GTHI: A Heuristic Algorithm to Detect Malicious Users in Smart Grids

Xiaofang Xia, Yang Xiao, *Senior Member, IEEE*, Wei Liang, *Senior Member, IEEE*, and Meng Zheng

***Abstract*—With many countries trying to establish their own smart grids, smart meters are massively deployed throughout the world. Although smart meters are manufactured with low tamper-resistant components, malicious users with just a moderate level of computer knowledge are able to launch cyber attacks. By manipulating electricity consumption readings to smaller values, malicious users can steal electricity from utility companies. To reduce the losses incurred by electricity theft, utility companies must provide preventative and detective methods to identify fraudulent behaviors. Our goal is to identify all malicious users in a neighborhood area network within the shortest detection time. To achieve this goal, we propose Group Testing based Heuristic Inspection (GTHI) algorithm, which can estimate the ratio of malicious users on-line, mainly by collecting the information that how many malicious users have been identified during the inspection process. Based upon the ratio of malicious users, the GTHI algorithm adaptively adjusts inspection strategies between an individual inspection strategy and a group testing strategy. This helps shorten the detection time. Furthermore, when applying the group testing strategy, the GTHI algorithm also determines the group size of users to be probed in line with the estimated malicious user ratio. Experiment results show that compared to existing methods, the GTHI algorithm has advantages of conducting fewer inspection steps or being more practical.**

***Index Terms*—Electricity theft, smart grid, smart meters, cyber attacks, security.**

I. Introduction

NOWADAYS, numerous countries are trying to establish their own smart grids. To achieve this goal, they must first massively deploy smart meters which are intelligent digital devices with the capability of two-way communication [1]. Smart meters can periodically transmit power consumption readings to utility companies from remote locations. This significantly reduces the human involvement in the billing process. Unfortunately, while bringing convenience and efficiency, smart meters also pose a great threat of electricity theft, which is usually committed via cyber attacks.

Smart meters at the end-user level are typically manufactured with low tamper-resistant components. By launching cyber attacks such as injecting computer virus to operating systems of smart meters, malicious users[1] can manipulate meters' readings to smaller values. It is reported that individuals with just a moderate level of computer knowledge are able to compromise smart meters [2]. Besides, physical attacks, which have been already used to steal electricity for a long time in traditional power systems, can also be employed to tamper with smart meters in smart grids. The most common physical attacks include, but are not limited to, bypassing energy meters and directly hooking from power lines. Thus, utility companies face even more serious electricity theft in smart grids than in traditional power systems.

The current annual losses caused by electricity theft are $89.3 billion around the world [3]. Undoubtedly, utility companies are the first ones suffering from electricity theft. It is reported that North American utility companies lose about $6 billion annually due to electricity theft [4]. All the money lost gets passed along to other customers in the form of higher electricity prices. Apart from economical losses, electricity theft also lowers the quality of power supplied to users, resulting in easier malfunctioning of appliances. More importantly, electricity theft prevents utility companies from having an accurate view of users' actual demand of electricity, and this results in that the utility companies do not generate enough amount of electricity that all users really need [5]. The shortage of electricity further leads to the unreliability of power systems [6]. For the countries where electricity theft is pervasive, such as India and Brazil, power outages become a way of life and put a drag on the development of economy [7].

To identify fraudulent behaviors related to electricity theft, researchers have proposed a lot of preventative and detective methods, among which classification-based methods and power measurement-based methods garner much attention. The classification-based methods detect users' electricity theft

Manuscript received February 27, 2018; revised June 23, 2018; accepted July 9, 2018. Date of publication July 11, 2018; date of current version June 4, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0101300; in part by the US National Science Foundation (NSF) under Grant CNS-1059265; in part by the National Natural Science Foundation of China under Grants 61374200, 61673371, and 71661147005; and in part by the Youth Innovation Promotion Association Chinese Academy of Sciences under Grant 2015157. Recommended for acceptance by S. Chellappan. *(Corresponding authors: Yang Xiao; Wei Liang.)*

Xiaofang Xia is with the Key Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290 USA (e-mail: xiaofangxia89@gmail.com).

Yang Xiao is with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290 USA(e-mail: yangxiao@ieee.org).

Wei Liang and Meng Zheng are with the Key Lab of Networked Control Systems and Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: weiliang@sia.cn; zhengmeng_6@sia.cn).

Digital Object Identifier 10.1109/TNSE.2018.2855139

1. Malicious users are users stealing electricity.

related behaviors by analyzing fine-grained electricity consumption readings. Their performances greatly depend on both normal and abnormal samples used in the training phase. However, in the real world, the abnormal samples are usually not easy to be obtained. The power measurement-based methods leverage redundant devices, such as sensors [8] and observer meters [9], to monitor users' electricity consumptions. They usually identify malicious users accurately. Nonetheless, such approaches often have exorbitant costs. For instance, the mutual inspection strategy [10] requires the installation of one extra inspector[2] for each user. To reduce the cost, the authors in paper [11] propose to install several inspectors for each neighborhood area network (NAN) in a smart grid.

Following the work in [11], in this paper, we aim to locate malicious users in an NAN within the shortest detection time using a limited number of inspectors. We propose the Group Testing based Heuristic Inspection (GTHI) algorithm. By collecting the information of the number of both malicious users and honest users that have already been identified during the inspection process, the GTHI algorithm estimates the ratio of malicious users in the NAN on-line. In line with the malicious user ratio, the GTHI algorithm adaptively adjusts inspection strategies between an individual inspection strategy and a group testing strategy during the inspection course. Specifically, if the malicious user ratio is higher than a specific threshold, the GTHI algorithm employs the individual inspection strategy whereby users are probed one by one. Otherwise, the GTHI algorithm applies the group testing strategy by which a group of users are probed as a whole in an inspection step. When the group testing strategy is applied, the estimated ratio also determines the group size of users to be probed. On the whole, a lower malicious user ratio implies a larger group size. If there are indeed malicious users among the group of users being probed, a few more inspection steps are consecutively conducted on these users until a malicious user is located. The contributions of this paper are highlighted as follows: (1) We propose the GTHI algorithm to locate malicious users, which can be used in both static cases where new malicious users do not appear and dynamic cases where new malicious users do appear during the inspection process; (2) The GTHI algorithm helps shorten the detection time, mainly by adaptively adjusting inspection strategies during the inspection course; (3) We provide performance analyses on the GTHI algorithm, mainly including the minimum upper bound of the number of inspection steps (i.e., the detection time) as well as the selection of the threshold for the estimated ratio of malicious users; (4) A series of experiments are conducted to evaluate the GTHI algorithm;

We organize the rest of this paper as follows. In Section II, we review related works on electricity theft detection. In Section III, we present the problem statement. In Section IV, we propose the GTHI algorithm. Experiment results are reported in Section VI. We conclude this paper in Section VII.

2. An inspector is in nature a function enhanced smart meter.

II. Related Works

Smart Grid is one special kind of cyber-physical systems in which security becomes more complex since they normally involve both cyber and physical aspects [12], [13]. Among all electricity theft detection methods, the classification-based methods [14]–[18] are the most popular. The data leveraged by the classification-based methods are naturally generated during users' daily life, and utility companies do not have to pour out a lot of money. However, the low investment implies poor performance. It is argued that the classification-based methods usually have a low detection rate but a high false positive rate [19]. For example, a support vector machine algorithm is employed to automatically extract users' consumption patterns from historical kWh consumption data, but the detection rate reaches just 70 percent [15]. Realizing that it is the lack of thorough dataset of attack samples which limits the detection rate, the authors in paper [18] propose to generate a synthetic attack dataset, benefiting from the fact that to a large extent, malicious users' consumption patterns are predictable. This increases the detection rate to 94 percent. However, malicious users are always tricky enough to apply new consumption patterns to escape the detection, and we obviously cannot predict all consumption patterns. Moreover, users' consumption patterns can also be impacted by many non-malicious factors, which include, but are not limited to, the change of residents, the change of appliances, and the change of seasonality [18], [20]. The non-malicious factors are important sources of the high false positive rate.

As the classification-based methods have so many shortcomings, the power-measurement-based methods [4]–[11], [21]–[29] are attracting more and more attentions. As aforementioned, these methods require the installation of redundant devices. By comparing readings measured by these devices and readings reported by users, malicious users can be accurately located. For example, the authors in [4], [9], [10], [21]–[25] propose to install a central observer meter to register power distributed to a group of users. The electricity theft detector proposed in paper [4] models adversaries' behaviors based upon the Lagrange polynomial interpolation. The papers [11], [26]–[28], focus on minimizing the detection time of malicious users, using a limited number of inspectors. If there is just one malicious user, Binary-Coded Grouping-based Inspection (BCGI) algorithm [28] can be utilized and the basic idea is to group users based upon the binary sequences of their identification numbers so that the unique malicious user can be identified by just one inspection step.

In World War II, the group testing problem was first proposed for accelerating and economizing the procedure of weeding out individuals infected with syphilitic [30]. Nowadays, group testing has a much wider range of practical applications, especially in industrial sectors. Some typical applications include removing leakers from a set of devices and finding the broken bulb on a Christmas tree [31]. Their common goal is to find out the defective elements with as few tests as possible.

