job $j \in A$ then
    For each job $k$ in $A$, $a'_k \leftarrow t, d'_k \leftarrow t + s_k - 1$;
    Forward the set of (forged) jobs in $A$ to the operator;
    $A \leftarrow \emptyset$
```

Since each demand is processed once, this algorithms has a complexity of $O(n)$. Similar to Lemma 4, we have the following observation for the constant-power model.

Lemma 5: Any clique X in an optimal (offline) solution that achieves C_{max} is a disjoint union of $X \cap K_i$, where $X \cap K_i$ is non-empty for at most r_2 different K_i, where $r_2 = (s_{max} - s_{min} + 1)\left(\left\lceil \frac{\max_j(l_j - s_j + 1)}{\min_j(l_j - s_j + 1)} \right\rceil + 1\right)$.

It is then straightforward to extend the proof of Theorem 2 to show that the above algorithm achieves at least a fraction $\frac{1}{r_2^{b-1}}$ of the optimal offline cost in the constant-power model.

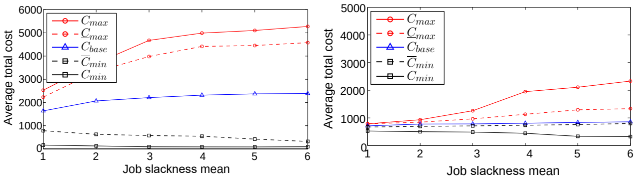
C. Limited Attacks

1) Offline Limited Attacks: To derive an offline limited attack in the constant-power model, we consider an algorithm similar to Algorithm 4. The optimal offline algorithm discussed in the previous section is first applied to find the optimal clique partitioning of sub-jobs when there is no budget constraint. Greedy algorithms are then applied twice; once to choose a set of cliques to fully compress, and to choose a set of sub-jobs within the first unchosen clique on the list, and the choice that results in a higher cost is adopted. In both cases, we require the total number of sub-jobs chosen to be bounded by βn. This ensures that the total number of modified jobs is also bounded by βn. Since finding the optimal clique partitioning may take exponential time in the worst case, this algorithm also has an exponential time complexity. Let $s_{avg} = (\sum_j s_j)/n$ denote the average service time requirement. Assume $C(E) = E^b, b \in \mathbb{R}, b \geq 1$. Since there are $\sum_j s_j$ sub-jobs in total and βn of them are compressed, following Theorem 4, the guaranteed performance of this attack readily becomes $C^1_{maxmin}(\beta) \geq \frac{1}{2}\left(\frac{\beta}{s_{avg}}\right)^b C_{max}$.

2) Online Limited Attacks: We then modify Algorithm 6 to obtain an online limited attack as we did for the total-energy model. The attacker maintains the set of active jobs in A, and samples a fraction β of them to be modified, saved in A'. At any time t, if $d_j = t + s_j - 1$ for some job j in A, all the job in A' are modified as in Algorithm 6. The algorithm also checks the two boundary conditions as we explained before to ensure that all the budget is used and no more. Assume $C(E) = E^b, b \in \mathbb{R}, b \geq 1$. Similar to Algorithm 5, this algorithm also has a complexity of $O(n)$. As in the total-energy model, although there is no worst-case guarantee, we expect that this simple attack obtains an expected cost that is at least a constant fraction of $\underline{C}_{maxmin}(\beta) \geq \left(\beta \frac{p_{min}}{p_{max}}\right)^b \underline{C}_{max}$ for $i.i.d.$ demands and when s_j is a constant for all j.

VI. Numerical Results

In this section, we provide numerical results that illustrate the impact of stealthy attacks under various settings. More results can be found in our online technical report [13].In this section, unless stated otherwise, the job arrivals are simulated as a Poisson arrival process with mean 3. We use a quadratic cost function $C(E) = E^2$ in all of our simulations.



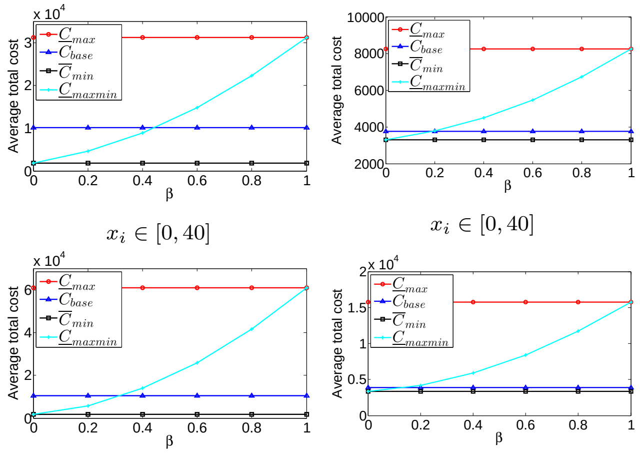
(a) total-energy model (b) constant-power model

Fig. 3: Comparison between the performance of a fully-compromised smart grid (offline and online attacks), the current grid, and an un-compromised smart grid (offline and online scheduling), under varying job allowance means.

