IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2016, accepted May 2, 2016, date of publication May 9, 2016, date of current version June 24, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2564418

Optimal Trust System Placement in Smart Grid SCADA Networks

MD. MAHMUD HASAN, (Student Member, IEEE), AND HUSSEIN T. MOUFTAH, (Fellow, IEEE)
School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, K1N 6N5, Canada

Corresponding author: M. M. Hasan (mhasa101@uottawa.ca)

ABSTRACT The objective of this paper is to propose a trust system placement scheme for smart grid supervisory control and data acquisition (SCADA) networks. The functionalities of a trust system include firewalling and network intrusion detection. It is capable of monitoring both ingress traffic and egress traffic. In order to minimize the capital expenditure (CAPEX) and the operational expenditure (OPEX), only a selected number of nodes are equipped with trust systems. Those nodes are known as the trust nodes. This paper studies the trust system placement problem from a network topological perspective. It develops a scheme that aims to defend SCADA networks, deploying minimal number of trust nodes. It uses a network segmentation approach to distribute the trust nodes. It considers the minimum spanning tree (MST) as a measure of geographic dispersion. In the segmentation approach, size balancing and geographic dispersion are two main concerns. The segment sizes affect the number of required trust nodes. On the other hand, geographic dispersion affects the response time. The proposed scheme computes trust nodes using linear programming problem (LPP) formulations and local search. Numerical analysis is conducted through case studies for the IEEE test system topologies. It reveals the consistency of performance, better quality of protection, and low computational time. The proposed scheme can be a useful cyber security planning tool for smart grid operators.

INDEX TERMS Cyber security planning, Internet of Things (IoT), smart grid, optimization, supervisory control and data acquisition (SCADA).

I. INTRODUCTION

Cyber security is one of the major challenges in smart grid implementations. This happens due to the fact that smart grid operations are highly dependent on the information and communication technology (ICT) infrastructures [1]–[4]. Supervisory control and data acquisition (SCADA) is an integral part of modern power systems. It is also described as the wide area measurement system (WAMS) in the smart grid literature [5]. It is mainly responsible for conveying measurement information and control messages. At present, it is in its fourth generation of architectures. A key feature of the fourth generation is the adoption of advanced technologies such as cloud computing and Internet of Things (IoT). As a result, the scope of cyber security concerns becomes much wider. Unauthorized modifications of SCADA traffic may cause a number of negative events such as complete disruption of grid operations. Therefore, information integrity deserves the highest priority in SCADA communications. Unauthorized accesses or intrusion to a SCADA network must be detected using appropriate technologies. This is why smart grid operators are required to deploy trust systems to monitor SCADA traffic. The functionalities of a trust system include firewalling and network intrusion detection. In order to minimize capital expenditure (CAPEX) and operational expenditure (OPEX), only a selected number of nodes are equipped with trust systems. Those nodes are known as the trust nodes. Optimal selection of trust nodes provides better defense against cyber-attacks and minimizes costs. It deals with the total number of trust nodes and their network topological locations. The following recommendations are obtained from the prior studies in this area [6], [7].

- A network needs to be segmented into small pieces to limit the impact of malicious activities. This can prevent the spreading of cyber-attacks when bordering nodes are equipped with trust systems. In this case, bordering nodes are defined as the nodes that are connected by inter-segment links. Thus bordering nodes are capable of monitoring both ingress and egress traffic. Figure 1 shows an illustrative view of a segmented SCADA network where trust systems are hosted by the bordering nodes.
- Segmentation problems are computationally expensive. They are commonly solved using heuristic methods. As SCADA networks are geographically distributed, propagation delay and segment size can affect

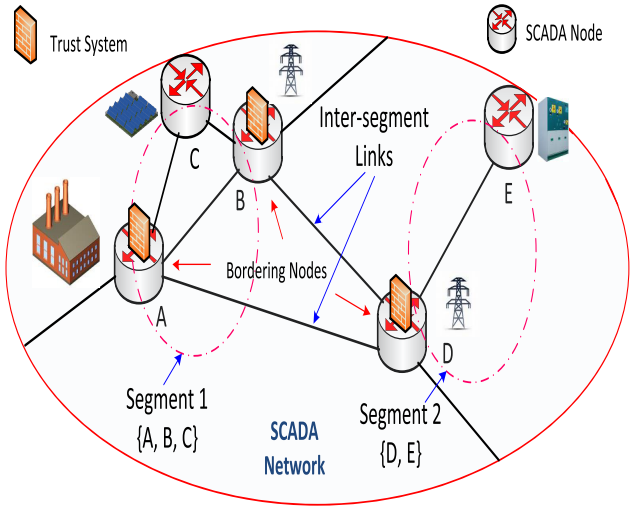

FIGURE 1. An illustrative view of a segmentation-based trust system placement.