The authors of papers [11]–[27] propose a series of binary tree based inspection approaches, among which Adaptive

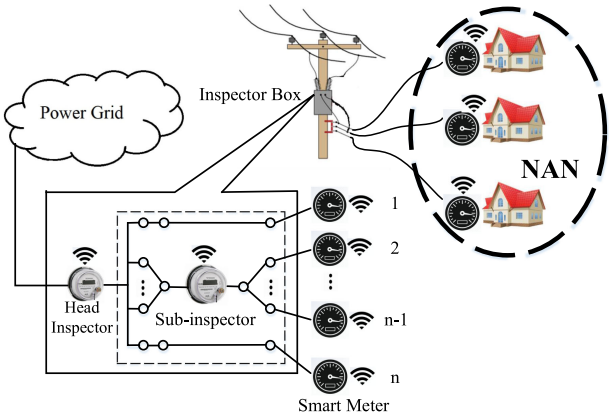


Fig. 1. A simplified smart metering system for an NAN.

Tree Inspection (ATI) algorithm [11] is the best. The ATI algorithm adopts a virtual tree approach whereas the GTHI algorithm adopts group testing. By estimating the ratio as well as the arrangement of malicious users, the ATI algorithm allows inspectors to skip some internal nodes on the binary tree and directly probe nodes at lower levels. Starting from the root of the tree, the ATI algorithm does a traversal-and-probe with a depth-first-search algorithm. Note that when the inspection process starts, the ATI algorithm probes all users. Before obtaining useful information, the inspectors have to conduct a lot of inspection steps. By contrast, when the inspection process starts, the GTHI algorithm probes just a small number of users. This allows to estimate the malicious user ratio after just a few inspection steps. Correspondingly, the inspection strategies can then be adjusted in time, which helps further shorten the inspection time to a large extent.

III. Problem Statement

A simplified smart metering system for an NAN is shown in Fig. 1. As we can see, at each user's premises, there is a smart meter which has the capability of two-way communication [32]. The smart meter records the corresponding user's electricity consumptions and then periodically reports the data to utility companies for the billing purpose. Assume that there are n users in the NAN. We denote these users by $U = \{1, 2, \ldots, n\}$. Most users report their electricity consumption readings honestly. These users are referred to as "honest users". However, there are also some unscrupulous users who manipulate their readings to smaller values, hoping to be billed for less or even free. As aforementioned, these users are referred to as "malicious users".

For the purpose of detecting malicious users, we install an "inspector box" [11] at an electrical pole in each NAN. In the inspector box, there are two categories of inspectors: a head inspector and several sub-inspectors. The head inspector is responsible for detecting the existence of malicious users and always monitors all users. As for the sub-inspectors, their responsibility is to locate all malicious users exactly. Notably, we in this paper assume that the set of users monitored by the sub-inspectors can be changed automatically or manually.

Algorithm 1: Probe

Require: G
Ensure: inspection result
1: **if** $r - \sum_{j \in G} q_j \geq \tilde{\delta} + \varepsilon$ **then**
2: There are malicious users in G;
3: **Return** inspection result "dirty";
4: **else**
5: All the users in G are honest;
6: **Return** inspection result "clean";
7: **end if**

Let G be a subset of users being monitored by an inspector. For the head inspector, we have $G = U$; and for sub-inspectors, we have $G \subset U$. In a reporting period, the inspector works as follows: (1) Measuring the total amount of electricity distributed to the users in G; (2) Receiving the reported electricity consumptions of the users in G. Let r denote the reading of one inspector. Since technical losses are inevitable during the electric power transmission, let δ denote the total amount of technical losses of the users in G. Let q_j denote the actual amount of electricity consumed by user $j, j \in U$. Then, we have $r = \delta + \sum_{j \in G} q_j$. Let q'_j denote the reported electricity consumption of user $j, j \in U$. Apparently, for honest users, we have $q_j = q'_j$; and for malicious users, we have $q_j > q'_j$. Thus, if there exist malicious users in the user set G, we can derive $r - \sum_{j \in G} q'_j = \delta + \sum_{j \in G}(q_j - q'_j) > \delta$.

In applications, it is very difficult to obtain the accurate value of technical losses. Usually, we estimate it with some existing mathematical models, e.g., [33]. Let $\tilde{\delta}$ denote the estimated value of δ. Since there usually exists a deviation between the real value δ and the estimated value $\tilde{\delta}$, we introduce a threshold, denoted by ε, to help judge whether there are malicious users in G. Specifically, we have the Algorithm 1, called the probing algorithm. The head inspector conducts probing operations at all reporting periods. When the head inspector gets an inspection result "dirty" at a certain period, it detects the existence of reading anomalies. The head inspector does not know which users are stealing electricity. For the purpose of locating malicious users exactly, the sub-inspectors start to constantly perform probing operations on different set of users. Note that in this paper, when a sub-inspector conducts the probing operation for one time, we say that the sub-inspector performs one inspection (step).

In this paper, we investigate the *Malicious Meter Inspection* (MMI) problem [11] which aims to locate all malicious users within the shortest detection time. Since each inspection lasts for one reporting period, which is usually set as 15 minutes in the smart grid, this goal can be abstracted as minimizing the number of inspection steps conducted by the sub-inspectors.

In the following context, the malicious users who commit electricity theft from the period when the head inspector first detects the existence of reading anomalies are referred to as "original malicious users". To locate the original malicious users, the sub-inspectors have to conduct a series of inspections, and this undoubtedly takes some time. During this period, some users who are previously honest are likely to start compromising

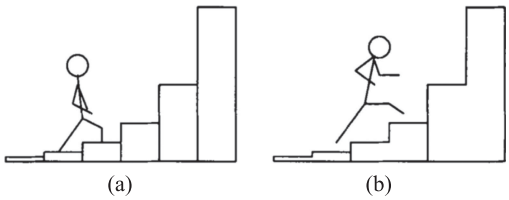


Fig. 2. (a) The doubling process; (b) the jumping process [35].

their smart meters. We refer to these users as "new malicious users". Based upon whether new malicious users appear during the inspection process, we can roughly divide the inspection cases in the *MMI* problem into the following two categories: (1) static cases where new malicious users do not appear during the inspection process; (2) dynamic cases where at least one new malicious user appears during the inspection process. The distinction of state cases and dynamic cases is for research purpose. In practice, we can use dynamic cases.

We assume that once a user is identified as being malicious, this user is immediately disconnected from the service of power as indicated in [34].

IV. The GTHI Algorihtm

We explain the proposed Group Testing based Heuristic Inspection (GTHI) algorithm next. We define a round of inspection as the inspection process where all the users whose statuses are first unclear are identified as either being malicious or being honest. When the head inspector detects the existence of reading anomalies, the sub-inspectors start the first round of inspection. Assume that after the statuses of all users are identified, the head inspector can still detect reading anomalies. In this case, the sub-inspectors then immediately start a new round of inspection to locate new malicious users. Normally, in static cases, the sub-inspectors conduct just one round of inspection to locate malicious users; in dynamic cases, they usually conduct multiple rounds of inspection.

In this paper, we let $\mathbf{N}$ denote the set of natural numbers and let $\mathbf{N}^+$ denote the set of positive integer numbers. Assume that sub-inspectors conduct a total number of $a, a \in \mathbf{N}^+$ rounds of inspections to locate all malicious users, including the original malicious users as well as the new malicious users. Let $W_i, i \in \{1, 2, \ldots, a\}$ denote the set of users that need to be probed at the ith round of inspection. Then, W_i contains the users whose status remains unclear at the beginning of the ith round of inspection. Apparently, at the first round of inspection, all the users should be probed. Let M_i denote the set of users that are identified as malicious at the ith round of inspection. Since the users in M_i are immediately disconnected from the service of power when identified as being malicious, these users should not be re-probed in the subsequent rounds of inspection. We have

$$W_i = \begin{cases} U, & \text{if } i = 1 \\ U \setminus \bigcup_{j=1}^{i-1} M_j, & \text{if } i = 2, 3, \ldots, a, \end{cases} \quad (1)$$

where U is the set of all users and "\" denotes the difference of two sets.

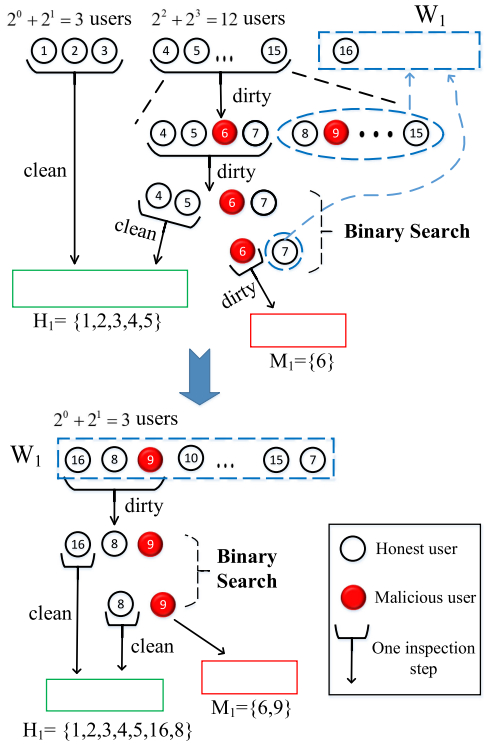


Fig. 3. An example of the GTHI algorithm under a static case.