Full Attacks: In Figure 3, we compare the performance of a non-compromised smart grid, a fully-compromised smart grid and the "dumb" grid (where all jobs are immediately scheduled upon their arrival), for both the total-energy model and the constant-power model, for a total of 20 jobs. All the job slackness are $i.i.d.$ exponential random variables, as well as the service time intervals. In the constant-power model, the job slackness mean is varied between 1 and 6, and the service time mean is fixed to 2. The power requirement per time slot, for each job, is uniformly distributed in the interval $[1, 5]$. For comparison purpose, for each job generated in the constant-power model, a job with the same arrival, slackness, and total power requirement is generated for the total-energy model. The plots report the average performance of both systems over 10 trials. For the total-energy model, $C_{min}, \overline{C}_{min}, C_{base}, \underline{C}_{max}$, and C_{max} correspond to the cost achieved by Algorithm 1, the AVR algorithm, the baseline cost (5), Algorithm 3, and Algorithm 2, respectively. For the constant-power model, they correspond to the lower bound obtained from the continuous relaxation of the minimization problem for the operator, the cost obtained by the Controlled Release (CR) policy [12], the baseline cost (6), the cost obtained by Algorithm 6, and that by the optimal offline full attacks discussed in Section V-B1, respectively.

We observe that, as the job slackness mean increases, for both models, further scheduling opportunities are offered to the legitimate operator, and hence further savings in the total cost are attained if the smart grid is not compromised. In the presence of an attacker, however, a similar flexibility is available to the attacker, and accordingly the severity of the attack increases as the job slackness mean increases. We also observe that the uncompromised total-energy system outperforms the constant-power model, in terms of total cost, due to the increased job scheduling flexibility in the former. For the same reason, attacks are more harmful for this model as well. In the total-energy model, when compared to the costs paid by the regular grid, an offline (online) attack causes an increase in cost by 154% (136%) with a job slackness mean of 1 and up to 220% (191%), while the expected cost to be paid for an uncompromised system should, in fact, decrease by values ranging in $200\% - 2500\%$. A similar comparison could be drawn in the constant-power model. Therefore, overall, the unprotected smart grid simulated here, not only does it fail to meet the cost savings prospected in a smart grid, it performs far worse than the current electric grid.

Online Limited Attacks: We now investigate the performance of online limited attacks and compare them with online full attacks. We assume that the operator schedules the set of (partially) modified demands using the AVR algorithm for the total-energy model, and the CR algorithm for the constant-power model. Since online attacks have lower complexity than their offline counterparts, we consider a larger setting with 100 jobs and each simulation is repeated 100 times. We consider the same power requirement, service time, and inter-



$x_i \in [0, 40]$

$x_i \in [0, 40]$

$x_i \in [0, 10](10\%), [40, 50](90\%)$

(a) total-energy model

$x_i \in [0, 10](10\%), [40, 50](90\%)$

(b) constant-power model

Fig. 4: Performance of a partially-compromised smart grid under online limited attacks with various values of β.

arrival time distributions as before. Theorem 2 and Theorem 3 together indicate that a higher cost can be expected if most jobs have large job slackness. To confirm this, we consider two job slackness distributions, (1) a uniform distribution between [0,40], and (2) a mixture of two types of demands, where 90% of demands have high elasticity with their slackness uniformly distributed in [40,50], and 10% of demands are more emergent with their slackness uniformly distributed in [0,10]. The attacks were conducted with β values ranging between 0 and 1. Figure 4 reports our results for these attacks, where the values for the corresponding online full attacks and the baselines are also plotted for reference. We observe that large job slackness can indeed enforce higher cost. For both models, even with a low fraction of jobs to be modified, the attacker still causes significant harm, compared to the uncompromised system. Moreover, the attacker becomes capable of driving the system to perform worse than its nominal point (the regular grid) with β as low as 0.4 and 0.2 for the total-energy model and the constant-power model, respectively.

VII. Observations and Suggestions

From our analytical studies and simulation results, we make several observations and suggestions to the operator for thwarting the new type of attacks that we consider in the paper.

Information Hiding: We observe that the attacker's capability is significantly constrained by the amount of information it has regarding the operator and the demand patterns. In particular, to derive the best β, the attacker needs to know the intrusion detection algorithm and the key parameters such as the significance level used by the operator. Moreover, the attacker requires some prior information about the demands to make best use of its budget, such as the number of demands and the ranges of their values. Therefore, one efficient approach to reduce the damage is to properly hide these information from the attacker, e.g., by introducing noise into the data and algorithms.