time-critical communications. It is better to form a segment with nearby nodes. The difference in segment sizes affects the distribution of trust nodes. It is not always possible to obtain equal sizes due to topological constraints. Larger segments are more vulnerable to spreading cyber-attacks. They expose more tolerance for malicious activities due to the large number of unmonitored hops.

In this paper, we propose a trust system placement scheme for smart grid SCADA networks. Our scheme aims to defend SCADA networks deploying minimal number of trust nodes. Our main contributions in this paper are as follows:

- We propose a network segmentation approach based on the minimum spanning tree (MST) partitioning problem. The proposed approach is computationally lightweight. It also ensures that each computed segment forms a connected graph. This is important since disconnection in a segment interrupts the alert propagation initiated by trust nodes.
- Our proposed scheme is versatile since it is compatible with both of the possible cyber security planning approaches: (i) defending the system based on a given amount resources and (ii) defending the system based on its requirements. The first approach may cause both redundancy and insufficiency of deployed resources. We find the second approach is more reasonable for a smart grid environment. As deployment of trust systems requires installation of specialized hardwares, long-term planning is better. The second approach is suitable for long-term planning. Initially, we develop the scheme for the second approach. Subsequently, we show that it can also be used in the first approach.
- Our proposed trust system placement scheme can also be used to estimate the minimum number of trust nodes required to protect SCADA networks. It distributes at least one trust node to each inter-segment link. Due to the small computational time, it may help developing an interactive planning tool, where the desired number of segments is an input. As more number of segments require more resources, smart grid operators can decide the number based on available resources.

The rest of this paper is organized as follows. Section II briefly discusses related work and motivation. Section III discusses the IoT era of SCADA and smart grid. Section IV describes the trust system placement problem. Section V proposes the solution scheme. Section VI presents numerical results. Finally, Section VII provides conclusion and future work.

II. RELATED WORK AND MOTIVATION

In this section, we provide a precise discussion of the most relevant literature and our research motivation.

A. CYBER SECURITY ISSUES IN SMART GRID SCADA NETWORKS

A wide range of communication technologies are required to support various smart grid operations [8]–[10]. The conventional power systems mostly rely on standalone SCADA. In contrast, the smart grid environment adopts Internet-based SCADA systems for measurements and control functionalities. This incurs higher degree of cyber security challenges. In [11], a novel Internet-based attack on smart grid SCADA system has been reported. In that attack, intruding agents alters load status information in a system. The possible impacts include inappropriate control messages, unnecessary change in demand side price information, and erroneous inputs to load distribution algorithms. There are three types of such attack based on targets: (type I) targets power plants; it disrupts operation or generation, (type II) targets power distribution and control systems; it corrupts state information that may lead to instability, (type III) targets consumer premises; it cause increment of load that can damage the grid. Defense mechanisms to these attacks include private key encryption for unicast communications, message authentication codes, group key encryption for multicast communications, user authentication, password protected access, and firewalling of SCADA traffic. In [4], a comprehensive study of data integrity attack on SCADA systems has been reported. As data integrity influences the state estimation process, it is a major security concern for an energy management system (EMS). The study focused on an unobservable low sparsity cyber-attack that requires coordination of less than five energy meters. It is infeasible to place a meter in each power flow line in a large scale electric grid. Therefore, a phase measurement unit (PMU) placement algorithm was proposed. In [12], a comparison among SCADA cyber security standards has been presented. The most important criteria are countermeasures, possible attacks, and threats. In [5], a key management scheme named WAKE for WAMS has been proposed. The WAKE is suitable for both unicast and multicast protocols. Its design considers a large geographically distributed system that consists of PMUs, phasor data concentrators (PDCs), WAN, and real-time database and data archiver. The PMUs and PDCs are involved in multicasting for communication redundancy. The WAKE uses the X.509 and RFC 5280 based public key infrastructure (PKI).