The ith round of inspection ends when all the users in W_i are identified as either being malicious or being honest. Let H_i denote the set of users that are identified as being honest at the ith round of inspection. When the ith round of inspection starts, we initiate $M_i = \emptyset$ and $H_i = \emptyset$, with $\emptyset$ being the empty set. When the ith round of inspection ends, we have $W_i = \emptyset$ and $U = H_i \cup \bigcup_{j=1}^{i} M_j$.

At the beginning of each round of inspection, we apply a group testing method, which is called the jumping strategy [36], to guide the inspection. The jumping strategy is actually an improvement over the doubling strategy [37]. For the purposes of comparison and better understanding, we illustrate the basic idea of both the doubling strategy and the jumping strategy in Fig. 2(a) and 2(b), respectively. Simply speaking, by the doubling strategy, the sub-inspectors probe disjoint sets of sizes $2^0, 2^1, 2^2, \ldots$, until an inspection result "dirty" is obtained. By contrast, according to the jumping strategy, the sub-inspectors probe disjoint user sets of sizes $2^0 + 2^1$, $2^2 + 2^3$, $2^4 + 2^5$, $\ldots$, until an inspection result "dirty" is obtained. Clearly, to identify the same number of honest users, the sub-inspectors save more inspection steps using the jumping strategy than using the doubling strategy. Based upon the jumping strategy, if at a certain round of inspection, all inspection results from the first to the $(j-1)$th inspection are "clean", the number of users to be probed at the jth inspection step is $2^k + 2^{k+1}$, with $k = 2(j-1), j \in \mathbf{N}^+$. For example, in Fig. 3 where static cases are considered, we assume that there are a total number of sixteen users. As shown in the figure, at the first inspection step, the number of users that are probed is $2^0 + 2^1 = 3$. Since we get an inspection result "clean" at the first inspection step, the number of users probed at the second

inspection step is $2^2 + 2^3 = 12$. Assume that the inspection results from the first to the jth inspection step are all "clean". Then, with these j inspection steps, the total number of users that are identified as honest is $(2^0 + 2^1) + (2^2 + 2^3) + \cdots + (2^{2j-2} + 2^{2j-1}) = 4^j - 1$.

During the whole inspection process, once a sub-inspector gets an inspection result "dirty" when probing $2^k + 2^{k+1}$ users, this sub-inspector then probes a subset of 2^k users out of the $2^k + 2^{k+1}$ users. By doing so, the sub-inspector can reduce the user set which contains malicious users to either a set of 2^k users or a set of 2^{k+1} users with 2^k more honest users. Specifically, if the sub-inspector gets an inspection result "dirty" when probing the subset of 2^k users, the remaining 2^{k+1} users are then put back with the users in W_i, waiting for further inspection in the near future. In this case, the sub-inspector then targets subsequent inspection steps on the subset of 2^k users. On the other hand, if the sub-inspector gets an inspection result "clean", then the 2^k users are evidently honest; and hence the sub-inspector then focuses on the remaining 2^{k+1} users. For example, in Fig. 3, the sub-inspector gets an inspection result "dirty" when probing the twelve users $\{4, 5, \ldots, 15\}$ at the second inspection step. As shown, this sub-inspector then probes four users $\{4, 5, 6, 7\}$ at the third inspection step, and correspondingly gets an inspection result "dirty". Thus, the users $\{8, 9, \ldots, 15\}$ are put back with user 16 which still remains in set W_1, waiting for further inspection. At this time, W_1 is updated as $\{16, 8, 9, \ldots, 15\}$. This sub-inspector then performs the subsequent inspection steps on the users $\{4, 5, 6, 7\}$.

After narrowing down the searching area to 2^k or 2^{k+1} users, we apply a binary search method, which is described as follows. For convenience of description, we use X to denote the 2^k or 2^{k+1} users. The sub-inspectors first probe $\lceil \frac{|X|}{2} \rceil$ users from the user set X, which are denoted as set X'. If we obtain an inspection result "dirty", we can conclude that there are malicious users in X'. In this case, the users in $X \setminus X'$ are put back with the users in W_i, and we update $X = X'$. On the other hand, if we get an inspection result "clean", all the users in X' are honest, and we update $X = X \setminus X'$. The above process repeats until just one user is left in set X, which is explicitly malicious. During the above inspection process, several honest users may also be identified. We conclude the above binary search process in lines $24 \sim 33$ in Algorithm 2. Apparently, when applying the above binary search method to locate a malicious user from a user set X containing malicious users, we need to conduct $\lceil \log_2 |X| \rceil$ inspection steps. As can be seen in Fig. 3, two more inspection steps are conducted when the binary search method is applied to locate one malicious user from users $\{4, 5, 6, 7\}$. The specific inspection process is elaborated as follows. At the fourth inspection step, users $\{4, 5\}$ are identified as being honest. At the fifth inspection step, user 6 is identified as being malicious and user 7 is put back into set W_1. At this time, M_1, W_1 is updated as $\{6\}$, $\{16, 8, 9, \ldots, 15, 7\}$, respectively.

After one user is identified as being malicious, the GTHI algorithm estimates the ratio of malicious users. Let y_i denote the ratio of malicious users at the ith round of inspection, which is initiated as $y_i = 0$ at the beginning of the ith round of inspection. In real applications, we do not know the value of y_i in advance. However, with the ith round of inspection going on, more and more users are identified as either being malicious or being honest. Since we randomly inspect users during the whole inspection process, the ratio of malicious users can be roughly estimated as

Algorithm 2: The GTHI Algorithm

Require: $U = \{1, 2, \ldots, n\}$
Ensure: M ▷ the whole set of malicious users
Globals: $M, W_i, M_i, H_i, \tilde{y}_i, i = 1, 2, \ldots, a$;
Initialization: $M \leftarrow \emptyset, W_1 \leftarrow U, W_i \leftarrow \emptyset, i = 2, 3, \ldots, a$;
$M_i \leftarrow \emptyset, H_i \leftarrow \emptyset, \tilde{y}_i \leftarrow 0, i = 1, 2, \ldots, a$ ▷ Initialization

1: $k \leftarrow 0, i \leftarrow 1$;
2: **while** the head inspector detects reading anomalies in W_i **do**
3: **while** the head inspector detects reading anomalies in W_i **do**
4: **if** ($|W_i| < 3$) or ($|W_i| \geq 3$ and $\tilde{y}_i \geq y_0$) **then**
5: pop one user, denoted as user j, out of W_i;
6: Probe({user j}); ▷ individual inspection
7: **if** the inspection result is dirty **then**
8: $M_i \leftarrow M_i \cup$ {user j};
9: **else** ▷ if the inspection result is clean
10: $H_i \leftarrow H_i \cup$ {user j };
11: **end if**
12: update $\tilde{y}_i$; ▷ $\tilde{y}_i \leftarrow \frac{|M_i|}{|M_i|+|H_i|}$
13: **else** ▷ if $|W_i| \geq 3$ and $y_i < y_0$
14: $X \leftarrow$ pop $\min\{2^k + 2^{k+1}, |W_i|\}$ users out of W_i;
15: PROBE(X);
16: **if** X contains malicious users **then**
17: $X' \leftarrow \frac{|X|}{3}$ users from X;
18: PROBE(X'); ▷ group testing
19: **if** X' contains malicious users **then**
20: $W_i \leftarrow W_i \cup (X \setminus X'), X \leftarrow X'$;
21: **else**
22: $H_i \leftarrow H_i \cup X', X \leftarrow X \setminus X'$;
23: **end if**
24: **while** $|X| > 1$ **do** ▷ binary search
25: $X' \leftarrow \lceil \frac{|X|}{2} \rceil$ users from X;
26: PROBE(X');
27: **if** X' contains malicious users **then**
28: $W_i \leftarrow W_i \cup (X \setminus X'), X \leftarrow X'$;
29: **else**
30: $H_i \leftarrow H_i \cup X'; X \leftarrow X \setminus X'$;
31: **end if**
32: **end while**
33: $M_i \leftarrow M_i \cup X$
34: **else**
35: $H_i \leftarrow H_i \cup X$;
36: **end if**
37: update $\tilde{y}_i$;
38: **if** $\tilde{y}_i == 0$ **then** ▷ $M_i = \emptyset$
39: $k \leftarrow k + 2$;
40: **end if**
41: **if** $0 < \tilde{y}_i < y_0$ **then**
42: $k \leftarrow \max(0, \lfloor \log_2 \frac{1}{\tilde{y}_i} \rfloor - 2)$; ▷ $k \geq 0$
43: **end if**
44: **end if**
45: **end while**
46: $M \leftarrow M \cup M_i, W_{i+1} \leftarrow U \setminus M, i \leftarrow i + 1$;
47: **end while**

$$\tilde{y}_i = |M_i|/(|M_i| + |H_i|), \tag{2}$$

where $\tilde{y}_i$ denotes the estimation of y_i and $|\cdot|$ denotes the cardinality of a set. Since the sets H_i and M_i change constantly, the malicious user ratio $\tilde{y}_i$ also varies during the inspection process.