Intrusion Detection: We suggest to develop robust intrusion detection schemes that can strike a balance between the potential loss from attacks and the cost of detection. In particular, we suggest to develop a better statistical modeling of time-elastic demands, and study advanced stream data mining algorithms that can deal with the high dimension of the demand data set. Moreover, as we discussed above, it is useful to develop intrusion detection algorithms that can make it hard for the attacker to derive efficient parameters to use.

Load Management: We note that the scheduling algorithm used by the operator has a big impact on the total energy cost, especially when the attacker can only compromise a small number of demands. We have provided efficient solutions for the operator in the total-energy model, but better solutions are needed for the constant-power model and more general demand models. For instance, Figure 4 indicates that online limited attacks are more efficient in the constant-power model. We believe that this is due in part to the poor performance of the CR algorithm in our setting. Moreover, it is important to develop *robust* algorithms that can provide a guaranteed performance even when part of demands have been modified by adversaries.

Robust and Adaptive Defense: We suggest to develop robust defense algorithms to identify the set of most critical channels (or smart meters) to protect. From our analysis, it is clear that those demands (or a set of overlapping demands) with highest power requirement and maximum time elasticity are most beneficial to the attacker, due to the large gap between C_{max} and C_{min} if we consider these demands only. When these demands are mostly generated by a given subset of customers, the corresponding links can be protected to efficiently reduce damage. In the face of more advanced attackers, however, a fixed defense strategy is insufficient, as the attacker can always identify the weakest link in the system. Therefore, it is important to study adaptive defense strategies in the face of strategic attackers.

VIII. Conclusion

In this paper, we have studied the performance of the smart grid, in terms of energy efficiency, in the presence of an active attacks on the system. In the presence of a limited intrusion detection mechanism at the grid operator, we have proposed optimal scheduling and undetectable attack strategies. We have derived lower and upper bounds on the maximum achievable cost by an attacker with low complexity, online algorithms. Overall, our theoretical analysis and numerical results show that the time-elasticity of electric load, when exploited by malicious attacks, could result in costs significantly higher than those expected for both the smart grid and the current electric grid, motivating the need for stronger intrusion detection and defense strategies for grid operators.

References

[1] Y. Abdallah, Z. Zheng, N. B. Shroff, and H. E. Gamal, "On the efficiency-vs-security tradeoff in the smart grid," in *Proc. of IEEE CDC*, 2012.

[2] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 57–64, 2010.

[3] T. Lui, W. Stirling, and H. Marcy, "Get smart," *IEEE Power and Energy Magazine*, vol. 8, no. 3, pp. 66–78, 2010.

[4] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security & Privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[5] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. of ACM CCS*, 2009.

[6] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[7] L. Jia, R. J. Thomas, and L. Tong, "Impacts of malicious data on real-time price of electricity market operations," in *Proc. of Hawaii International Conference on System Science*, 2012.

[8] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011.

[9] D. Grochocki, J. H. Huh, R. Berthier, R. Bobba, W. H. Sanders, A. A. Cárdenas, and J. G. Jetcheva, "Ami threats, intrusion detection requirements and deployment recommendations," in *IEEE SmartGridComm*, 2012.

[10] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *IEEE SmartGridComm*, 2010.

[11] F. M. Cleveland, "Cyber security issues for advanced metering infrastructure," in *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008.

[12] I. Koutsopoulos and L. Tassiulas, "Optimal control policies for power demand scheduling in the smart grid," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1049–1060, 2012.

[13] Y. Abdallah, Z. Zheng, N. B. Shroff, and H. E. Gamal, "The Impact of Stealthy Attacks on Smart Grid Performance: Tradeoffs and Implications," Technical Report, available online at https://arxiv.org/abs/1502.06004, 2014.

[14] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, pp. 1344–1371, 2013.

[15] J. M. Hendrickx, K. H. Johansson, R. M. Jungers, H. Sandberg, and K. C. Sou, "Efficient computations of a security index for false data attacks in power networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3194–3208, 2014.

[16] D. Deka, R. Baldick, and S. Vishwanath, "Optimal hidden scada attacks on power grid: A graph theoretic approach," in *Proc. of ICNC*, 2014.

[17] L. Xie, Y. Mo, and B. Sinopoli, "False data injection attacks in electricity markets," in *Proc. of SmartGridComm*, 2010.

[18] Y. Zhao, A. Goldsmith, and H. V. Poor, "Fundamental limits of cyber-physical security in smart power grids," in *Proc. of IEEE CDC*, 2013.

[19] R. Berthier and W. H. Sanders, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *IEEE PRDC*, 2011.