Cyber security issues in SCADA networks are resolved using two major methodologies: (i) protection through monitoring and analysis; and (ii) cryptographic proctection. Trust systems are deployed to monitor and analyze network traffic. Our current work focuses on their optimal placement to accomplish efficient utilization.

B. TRUST SYSTEM PLACEMENT IN SMART GRID NETWORKS

We found two significant contributions in the area of trust system placement for smart grid networks. Those placement methods were proposed in [6] and [7]. The problem of placement was reported as an NP-hard problem in both cases.

In [7], a heuristic solution to the placement problem was proposed. The solution was obtained using a mixed integer linear programming (MILP) formulation for power system SCADA networks. The formulation considered a fixed number of trust systems for protecting a given SCADA network. In the network, links are weighted with propagation delays and nodes are weighted with a fixed packet processing delay. To attain the solution, the network is segmented into small pieces called domains or compartments. Those domains are formed based on preset timing thresholds. The number of segments are not predefined. It is only determined by the solution. Though it is a heuristic approach, the computational time is still considerably high for the large networks.

In [6], a layered network architecture was considered to address the trust system placement problem. The architecture was developed for smart grid advanced metering infrastructures (AMIs). A set packing problem was formulated to compute the hierarchical set of trust nodes. The problem includes a constraint called tolerance level. The constraint was introduced to set the maximum number of intermediary nodes between two trust nodes in each layer. A heuristic solution was proposed for secure and low overhead routing. It selects trust nodes from the interfacing nodes between different layers.

C. MOTIVATIONAL FACTS

In the light of the relevant literature, we found the following facts that motivate our current work:

- In the literature, trust system placement problems are reported as NP-hard. The existing heuristics are appropriate for certain scenarios of smart grid networks: (i) a given number of trust systems with timing thresholds [7], and (ii) layered architecture with tolerance levels [6]. There are still many more possible scenarios that are open for contributions. We consider a scenario of cyber security planning, where smart grid operators can decide the number of geographic divisions for their SCADA networks. Those divisions are termed as the network segments.
- In [6] and [7], the problem formulation used a fixed amount of packet processing delay for all nodes in a network. An identical value for the packet processing delay was assigned to every node. It becomes the weight for every node. As a result, nodes are not distinguishable based on their weights. In other words, nodes are identical in terms of weight. For a fixed number of hops, the value does not have any impact on route selections. Trust nodes are in fact computed based on their adjacent links' weights, which are propagation delays. Therefore, the placement problems can be simplified by excluding those 'identical' node weights. This simplification will not affect the solution. Our formulation excludes such node weights. Another fact about the node weights is their contributions to response timing considered in [7]. In reality, trust systems' response time depends on both types of delays fixed and random. There supposed to be a random queueing delay at every node. The queueing delay was ignored by assuming 'no congestion' in the network. Some cyber-attacks are in fact capable of causing network congestion. The distributed denial of service (DDoS) is a prominent example of such attack [13]. Therefore, satisfying timing thresholds based on the fixed delay parameters can only partially address the time criticality issue. Our formulation solves the placement problem from a topological perspective. It tries to minimize the geographic dispersion of each segment without setting thresholds. It assumes an expansion planning is necessary to resolve the time criticality issue. It is a realistic assumption since expansion planning is an integral part of a real network deployment process. From the time criticality point of view, our approach is qualitative rather than quantitative.
- The MST problem is a very useful concept in network design. It is used to obtain suboptimal solution to some computationally expensive problems such as the traveling salesman problem [14]. Our current work exploits the graph theoretic properties of MSTs to reduce computational times. A network can be approximated by its MST. In addition, we adopt the MST as a measurement of geographic dispersion for SCADA networks that are weighted by propagation delays.

III. THE IoT ERA OF SCADA AND SMART GRID

The fourth generation of SCADA architectures is currently being implemented. It is increasingly adopting advanced technologies such as cloud computing and IoT. Thus cloud-based services are available for supporting SCADA systems [15]. In the previous generations, SCADA activities were basically confined to proprietary networks. In contrast, the current generation is mostly Internet-based. This incurs additional cyber security risks in the system. Electric power grids are one of the most prominent application areas of SCADA systems. Their operations are heavily dependent on ICTs under the smart grid environment. Smart grid architectures include the Internet-based SCADA for supporting measurements and control functionalities [11]. Thus the fourth generation is compliant to the smart grid perspective. SCADA networks are deployed to serve three operational domains: generation, transmission, and distribution. As these domains are interconnected, cyber-attacks can easily propagate from one to another. Each domain is equipped with