Based upon the value of $\tilde{y}_i$, the GTHI algorithm adaptively adjusts inspection strategies. Let $y_0, 0 \leq y_0 \leq 1$ be a threshold parameter that is chosen prior to the inspection (we explain how to choose the parameter y_0 later). At the ith round of inspection, if we have $\tilde{y}_i \geq y_0$, the GTHI algorithm applies the individual inspection strategy, by which just one user is probed at the next inspection step. If the inspection result is "clean", this user is obviously honest; otherwise, if the inspection result is "dirty", this user is clearly malicious.[3] On the other hand, if $0 < \tilde{y}_i < y_0$, the GTHI algorithm applies the group testing strategy, by which a group of users are probed as a whole in the next inspection step.

If the group testing strategy is adopted, the number of users to be probed is $2^k + 2^{k+1}$, with $k = \max\{0, \lfloor \log_2 \frac{1}{\tilde{y}_i} \rfloor - 2\}$. Obviously, if the inspection result is "clean", all the $2^k + 2^{k+1}$ users are honest; otherwise, if the inspection result is "dirty", a sub-inspector conducts one more inspection on a subset of 2^k users. As aforementioned, this reduces the user set containing malicious users to either size 2^k users or size 2^{k+1} users. Then, following a binary search process, we locate one malicious user after k or $k+1$ more inspection steps. In Fig. 3, after users $\{4, 5\}$ are identified as being honest, the user set H_1 is updated as $H_1 = \{1, 2, 3, 4, 5\}$. After user 6 is identified as being malicious, the user set M_1 is updated as $M_1 = \{6\}$, and hence we have $\tilde{y}_1 = \frac{|M_1|}{|H_1|+|M_1|} = \frac{1}{6}$. Without loss of generality, in Fig. 3, we assume $y_0 = 0.5$. Since $\tilde{y}_1 < y_0$, we have $k = \max\{0, \lfloor \log_2 6 \rfloor - 2\} = 0$. Thus, the number of users to be probed next is 3. As shown at the bottom of Fig. 3, the three users $\{16, 8, 9\}$ is probed, with an inspection result "dirty" being obtained. One more inspection step is then conducted on user 16, which is proved to be honest. Afterwards, a sub-inspector conducts another more inspection step on user 8. Since we get an inspection result "clean", we can know that user 8 is honest, and infer that user 9 is malicious.

Clearly, if the head inspector stops detecting reading anomalies, all malicious users are located and the inspection process can be terminated. With the cooperation of the head inspector, after all malicious users are located, the sub-inspectors can avoid useless inspection steps on the users whose statuses have not been determined but are actually honest. For example, in Fig. 3, after the malicious users $\{6, 9\}$ are identified, the sub-inspectors do not need to conduct more inspection steps on the users $\{10, 11, \ldots, 15, 7\}$ which are still left in set W_1 but are actually honest.

If there are fewer than 3 users left in W_i (i.e., $|W_i| < 3$), but there are malicious users among them, the users in W_i are probed individually. We conclude the above inspection strategies in Algorithm 2, where M denotes the whole set of malicious users. Technically, we have $M = \bigcup_{i=1}^{a} M_i$.

3. Note that during the whole inspection process, we can identify a user as malicious if and only if this user is probed alone and the corresponding inspection result is "dirty".

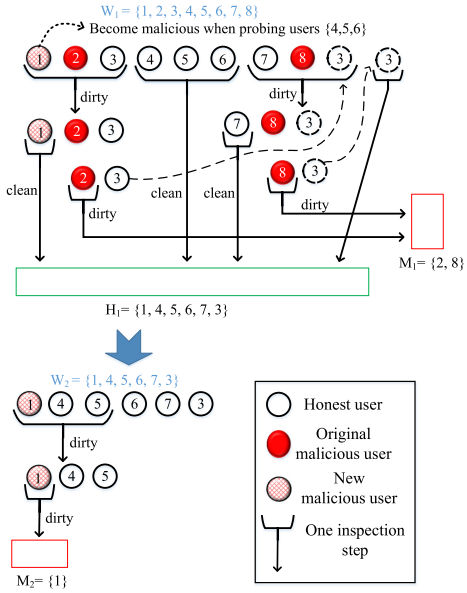


Fig. 4. An example of the GTHI algorithm under a dynamic case.

For better understanding, we illustrate how the GTHI algorithm runs in a dynamic case in Fig. 4. Assume that there are eight users $\{1, 2, \ldots, 8\}$ and that user 1 is honest at first but becomes malicious during the first round of inspection, where three users are probed at the first inspection step. With two more inspection steps, users $1, 2$ are identified as being honest and being malicious, respectively; user 3 is put back with the users left in set W_1. Thus, after the third inspection step, we have $y_1 = \frac{1}{2}$. Since $\max\{0, \lfloor \log_2 \frac{1}{y_1} \rfloor - 2\} = 0$, the number of users to be probed at the fourth inspection step is $2^0 + 2^1 = 3$. As can be seen, these three users are chosen as users $\{4, 5, 6\}$, which are identified as being honest. This means that we currently have $H_1 = \{1, 4, 5, 6\}$ and $M_1 = \{2\}$. Apparently, y_1 is now updated as $\frac{1}{5}$. Since $\max\{0, \lfloor \log_2 \frac{1}{y_1} \rfloor - 2\} = 0$, we further probe 3 users at the fifth inspection step, obtaining an inspection result "dirty". As can be observed, two more inspection steps are then consecutively conducted on user 7 and user 8, who are identified as being honest and being malicious, respectively. At the eighth inspection step, we identify user 3 as being honest. In Fig. 4, we assume that user 1 becomes malicious when the sub-inspector is probing users $\{4, 5, 6\}$ at the fourth inspection step. Thus, after statuses of all the users are identified as being either malicious or honest, the head inspector still can detect the existence of reading anomalies. This incurs the second round of inspection. As illustrated in the lower part of Fig. 4, to locate the new malicious user 1, the sub-inspectors conduct two more inspection steps on users $\{1, 4, 5, 6, 7, 3\}$ at the second round of inspection.

V. Performance Analysis

A. Bounds of the Number of Inspection Steps

In the following, we analyze the number of inspection steps conducted by the sub-inspectors when applying the GTHI

algorithm to locate malicious users under the static cases. Essentially, the inspection process under dynamic cases can be regarded as multiple rounds of the inspection process under static cases. Thus, we can conclude that better performances under static cases implies better performances under dynamic cases.

Let $t(n,m)$ denote the number of inspection steps conducted by the sub-inspectors when we apply the GTHI algorithm to locate m malicious users from a total number of n users under static cases.

Lemma 1 (Assume that there is only one malicious user). Then, we have 1) $t(n,1) \geq 1$, if $n < 3$; 2) $t(n,1) \geq 2$, if $n \geq 3$; 3) $t(n,1) \leq n$, if $n < 3$; 4) $t(n,1) \leq 3j$, if $\exists j \in \mathbf{N}^+$, $4^j - 1 \leq n < 4^j - 1 + 2^{2j-1}$; and 5) $t(n,1) \leq j + 1 + \lceil \log_2 (n - 4^j + 1) \rceil$, if $\exists j \in \mathbf{N}^+, 4^j - 1 + 2^{2j-1} < n \leq 4^{j+1} - 1$.

Proof. 1) If $n < 3$, the sub-inspectors probe users individually. If the user who is probed first happens to be the unique malicious user, the whole inspection process ends after one inspection step. 2) If $n \geq 3$, the sub-inspectors probe three users at the first inspection step and need to conduct at least one more inspection step to locate a malicious user. 3) When $n < 3$, we obtain the maximum number of inspection steps when the unique malicious user happens to be the user that is probed at the last inspection step.