[20] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, "Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study," *IEEE Systems Journal*, vol. 9, no. 1, pp. 31–44, 2015.

[21] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced cpu energy," in *Proc. of IEEE FOCS*, 1995, pp. 374–382.

[22] D. Gijswijt, V. Jost, and M. Queyranne, "Clique partitioning of interval graphs with submodular costs on the cliques," *RAIRO-Operations Research*, vol. 41, no. 03, pp. 275–287, 2007.

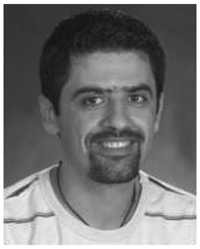
[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2009.

[24] B. Hajek, "Performance of global load balancing by local adjustment," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1398–1414, 1990.

Yara Abdallah (S'08) received the B.S. degree in electrical and computer engineering from Mansoura University, Mansoura, Egypt, in 2008, and the M.S. degree in wireless technology from the Wireless Intelligent Networks Center (WINC), Nile University, Egypt, in 2010. She is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH. Her research interests are wireless physical layer security and IEEE 802.11 security.

Zizhan Zheng (S'07-M'10) received his Ph.D. degree in Computer Science and Engineering from The Ohio State University in 2010, and the M.S. degree in Computer Science from Peking University, China, in 2005. He was a postdoctoral researcher in the ECE department at The Ohio State University from 2010-2014. He is currently an associate specialist in the CS department at University of California, Davis. His research is in the areas of network optimization and cybersecurity.

Ness B. Shroff (S'91-M'93-SM'01-F'07) received his Ph.D. degree in Electrical Engineering from Columbia University in 1994. He joined Purdue university immediately thereafter as an Assistant Professor in the school of ECE. At Purdue, he became Full Professor of ECE in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined The Ohio State University, where he holds the Ohio Eminent Scholar endowed chair in Networking and Communications, in the departments of ECE and CSE. He holds or has held visiting chaired professor positions at Tsinghua University, Beijing, China and Shanghai Jiaotong University, Shanghai, China, and a visiting position at the Indian Institute of Technology, Bombay, India. Dr. Shroff is currently an editor at large of IEEE/ACM Trans. on Networking, and senior editor of IEEE Transactions on Control of Networked Systems. He has received numerous best paper awards for his research and listed Thomson Reuters Book on The Worlds Most Influential Scientific Minds as well as noted as a highly cited researcher by Thomson Reuters (previously ISI). He also received the IEEE INFOCOM achievement award for seminal contributions to scheduling and resource allocation in wireless networks.



Hesham El Gamal (M'99-SM'03-F'10) received the B.S. and M.S. degrees in electrical engineering from Cairo University, Cairo, Egypt, in 1993 and 1996, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Maryland at College Park, in 1999. From 1993 to 1996, he served as a Project Manager with the Middle East Regional Office of Alcatel Telecom. From 1996 to 1999, he was a Research Assistant with the Department of Electrical and Computer Engineering, University of Maryland at College Park. From February 1999 to December 2000, he was with the Advanced Development Group, Hughes Network Systems (HNS), Germantown, MD, as a Senior Member of Technical Staff. Since January 2001, he has been with the Electrical and Computer Engineering Department, Ohio State University, where he is now a Professor. He held visiting appointments at UCLA, Institut Eurecom, and served as a Founding Director for the Wireless Intelligent Networks Center (WINC) at Nile University (2007-2009). He holds 12 patents. Dr. El Gamal is a recipient of the HNS Annual Achievement Award (2000), the OSU College of Engineering Lumley Research Award (2003, 2008), the OSU Electrical Engineering Department FARMER Young Faculty Development Fund (2003-2008), the OSU Stanley E. Harrison Award (2008), and the National Science Foundation CAREER Award (2004). He has served as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS (2001-2005), an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING (2003-2007), a Guest Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY Special Issue on Cooperative Communications (2007), a member of the SP4COM Technical Committee (2002-2005), a Co-Chair of the Globecom'08 Communication Theory Symposium, and a Co-Chair of the 2010 IEEE Information Theory Workshop. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and a Guest Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY.

Tarek M. El-Fouly (M'06-SM'13) received his DEA and PhD from the University of Franche Comte in France, in 1996 and 2000 respectively. He has worked as an assistant professor at the university of Ain Shams Cairo Egypt before joining Qatar University He is currently an assistant professor in the college of engineering at Qatar University. He has over 10 years of experience in computer network research. Dr. Elfouly published over 50 papers, more than half of them are related to wireless sensing and network security. Dr. Elfouly has many projects under development related to assistive technologies for people with disabilities. His projects won many national and regional awards. His research interests include network security and protocols, physical layer security and wireless sensor networks especially in the field of structural health monitoring and health applications.