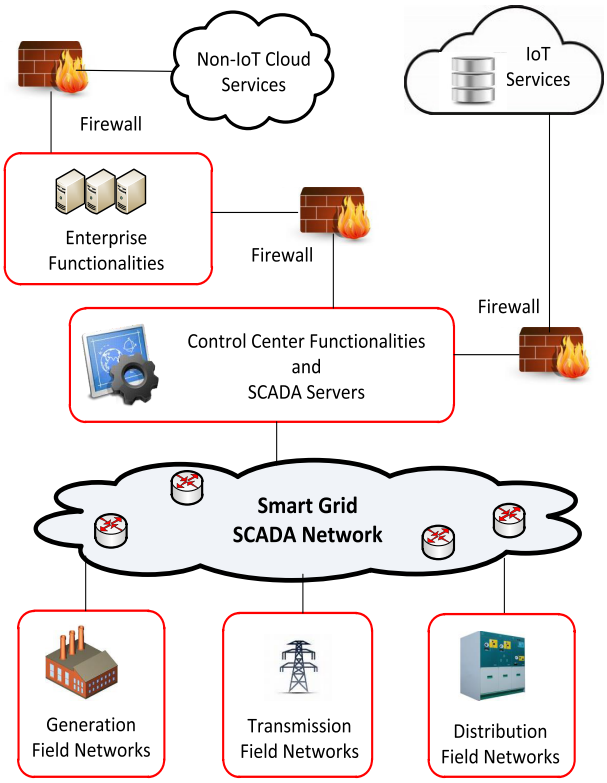

FIGURE 2. A conceptual depiction of smart grid SCADA systems in the IoT era.

field level devices such as remote terminal units (RTUs), programmable logic controllers (PLCs), and intelligent electronic devices (IEDs). These field level devices communicate with locally deployed sensors and actuators using a field network. They acquire measurements from sensors and delivers control commands to actuators. An operational domain includes a number of geographically distributed field networks. Typically, each power system bus is supported by a SCADA node and a field network. The SCADA node works as a gateway router for the field network. The SCADA network takes place between the smart grid control center and field networks. It conveys the acquired measurements to the control center, and then conveys control messages to field networks. The control center provides a wide range of functionalities such as system management, energy management, human machine interface (HMI) stations, forecasting and demand response, fault management, and asset management. The IoT technologies offer new Internet-based interfaces to ease control center functionalities. This creates opportunities for new services that use advanced data analysis and predictive analytics [16]–[18]. Figure 2 shows a conceptual depiction of smart grid SCADA systems in the IoT era. The enterprise functionalities include business tasks such as billing and dynamic pricing. They are separated from the control center and SCADA. The non-IoT cloud services are offered to enterprise functionalities. On the other hand, the IoT services are offered to SCADA and control center functionalities. The control center and SCADA components form a control network. There are three firewalls to monitor network traffic. Two of them are placed between smart grid and external service providers. The remaining one is placed between the enterprise and control networks. In the conventional power systems, the interface between external service providers and control network was absent. The smart grid adds new vulnerabilities to the SCADA network. Cyber-attacks can be originated from both domains internal and external.

IV. PROBLEM DESCRIPTION

Trust systems comprise specialized hardwares and software agents that monitor TCP and UDP traffic. They are capable of firewalling and intrusion detection [20]. They initiate and distribute alert messages whenever malicious activities are detected. They can be placed in smart grid network components to provide protection against cyber-attacks. Only a selected number of nodes can host trust systems due to budgetary constraints. Those host nodes are called trust nodes. The trust system placement problem deals with the selection of trust nodes.