4) and 5) a) When $n = 4^j - 1$, we obtain the maximum number of inspection steps in the following situation: all the inspection results from the first to the $(j-1)$th inspection steps are "clean", and the inspection result of the jth inspection step is "dirty". $2^{2j-2} + 2^{2j-1}$ users are probed at the jth inspection step. The sub-inspectors then conduct one more inspection step on 2^{2j-2} users out of them. If the inspection result is "dirty", then the sub-inspectors need to conduct $2j-2$ more inspection steps. Otherwise, if the inspection result is "clean", then the sub-inspectors need to conduct $2j-1$ more inspection steps. Thus, for the case $n = 4^j - 1$, we have $t(n,1) \leq \max\{j + 1 + (2j-2), j + 1 + (2j-1)\} = 3j$.

b) When $4^j - 1 < n \leq 4^{j+1} - 1$, assume that the unique malicious user is distributed among the first $4^j - 1$ users. Since the inspection process terminates after the unique malicious user is located, we can claim that the inspection process in this situation is the same as that in a). Therefore, we have $t(n,1) \leq 3j$.

c) When $4^j - 1 < n \leq 4^{j+1} - 1$, assume that the unique malicious user is distributed among the last $n - 4^j + 1$ users. Since the first $4^j - 1$ users are honest, the first j inspection steps conducted have inspection results "clean". Then, the sub-inspector conducts the $(j+1)$th inspection step on all the remaining $(n - 4^j + 1)$ users, obtaining an inspection result "dirty". Afterwards, a process of binary search starts. Until the unique malicious user is finally identified, the sub-inspectors have to conduct $\lceil \log_2(n - 4^j + 1) \rceil$ more inspection steps. Hence, we have $t(n,1) \leq j + 1 + \lceil \log_2(n - 4^j + 1) \rceil$.

d) Combining b) and c), we have $t(n,1) \leq \max\{3j, j + 1 + \lceil \log_2(n - 4^j + 1) \rceil\}$. When $n - 4^j + 1 < 2^{2j-1}$, we have $j + 1 + \lceil \log_2(n - 4^j + 1) \rceil < 3j$. Thus, we can conclude when $4^j - 1 < n < 4^j - 1 + 2^{2j-1}$, we have $t(n,1) \leq 3j$. After combining a), we complete the proof of 4). Otherwise, if $n - 4^j + 1 \geq 2^{2j-1}$, we have $j + 1 + \lceil \log_2(n - 4^j + 1) \rceil \geq 3j$, and thus, we complete the proof of 5). ■

Lemma 2. Assume that all of n users are malicious. We have $t(n,n) = n$, if $n < 3$ and $t(n,n) = n + 1$, if $n \geq 3$.

Proof. Similar to the proof analysis of Lemma 1, if $n < 3$, the users are probed individually and we can easily obtain $t(n,n) = n$. For the cases $n > 3$, three users are probed at the first inspection step. After the first malicious user is located at the second inspection step, the ratio of malicious user is estimated to be one. This implies that the remaining $(n-1)$ users are then probed one by one. Therefore, the total number of inspection steps is $2 + (n-1) = n + 1$. ■

Theorem 1. Assume that there are m malicious users. For $0 < m \leq \frac{n-1}{e}$ where e is the natural constant, we have $t(n,m) \leq \frac{3}{2}\log_2(n+1) + m\log_2\frac{n-1}{m} + 1.42(m-1)$.

Proof. Consider the *while* loop from line 3 to line 45 in Algorithm 2. There are three possible flows of this loop:

Flow 1: A sub-inspector probes disjoint user sets of sizes $2^0 + 2^1, 2^2 + 2^3, 2^4 + 2^5, \ldots$, until obtaining an inspection result "dirty". Assume that the sub-inspector gets a total number of $j-1$ inspection results "clean" before obtaining an inspection result "dirty". Then, the number of users probed at the jth inspection step is $2^{2(j-1)} + 2^{2j-1}$. According to the proof analysis of Lemma 1, we know that the sub-inspector locates one malicious user from a total number of $4^j - 1$ users and the maximum number of inspection steps to be conducted is $3j$. Since $4^j - 1 \leq n$, we can derive $j \leq \lfloor \frac{\log_2(n+1)}{2} \rfloor$.

Flow 2: After at least one malicious user is identified, the sub-inspectors probe $2^k + 2^{k+1}$ users, with $k = \max\{0, \lfloor \log_2 \frac{1}{\tilde{y}_i} \rfloor - 2\}$. If the inspection result is "clean", the $3 \cdot 2^k$ users are honest. Otherwise, there is at least one malicious user among them. In this case, we first probe 2^k users. If the inspection result is "dirty", then k more inspection steps are conducted to locate one malicious user from the 2^k users. Otherwise, if the inspection result is "clean", then $k+1$ more inspection steps are conducted toward the left 2^{k+1} users. Thus, to locate one malicious user from $3 \cdot 2^k$ users, the maximum number of inspection steps is $\max\{1 + 1 + k, 1 + 1 + (k+1)\} = k + 3$.

Flow 3: The users are probed individually.

Let m_i denote the total number of malicious users that are detected in all occurrences of Flow $i (i = 1, 2, 3)$. Apparently, we have $m_1 + m_2 + m_3 = m$, with $m_1 = 1$. Let $A_1, A_2, \ldots, A_{m_2}$ be the m_2 subsets that are inspected in the corresponding m_2 occurrences of Flow 2. Let $a_i = |A_i|, i = 1, 2, \ldots, m_2$. Clearly, for any A_i, we have $a_i = 2^k + 2^{k+1}$. According to the analysis of *F*low 2, to locate one malicious user from A_i, the sub-inspectors conduct at most $\log_2 \frac{a_i}{3} + 3$ inspection steps. Therefore, we can derive $t(n,m) \leq 3j + \sum_{i=1}^{m_2}(\log_2 \frac{a_i}{3} + 3) + m_3 \leq 3j + \sum_{i=1}^{m_2} (\log_2 a_i) + (3 - \log_2 3)m_2 + m_3$

From the convexity of $\log_2(x)$, it follows that $\sum_{i=1}^{m_2} \log_2 a_i \leq m_2 \log_2 \frac{\sum_{i=1}^{m_2} a_i}{m_2} \leq m_2 \log_2 \frac{n - m_1 - m_3}{m_2} \leq m_2 \log_2 \frac{n-1}{m_2}$. Let $f(x) = x \log_2 \frac{n-1}{x}$. Then, we have $f'(x) = \log_2 \frac{n-1}{x} - \frac{1}{\ln 2}$, where $\ln(\cdot)$ denotes the logarithm with the base of natural constant $e = 2.71828 \cdots$. Apparently, when $x < 2^{-\frac{1}{\ln 2}}(n-1) = \frac{n-1}{e}$, the function $f(x)$ increases monotonically. Thus, when $m \leq \frac{1}{e}(n-1)$, we can derive from the above equation that

$\sum_{i=1}^{m_2} \log_2 a_i \leq m_2 \log_2 \frac{n-1}{m_2} \leq m \log_2 \frac{n-1}{m}$. Substituting $j \leq \lfloor \frac{\log_2(n+1)}{2} \rfloor$ and $\sum_{i=1}^{m_2} \log_2 a_i \leq m \log_2 \frac{n-1}{m}$ into the previous result $t(n,m) \leq 3j + \sum_{i=1}^{m_2} (\log_2 a_i) + (3 - \log_2 3)m_2 + m_3$, we can derive $t(n,m) \leq 3j + \sum_{i=1}^{m_2} (\log_2 \frac{a_i}{3} + 3) + m_3 \leq 3j + m \log_2 \frac{n-1}{m} + (3 - \log_2 3)(m_2 + m_3) \leq \frac{3}{2} \log_2(n+1) + m \log_2 \frac{n-1}{m} + 1.42(m-1)$. ■

Theorem 2. Assume that there are m malicious users. For any $0 < m \leq n$, we have $t(n,m) \leq \frac{3}{2} \log_2(n+1) + \frac{\log_2 e}{e}(n-1) + 1.42(m-1)$, where e is the natural constant.

Proof. From the analysis in the proof of Theorem 1, we can know that the function $f(x) = x \log_2 \frac{n-1}{x}$ obtains the maximum value when $x = \frac{n-1}{e}$. Thus, from the inequality in Theorem 1, we derive $t(n,m) \leq \frac{3}{2} \log_2(n+1) + m \log_2 \frac{n-1}{m} + 1.42(m-1) \leq \frac{3}{2} \log_2(n+1) + \frac{\log_2 e}{e}(n-1) + 1.42(m-1)$. ■

B. Selection of Parameter y_0

On the whole, the GTHI algorithm applies two inspection strategies: (1) an individual inspection strategy whereby users are inspected one by one, as summarized in lines $4 \sim 12$ in Algorithm 2; (2) a group testing strategy whereby sub-inspectors probe a group of users for one inspection step, as summarized in lines $14 \sim 43$ in Algorithm 2.

During the inspection process, the value of y_0 plays an important role when the sub-inspectors determines which inspection strategy is to be applied. Specifically speaking, at the ith round of inspection, if $\tilde{y_i} \geq y_0$, the individual inspection strategy is applied; otherwise, if $0 \leq \tilde{y_i} < y_0$, the group testing strategy is employed. We next discuss how to choose the parameter y_0 such that the GTHI algorithm can achieve the best performance on average.

Theorem 3. Assume that we apply the GTHI algorithm to locate m malicious users. The average number of inspection steps achieves the minimum when $y_0 = \frac{1}{3}$.

Proof. As aforementioned, when a certain round of inspection starts, the sub-inspectors probe disjoint user sets of sizes $2^0 + 2^1, 2^2 + 2^3, 2^4 + 2^5, \ldots$, until an inspection result "dirty" is obtained. This process is not impacted by the parameter y_0, and hence omitted in the following discussion.