In a smart grid environment, electric power grids are assumed to be accompanied by representative communication networks. It means that each SCADA node corresponds to a power grid bus. Thus each SCADA link corresponds to a power grid branch. Such SCADA networks can be represented by undirected graphs. In which, links are weighted based on their propagation delays. Network segmentation is a way of solving the trust system placement problem. Its advantages are twofold: (i) it limits the spreading of cyber-attacks and (ii) it provides economical solutions since only bordering nodes are eligible for hosting trust systems. As SCADA nodes are geographically distributed, geographic dispersion affects trust systems' response time. We consider the MST weight of each segment as a measure of geographic dispersion. Therefore, the ideal segmentation problem is described as follows. The objective is to minimize the sum of MST weights of all segments in a SCADA network. The minimum sum is obtained when the MST weight of each segment is minimized. The constraint is the maximum segment size. The constraint tries to keep the size as uniform as possible. In the mathematical formulation two additional constraints are required. There are two major sets: the SCADA node set (V) and the segment set (S). For any segment s, τ_s^{mst} is the MST weight and V^s is the set of member nodes. The objective function is given in (1) and defined in (2). The weight of a link e^s in the segment s is denoted by $w(e^s)$. The set of the MST link in a segment s is denoted by MST^s. Constraint (3) ensures that the whole network will be segmented. The union of all segment node sets returns the SCADA network node set. Constraint (4) ensures that a node can only be located in one segment. Thus the intersection of any two segment node sets is an empty set. Constraint (5) limits the maximum number of nodes located in one segment. The cardinality of SCADA node set, segment set, and node set of the segment s are denoted by N, K, and $|V^s|$ respectively. The exact solution

to this problem requires an exhaustive search to examine all the possible combinations of segments. The exhaustive search is computationally expensive. To meet the constraint (5), the network graph needs to be dense enough. Even for an exhaustive search, an optimum solution may not always exist due to topological constraints.

The Ideal Segmentation Problem:

$$\min \sum_{s \in S} \tau_s^{mst}, \tag{1}$$

where

$$\tau_s^{mst} = \sum_{e^s \in MST^s} w(e^s), \quad \forall s, \tag{2}$$

subject to

$$\bigcup_{s \in S} V^s = V, \tag{3}$$

$$V^s \cap V^{s'} = \varnothing, \quad \forall s \neq s' \text{ and } s, s' \in S, \tag{4}$$

$$|V^s| \leq \left\lceil \frac{N}{K} \right\rceil, \quad \forall s. \tag{5}$$

Whenever segments are computed, trust nodes are selected from the bordering node set between segments. Thus inter-segment traffic are monitored by trust systems. We discuss details of the trust node computation in the next section.

V. SOLUTION APPROACH

A segmentation approach is in the heart of our proposed solution. At first, segments are computed using a heuristic method. Bordering nodes are then identified to select hosts for trust systems. Before going to explain the solution approach, there are two important clarifications on terminologies used throughout this paper: (i) to be consistent, we use the network theory terms 'node' and 'link' instead of graph theory terms 'vertex' and 'edge', and (ii) for convenience, we use the term 'partition' when it is a subtree and the term 'segment' when it is a subgraph. Table 1 shows the symbols used in the solution approach. Symbols that are used only inside the algorithms are defined in the corresponding descriptions.

Our main considerations in the segmentation method are as follows.

• As an exhaustive search is computationally expensive, we develop a computationally lightweight algorithm based on the graph theoretic properties of MSTs. Inputs of the segmentation problem are the network graph and the number of segments; and output is the set of segments that indicates member nodes. Our method reduces the graph into its MST and then cut the MST into partitions. Each MST partition is belonging to a particular segment. Thus an MST partition can be treated as a segment skeleton. For a graph with N nodes, the MST contains $(N-1)$ number of links. To obtain partitions, we need to eliminate $(K-1)$ links from the MST. As a result, rest of the $(N-1)-(K-1) = N-K$ number of links create K partitions. The following proposition describes a useful property of MSTs.

Proposition 1: Any MST partition forms the local MST for its node set.

Proof: As the MST is the shortest path that connects all the nodes in a network, it is an optimum path. It implies that the principle of optimality is applicable to the MST [19]. According to the principle of optimality, the MST links are always belonging to the shortest paths computed for a given node set in the same network. Therefore, any partition of the MST is actually the local MST.

• Any MST partition is a connected graph. Therefore, any computed segment in our method is always a connected graph.

• In the segmentation problem, there is a trade off between the MST weight (propagation delay) and segment size. It is basically a multi-objective problem. It is computationally expensive to optimize both of them together. Moreover, uniform segment sizes are not always feasible due to topological

TABLE 1. Description of symbols.