When $0 < \tilde{y_i} < y_0$, the number of users to be probed next is $2^k + 2^{k+1}$, with $k = \max\{0, \lfloor \log_2 \frac{1}{\tilde{y_i}} \rfloor - 2\}$. That is to say, if $\tilde{y_i} < \frac{1}{4}$, we have $k = \lfloor \log_2 \frac{1}{\tilde{y_i}} \rfloor - 2 \geq 0$; otherwise, if $\frac{1}{4} < \tilde{y_i} < y_0$, we have $k = 0$.

In the cases where $\tilde{y_i} < \frac{1}{4}$, the average number of malicious users among the $2^k + 2^{k+1}$ users is $\tilde{y_i}(2^k + 2^{k+1}) = 3\tilde{y_i} 2^{\lfloor \log_2 \frac{1}{\tilde{y_i}} \rfloor - 2} \leq \frac{3}{4} \tilde{y_i} 2^{\lfloor \log_2 \frac{1}{\tilde{y_i}} \rfloor} < 1$. To locate one malicious user from $2^k + 2^{k+1}$ users, if we apply the individual inspection strategy, the sub-inspectors conduct at most $3 \cdot 2^k$ inspection steps. In contrast, if we use the group testing strategy, the sub-inspectors conduct at most $k+2$ inspection steps. Since $k + 2 \leq 3 \cdot 2^k$, we can conclude that if $\tilde{y_i} < \frac{1}{4}$, the sub-inspectors conduct an average number of fewer inspection steps using the group testing strategy than that using the individual inspection strategy.

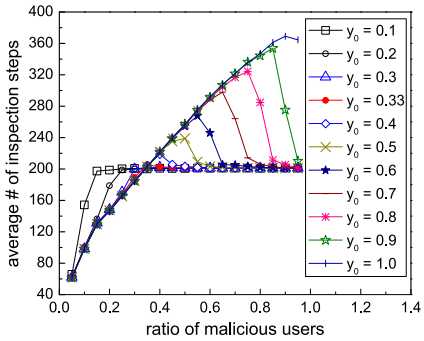
On the other hand, in the cases where $\frac{1}{4} < \tilde{y_i} < y_0$, the number of users to be probed next is $2^0 + 2^1 = 3$. If there is just one malicious user among the above three users, to locate this malicious user, the sub-inspectors conduct at most three inspection steps by both the group testing strategy and the individual inspection strategy. However, if there are more than two malicious users among the above three users, the individual inspection strategy outperforms the group testing strategy. Specifically speaking, by the group testing strategy, just one malicious user is located after the sub-inspectors conduct three inspection steps. In contrast, by the individual inspection strategy, all malicious users are located after the sub-inspectors conduct three inspection steps. Thus, if on average there is at most one malicious user among three users, the group testing strategy, on the whole, outperforms the individual inspection strategy. That is to say, if we have $3\tilde{y_i} \leq 1$, the group testing strategy should be applied. To achieve this, we should set $y_0 = \frac{1}{3}$. ■

C. Impacts of Threshold ε on Detection Accuracy

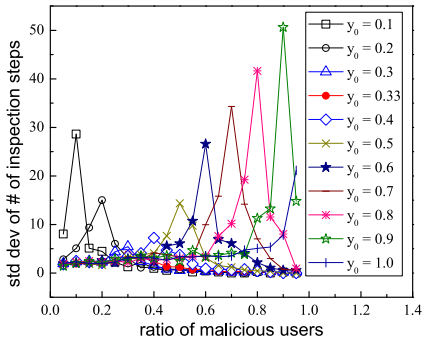
Since it is difficult to estimate users' technical losses accurately in practical applications, the threshold ε is introduced to guide the inspection process. The detection accuracy is defined as the ratio of the number of malicious and honest users who are identified correctly to the total number of users. We investigate the impacts of the threshold ε on the detection accuracy by conducting experiments. For easy implementation, all the users are assumed to experience the same technical losses. We assume that technical losses are accurately estimated. In the experiments, each user experiences about 1 kWh technical losses. We set the total number of users as 40 or 50, respectively, among which there are 10 malicious users whose stolen electricity at each period is randomly chosen from the interval $(0.1, 1)$ kWh. The simulation results show that as the threshold ε increases from 0 to 3, the detection accuracy declines gradually from 1 to about 0.93.

VI. Experiment

The experiments are conducted in Python 2.7.13 on an integrated development environment platform—PyCharm Community Edition 2017.1.3. The users' actual electricity consumption data are generated based upon a dataset of individual household electric power consumption in [38]. The data are measurements of electric power consumption in households with a one-minute sampling rate over a period of almost four years. Let q_0 denote the recorded electricity consumption of this individual. Then, q_j (the actual electricity consumption of user j) is generated as follows: $q_j = c_j q_0, \forall j \in U = \{1, 2, \ldots, n\}$, where c_j is a constant randomly chosen from the interval $(0, 2)$. In the experiments, honest users report their electricity consumptions as consumed. On the other hand, the reported readings of malicious users amount to just 10 to 50 percent of the actual electricity consumptions. We assume that users' technical losses are about 5 percent of the actual electricity consumptions. Note that each piece of data in the following figures is averaged over 30 times of repeated experiments.



(a) the average # of inspection steps



(b) the standard deviation of the # of inspection steps

Fig. 5. Static cases: $n = 200$.

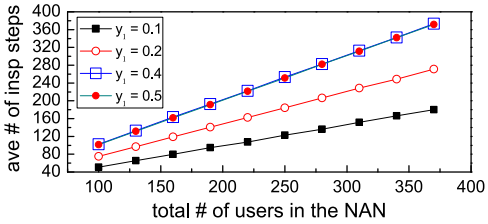


Fig. 6. Static cases: $y_0 = 0.33$.

A. Static Cases

In Fig. 5, we investigate how the average and the standard deviation of the number of inspection steps changes with the ratio (number) of malicious users.

Fig. 5(a) shows: 1) when $y_0 \leq 0.33$, the average number of inspection steps first increases and then stays stable; 2) for when $0.4 \leq y_0 \leq 1$, the average number of inspection steps first rises and then falls; 3) when $0 < y_0 \leq 0.33$, the maximum value is about 200; 4) when $0.4 \leq y_0 \leq 1$, the maximum value is greater than 200; 5) a larger y_0 implies a larger malicious user ratio at which the average number of inspection steps achieves its maximum value; 6) when $0.4 \leq y_0 \leq 1$, a larger y_0 also implies a greater maximum number of inspection steps. and 7) on the whole, regardless of the ratio of malicious users, the sub-inspectors conduct fewest inspection steps on average when we set $y_0 = 0.33$, and this verifies Lemma 3.

Fig. 5(b) shows: 1) for any given y_0, with y increasing from 0 to 1, the standard deviation of the number of inspection steps first rises and then falls; 2) regardless of the value of y_0, it achieves the maximum value when $y = y_0$; and 3) on the whole, the curve for the case $y_0 = 0.33$ is the flattest among all the curves.

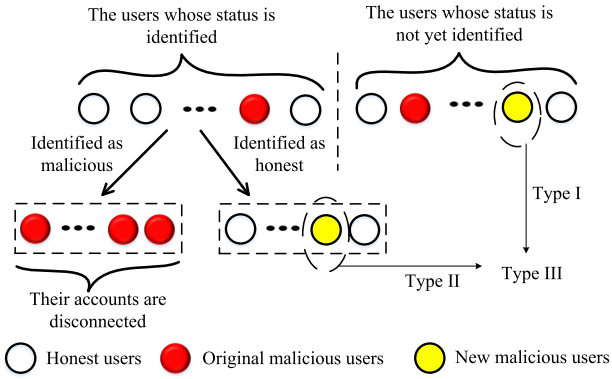


Fig. 7. Illustration for dynamic cases.

Fig. 6 shows how the average number of inspection steps changes with the total number of users, i.e., n: 1) regardless of the ratio of malicious users (i.e., y_1), the average number of inspection steps increases monotonically with n; 2) on the whole, for a given n, a greater y_1 implies more inspection steps; and 3) the curves for the two cases $y_1 = 0.4$ and $y_1 = 0.5$ coincide with each other and this is consistent with the results in Fig. 5.

B. Dynamic Cases

As aforementioned, the accounts of malicious users are disconnected once they are located. Thus, the new malicious users are likely to be among the users who are identified as being honest and/or the users whose status is not yet identified, as shown in Fig. 7. We categorize the dynamic cases into the following types: 1) Type I dynamic cases where new malicious users belong to the users whose status is not yet identified in the inspection process; 2) Type II dynamic cases where new malicious users belong to the users who are identified as being honest in the inspection process; 3) Type III dynamic cases where new malicious users belong to both the users who are identified as being honest and the users whose status is not yet identified in the inspection process.

For Type I dynamic cases, the sub-inspectors simply continue the current round of inspection on the users whose status is not yet determined. On the other hand, in Type II and Type III dynamic cases, the sub-inspectors conduct multiple rounds of inspections, and start a new round of inspection mainly for locating the new malicious users which have been identified as being honest.

Fig. 8 shows dynamic cases in which we assume that there appear new malicious users during the first round of inspection and the whole inspection process ends after the second round of inspection. Let h denote the number of new malicious users who start committing electricity theft after they are identified as being honest at the first round of inspection. Let w denote the number of new malicious users who start committing electricity theft before they are identified as being malicious or honest at the first round of inspection.