Symbol	Description
$G(V, E)$	SCADA network graph. It is an undirected graph.
$T(V, E^{MST})$	MST of a given network $G(V, E)$.
V	Node set of the SCADA Network.
E	Link set of the SCADA Network.
N	Size of the SCADA Network in terms of node. $\|V\| = N$.
E^{MST}	MST Link set of the SCADA Network, $E^{MST} \subset E$ and $\|E^{MST}\| = N-1$.
α, β	Are weighting factors for multiple objectives.
S	Set of network segments.
K	Total number of segments to be created, $\|S\| = K$.
e, s, b	Are index variables for links, segments, and bordering nodes respectively.
$e_{u \leftrightarrow v}$	Undirected link between nodes u and v. It is an alternative representation for e.
d_u^{MST}, d_v^{MST}	Are degrees of the node u and v in the MST.
$d_{min}^{MST}(e_{u \leftrightarrow v})$	Minimum degree of the link between nodes u and v in the MST. It is defined by (6).
$w(e)$	Weight of the link e in terms of propagation delay.
$\widetilde{w}(e)$	Normalized weight of the link e with respect to the maximum link weight. It is defined by (9).
$X_I(l)$	Variable set for bordering nodes belonging to the inter-segment link l, $x_I \in X_I(l)$.
$L_{ss\prime}$	Set of inter-segment links between segments s and $s\prime$.
$B(s)$	Set of bordering nodes in the segment s.
Q	Total number of trust systems required.
M	Number of available trust systems.
Y	Vector for binary decision variables for the MST link elimination, $Y = (y_e)_{(N-1)\times 1}$.
X	Vector for binary decision variables for trust node selection, $X = (x_{sb})_{\sum_{s \in S} \|B(s)\| \times 1}$.

constraints. Therefore, we relax the constraint (5). In our MST based method, primary consideration is the topology awareness. And then we try to balance the segments as much as possible. To consider the impact of topology, we define a metric named the minimum degree of a link in the MST. It is defined as follows.

$$d_{min}^{MST}(e_{u\leftrightarrow v}) = \min(d_u^{MST}, d_v^{MST}). \tag{6}$$

The degree of a node refers to the number of links connected to that node. The metric in (6) helps identification of leaf nodes. It is also useful in the handling of star connections. The minimum degree of a link is always one if it connects a leaf node.

Our solution to the trust system placement problem can be divided into three major parts: (i) SCADA network segmentation, (ii) local search for repartitioning, and (iii) Trust node selection. They are presented by Algorithm 1, Algorithm 2, and Algorithm 3 respectively. Algorithm 2 is an embedded function within Algorithm 1. The output of Algorithm 1 is an input to Algorithm 3.

Two linear programming problems (LPPs) are solved as part of Algorithm 1 and Algorithm 3. It is worthwhile to describe them before describing the algorithms. The LPP1 is used to create initial partitions. It is a multi-objective optimization problem. It eliminates MST links based on their minimum degrees and normalized weights. These two parameters are combined using two weighting factors: α and β. As our first priority is the topology awareness, we set $\alpha = 1$ and $\beta = 0.5$. This value of beta ensures dominance of the minimum degree. The LPP1 is a maximization problem. The links with higher minimum degree and higher propagation delay are going to be eliminated so that partitions are formed. The decision variable of LPP1 is $Y = (y_e)_{(N-1)\times 1}$. It is a link incidence vector, such that

$$y_e = \begin{cases} 1, & \text{if } e \in E^{MST} \text{ is selected for elimination;} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The objective is given in (8). The normalized weight is defined in (9). Constraint (10) ensures the number of links to be eliminated. Constraint (11) ensures there will be no singleton nodes. It means a partition must contains at least two connected nodes.

LPP1: Initial Tree Partitioning:

$$\max_{Y} \sum_{e\in E^{MST}} \left(\alpha d_{min}^{MST}(e) + \beta \widetilde{w}(e)\right) y_e, \tag{8}$$

where

$$\widetilde{w}(e) = \frac{w(e)}{\arg\max_{w(e)} w(e)}, \quad \forall e \in E, \tag{9}$$

subject to

$$\sum_{e\in E^{MST}} y_e = K - 1, \tag{10}$$

$$y_e - d_{min}^{MST}(e) < 0, \quad \forall e \in E^{MST}, \tag{11}$$

$$y_e \in \{0, 1\}, \quad \forall e \in E^{MST}. \tag{12}$$

LPP2: Trust Node Computation:

$$\min_{X} \sum_{s\in S} \sum_{b\in B(s)} x_{sb}, \tag{13}$$

subject to

$$\sum_{x_I \in X_I(l)} x_I \geq 1, \quad \forall l \in L_{ss\prime};\ \forall s,\ s\prime \in S, \tag{14}$$

$$x_{sb} \in \{0, 1\}, \quad \forall s \in S \text{ and } \forall b \in B(s), \tag{15}$$

where

$$X_I(l) = \{x_{sb}, x_{s\prime b\prime}\}, \quad b \in B(s),\ b\prime \in B(s\prime),\ s \neq s\prime. \tag{16}$$