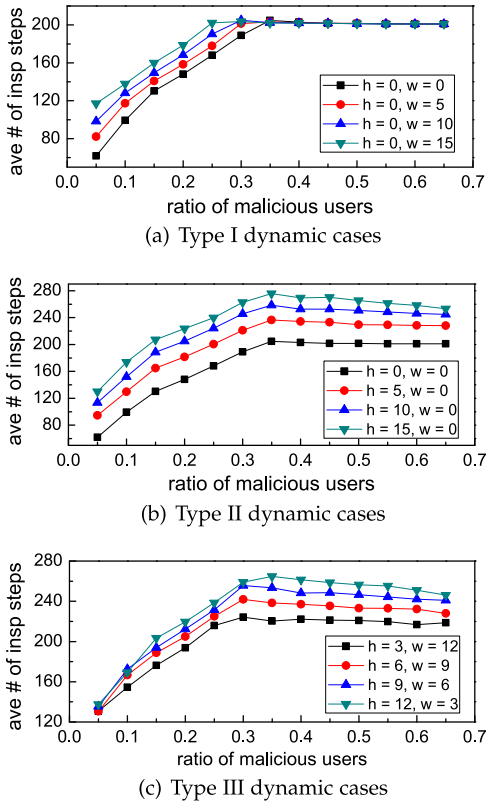


Fig. 8. Dynamic cases: $y_0 = 0.33, n = 200$.

Fig. 8(a) shows the Type I dynamic cases, where $h = 0, w > 0$. For comparison purposes, we show the curve of the case $h = 0, w = 0$. When $y \leq 0.35$, for a given the ratio of malicious users, a great w implies a larger average number of inspection steps. In contrast, when $y > 0.35$, regardless of the value of y or w, the average number of inspection steps stays almost 200 and this is because the users are usually inspected individually.

Fig. 8(b) shows the Type II dynamic cases, where $h > 0, w = 0$. Regardless of the ratio of malicious users, a greater h implies a larger average number of inspection steps.

Fig. 8(c) shows the Type III dynamic cases, where $h > 0, w > 0$. We consider four cases where the total number of new malicious users appearing in the first round of inspection is the same, i.e., 15. For a given ratio of malicious users, a smaller h implies a much smaller average number of inspection steps, even though it is with a greater w. This indicates that for the dynamic cases, h has a greater impact on the number of inspection steps than w. This is because the sub-inspectors do not need to incur another round of inspection to locate the new malicious users who start stealing electricity before they are identified as being honest or malicious. On the contrast, for the cases where $h > 0$, another round of inspection are surely started.

C. GTHI Versus BCGI

Fig. 9 compares the GTHI algorithm with the BCGI algorithm [28]. We assume that there is just one malicious user, in which case the BCGI algorithm can be applied. Fig. 9(a) shows that regardless of the total number of users, the sub-inspectors conduct many more inspection steps to locate the unique malicious user using the BCGI algorithm than using the GTHI algorithm. As pointed out in the paper [28], the BCGI algorithm conducts just one inspection step to locate a unique malicious user. However, we should not neglect the fact that for a given total number of users, the BCGI algorithm requires many more inspectors than the GTHI algorithm, which needs just two inspectors—one head inspector and one sub-inspector, as shown in Fig. 9(b). Above all, comparing to the BCGI algorithm, the GTHI algorithm has the greatest advantage that it can be applied to a wider range of applications. That is, it can be applied not only when there is one malicious user, but also when there are multiple malicious users.

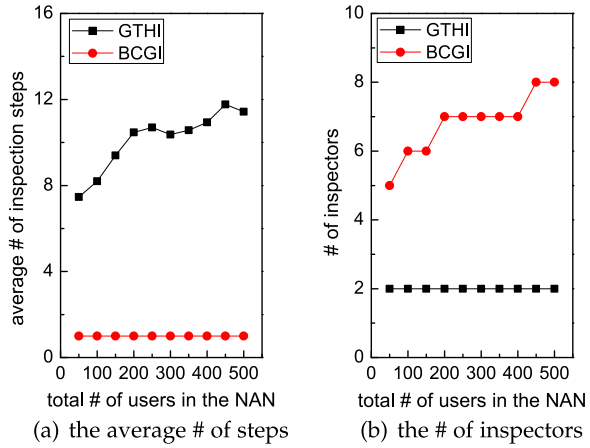


Fig. 9. GTHI versus BCGI. $y_0 = 0.33$.

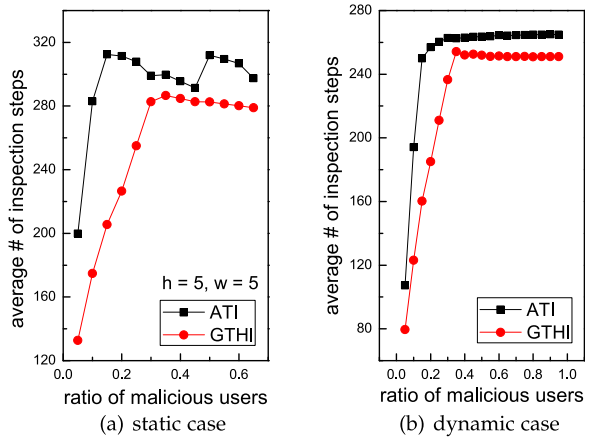


Fig. 10. GTHI versus ATI. $y_0 = 0.33$ and $n = 250$.

D. GTHI Versus ATI

Fig. 10 compares the GTHI algorithm with the ATI algorithm. Fig. 10(a) shows static cases. Regardless of the ratio of malicious users, the sub-inspectors conduct fewer inspection steps using the GTHI algorithm than using the ATI algorithm. For the ATI algorithm, the average number of inspection steps achieves the maximum when the ratio of malicious users is

about 0.2. As for the GTHI algorithm, it achieves the maximum number when y is about 0.35. The maximum number of inspection steps of the ATI algorithm is greater than that of the GTHI algorithm. The performance gap of the above two approaches first rises and then falls. It becomes the largest when the actual ratio of malicious users is between 0.1 and 0.2.

Fig. 10(b) shows the dynamic cases. We assume that the inspection process ends after the second round of inspection. We assume $h = 5$ and $w = 5$. With y increasing from 0.05 to 0.7, the average number of the GTHI algorithm increases monotonically until it reaches the maximum value, which is about 280. As a comparison, with regard to the ATI algorithm, the average number of the inspection steps has a repeated pattern of first increasing and then decreasing. In the dynamic case, we can also observe that regardless of the ratio of malicious users, the sub-inspectors conduct more inspection steps to locate the malicious users using the ATI algorithm than using the GTHI algorithm.

We also conduct experiments when the ratio of malicious users ranges from 0.01 to 0.1 (the figure was omitted). In both static cases and dynamic cases, the GTHI algorithm outperforms the ATI algorithm.

VII. Conclusion and Future Work

In this paper, we propose the GTHI algorithm which is able to estimate the ratio of malicious users on-line and adaptively adjusts inspection strategies during the inspection process. We derive the minimum upper bound of the number of inspection steps. It is proved that when the threshold for the estimated malicious user ratio is set as $\frac{1}{3}$, the GTHI algorithm can achieve the best performance on average. The GTHI algorithm can be applied in both static cases and dynamic cases. Just one round of inspection is conducted in static cases, whereas multiple rounds of inspection are usually conducted in dynamic cases. The GTHI algorithm outperforms existing methods in some aspects: compared with the BCGI algorithm, it has a wider range of applications; compared with the ATI algorithm, it can locate malicious users within much shorter detection time, regardless of the ratio of malicious users.

In future work, we will consider more categories of malicious users: 1) malicious users who start committing electricity theft from the beginning of the inspection process and do not turn into honest until caught by utility companies; 2) malicious users who start committing electricity theft in the middle of the inspection process and do not turn into honest until caught by utility companies; 3) malicious users who start committing electricity theft from the beginning of the inspection process and turn into honest before caught by utility companies; 4) malicious users who start committing electricity theft in the middle of the inspection process and turn into honest before caught by utility companies; 5) malicious users who intermittently commit electricity theft, i.e., they constantly repeat the process of stealing electricity and then turning into honest during the inspection process.

In this study, we consider static cases where there is only the first category of malicious users as well as the dynamic cases where both the first and the second categories of malicious users coexist. For future work, we will consider a more complex situation where more types of malicious users coexist.

Acknowledgment

The authors would like to thank Charlene Lucky Coburn for comprehensive editing and mentorship in English writing.

References

[1] I. Hosni and N. Hamdi, "Distributed cooperative spectrum sensing with wireless sensor network cluster architecture for smart grid communications," *Int. J. Sensor Netw.*, vol. 24, no. 2, pp. 118–124, 2017.

[2] B. Krebs, FBI: Smart Meter Hacks Likely to Spread. 2012. [Online]. Available: https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/

[3] Northeast Group, LLC, World Loses $89.3 Billion to Electricity Theft Annually, $58.7 Billion in Emerging Markets. 2014. [Online]. Available: http://www.prnewswire.com/news-releases/world-loses-893-billion-to-electricity-theft-annually-587-billion-in-emerging-markets-300006515.html

[4] W. Han and Y. Xiao, "NFD: Non-technical loss fraud detection in smart grid," *Comput. Security*, vol. 65, pp. 187–201, Mar. 2017.

[5] J. Mu, W. Song, W. Wang, and B. Zhang, "Self-healing hierarchical architecture for zigbee network in smart grid application," *Int. J. Sensor Netw.*, vol. 17, no. 2, pp. 130–137, 2015.

[6] J. Jow, Y. Xiao, and W. Han, "A survey of intrusion detection systems in smart grid," *Int. J. Sensor Netw.*, vol. 23, no. 3, pp. 170–186, 2017.

[7] J. Smith, Smart Meters Take Bite out of Electricity theft. 2011. [Online]. Available: http://news.nationalgeographic.com/news/energy/2011/09/110913-smart-meters-for-electricity-theft/

[8] C. H. Lo and N. Ansari, "Consumer: A novel hybrid intrusion detection system for distribution networks in smart grid," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 33–44, Jun. 2013.

[9] C. J. Bandim, J. E. R. Alves, A. V. Pinto, F. C. Souza, M. R. B. Loureiro, C. A. Magalhaes, and F. Galvez-Durand, "Identification of energy theft and tampered meters using a central observer meter: A mathematical approach," in *Proc. IEEE Power Energy Soc. Gen. Meet. Transmiss. Distrib. Conf. Expo.*, 7–12 Sept. 2003, pp. 163–168.

[10] 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, Jan. 2013.

[11] Z. Xiao, Y. Xiao, and D. H. C. Du, "Exploring malicious meter inspection in neighborhood area smart grids," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 214–226, Mar. 2013.

[12] T. Yucelen, W. M. Haddad, and E. M. Feron, "Adaptive control architectures for mitigating sensor attacks in cyber-physical systems," *Cyber-Phys. Syst.*, vol. 2, no. 1–4, pp. 24–52, 2016.

[13] C. Liu, S. Ghosal, Z. Jiang, and S. Sarkar, "An unsupervised anomaly detection approach using energy-based spatiotemporal graphical modeling," *Cyber-Phys. Syst.*, vol. 3, no. 1–4, pp. 66–102, 2017.

[14] A. H. Nizar, Z. Y. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 946–955, Aug. 2008.

[15] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad, "Nontechnical loss detection for metered customers in power utility using support vector machines," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 1162–1171, Apr. 2010.

[16] C. C. O. Ramos, A. N. de Sousa, J. P. Papa, and A. X. Falcao, "A new approach for nontechnical losses detection based on optimum-path forest," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 181–189, Feb. 2011.

[17] Y. Zhou, X. Chen, A. Y. Zomaya, L. Wang, and S. Hu, "A dynamic programming algorithm for leveraging probabilistic detection of energy theft in smart home," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 4, pp. 502–513, Dec. 2015.

[18] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 216–226, May 2016.

[19] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. S. Shen, "Energy-theft detection issues for advanced metering infrastructure in smart grid," *Tsinghua Sci. Technol.*, vol. 19, no. 2, pp. 105–120, Apr. 2014.

[20] K. Matsui, "An information provision system according to residents indoor comfort preferences for energy conservation," *Cyber-Phys. Syst.*, vol. 1-4, no. 2, pp. 121–142, 2017.
[21] W. Han and Y. Xiao, "NFD: A practical scheme to detect non-technical loss fraud in smart grid," in *Proc. IEEE Int. Conf. Commun.*, Jun. 10–14, 2014, pp. 605–609.
[22] W. Han and Y. Xiao, "CNFD: A novel scheme to detect colluded non-technical loss fraud in smart grid," in *Proc. Int. Conf. Wireless Algorithms Syst. Appl.*, Aug. 8–10, 2016, pp. 47–55.
[23] W. Han and Y. Xiao, "A novel detector to detect colluded non-technical loss frauds in smart grid," *Comput. Netw.*, vol. 117, pp. 19–31, Apr. 2017.
[24] W. Han and Y. Xiao, "FNFD: A fast scheme to detect and verify non-technical loss fraud in smart grid," in *Proc. ACM Int. Workshop Traffic Measurements Cybersecurity*, 30 May - 03 Jun. 2016, pp. 24–34.
[25] W. Han and Y. Xiao, "Design a fast non-technical loss fraud detector for smart grid," *Security Commun. Netw.*, vol. 9, no. 18, pp. 5116–5132, Dec. 2016.
[26] X. Xia, W. Liang, Y. Xiao, M. Zheng, and Z. Xiao, "A difference-comparison-based approach for malicious meter inspection in neighborhood area smart grids," in *Proc. IEEE Int. Conf. Commun.*, Jun. 8–12, 2015, pp. 802–807.
[27] X. Xia, W. Liang, Y. Xiao, and M. Zheng, "Difference-comparison-based malicious meter inspection in neighborhood area networks in smart grid," *Comput. J.*, vol. 60, pp. 1852–1870, Dec. 2017.
[28] X. Xia, W. Liang, Y. Xiao, and M. Zheng, "BCGI: A fast approach to detect malicious meters in neighborhood area smart grid," in *Proc. IEEE Int. Conf. Commun.*, Jun. 8–12, 2015, pp. 7228–7233.
[29] E. McCary and Y. Xiao, "Malicious device inspection home area network in smart grids," *Int. J. Sensor Netw.*, vol. 25, no. 1, pp. 45–62, 2017.
[30] R. Dorfman, "The detection of defective members of large populations," *The Ann. Math. Statist.*, vol. 14, no. 4, pp. 436–440, Dec. 1943.
[31] M. Sobel and P. A. Groll, "Group testing to eliminate efficiently all defectives in a binomial sample," *Bell Syst. Tech. J.*, vol. 38, pp. 1179–1252, 2017.
[32] S. Ma, Y. Yang, Y. Qian, H. Sharif, and M. Alahmad, "Energy harvesting for wireless sensor networks: Applications and challenges in smart grid," *Int. J. Sensor Netw.*, vol. 21, no. 4, pp. 226–241, 2016.
[33] P. N. Rao and R. Deekshit, "Energy loss estimation in distribution feeders," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1092–1100, Jun. 2006.
[34] Carolina Country, Stealing Electricity - Another Way to Get Electrocuted or Land in Jail, 2013. [Online]. Available: https://www.carolinacountry.com/your-energy/between-the-lines/departments/between-the-lines/stealing-electricity
[35] D.-Z. Du and F. K. Hwang, *Series on Applied Mathematics: Combinatorial Group Testing and Its Applications*, vol. 12, 2nd ed. Singapore: World Scientific Publishing Co., 2000.
[36] D.-Z. Du, G.-L. Xue, S.-Z. Sun, and S.-W. Cheng, "Modifications of competitive group testing," *SIAM J. Comput.*, vol. 23, pp. 82–96, Feb. 1994.
[37] A. Bar-Noy, F. K. Hwang, I. Kessler, and S. Kutten, "Competitive group testing in high speed networks," *Discrete Appl. Math.*, vol. 52, pp. 29–38, May 1994.
[38] UCI Machine Learning Repository, Individual Household Electric Power Consumption Data Set. 2012. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption/

Xiaofang Xia received the B.E. degree with Xiangtan University, Xiangtan, China, in 2012. She is currently working toward the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. She was a Visiting Scholar with the Department of Computer Science, University of Alabama, USA, from August 2016 to February 2018. Her research interests are mainly in cybersecurity and smart grid security.

Yang Xiao (Senior Member, IEEE) is currently a Professor with the Department of Computer Science, University of Alabama, Tuscaloosa, AL, USA. He has authored/coauthored more than 200 journal papers and more than 200 conference papers. His current research interests include cyber-physical systems, Internet of Things security, wired/wireless networks, smart grid, and telemedicine. He was a Voting Member of IEEE 802.11 Working Group from 2001 to 2004, involving IEEE 802.11 (WIFI) standardization work.

Wei Liang (Senior Member, IEEE) received the Ph.D. degree in mechatronic engineering from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2002. is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences, Beijing, China. As a Primary Participant/a Project Leader, she developed the WIA-PA and WIA-FA standards for industrial wireless networks, which are specified by IEC 62601 and IEC 62948, respectively. Her research interests include industrial wireless sensor networks and wireless body area networks. She was the recipient of the International Electrotechnical Commission 1906 Award in 2015 as a Distinguished Expert of industrial wireless network technology and standard.

Meng Zheng He received the B.S. degree in information and computing science and the M.S. degree in operational research and cybernetics from Northeastern University, Shenyang, China, in 2005 and 2008, respectively, and the Ph.D. degree in mechatronic engineering from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, in 2012. is currently an Associate Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. From 2010 to 2012, he was a Visiting Student with the Fraunhofer Institute for Telecommunication, Heinrich Hertz Institute, Berlin, Germany. His research interests include wireless ad hoc and sensor networks, cognitive radio networks, and security in smart grids.