The LPP2 is used to select trust nodes when segments are ready. It is a minimization problem. It selects at least one bordering node from each inter-segment link. The value of the objective function indicates the number of trust system required to protect the network. The decision variable of is $X = (x_{sb})_{\sum_{s\in S}|B(s)|\times 1}$. It is a bordering node incidence vector, such that

$$x_{sb} = \begin{cases} 1, & \text{if } b \in B(s) \text{ is selected,} \quad s \in S; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The objective is given in (14). Constraint (15) ensures at least one trust node is selected for each inter-segment link.

Algorithm 1 is described as follows. After initialization, the MST of a SCADA network ($T(V, E^{MST})$) is computed using the Kruskal algorithm (Line 3). The minimum degree of each MST link is computed using (6). The weight of each MST link is normalized using (9). Initial partitions of the MST is obtained by solving LPP1. LPP1 computes E^I, the set of MST links to be eliminated (Line 8). The set of remaining MST links E^{SS} is obtained (Line 9). To identify the initial partition set, the function Disjointset is used (Line 10). It computes the node sets for each initial partition. To balance partition sizes, a local search is done (Line 11-27). For each pair of oversized and undersized partitions, the local search computes the minimum subpartition ($\triangle_{min}$) of the oversized partition that is adjacent to the undersized partition. If the size of $\triangle_{min}$ meets the condition for better balancing between partitions (Line 18), an adjustment for repartitioning occurs (Line 19). The subpartition is added to the undersized partition and deducted from the oversized partition. Thus both partitions are updated. The computation of $\triangle_{min}$ is described by Algorithm 2. The local search continues until there is a possibility of size balancing. If none of the pairs of oversized and undersized partitions remains adjustable (Line 24), the local search is terminated (Line 25).

Algorithm 2 is used to compute $\triangle_{min}$. Its inputs are node set of the oversized partition (s_i), node set of the undersized partition (s_j), and the MST. At first, it checks the adjacency of the partitions (Line 2). If the partitions are not adjacent, there will be no adjustment possible (Line 3). If the partitions are adjacent, it initializes the set of subpartition sizes (δ_{size}), the set of subpartition node sets (δ_{set}), and an indexing variable n (Line 5). It just recursively use the Disjointset function

Algorithm 1 SCADA Network Segmentation

Input: $G(V, E)$, K;
Output: $S = \{s_1, s_2, \ldots, s_K\}$;

1: **begin**
2: $E^{SS} = \varnothing, N = |V|, k_s = \left\lceil \frac{N}{K} \right\rceil, \phi = 0$; // Initialization
3: $T(V, E^{MST}) \leftarrow$ **Kruskal**$(G(V, E))$; // Computation of the MST using the Kruskal algorithm
4: **for all** $e \in E^{MST}$ **do**
5: Compute the minimum degree $d_{min}^{MST}(e)$;
6: Compute the normalized weight $\widetilde{w}(e)$;
7: **end for**
8: $E^I \leftarrow$ **Solve LPP1**; // This will select the $(K-1)$ number of links that needs to be eliminated to creat the initial tree partitions
9: $E^{SS} = \{E^{MST} \setminus E^I\}$; // Returns $(N-K)$ number of links
10: $S = \{s_1, s_2, \ldots, s_K\} \leftarrow$ **Disjointset**(V, E^{SS}); // Initial partition sets
11: **while** $\phi < 1$ **do**
12: Starting local search for the MST repartitioning
13: $count = 0$;
14: **for** $i = 1$ to K **do**
15: **for** $j = 1$ to K **do**
16: **if** $|s_i| > k_s$ **and** $|s_j| < k_s$ **then**
17: Compute $\triangle_{min}$ the minimum subpartition belonging to s_i that is adjacent to s_j
18: **if** $\big((|k_s - |s_i||) + (|k_s - |s_j||)\big) > \big((|k_s - |s_i| + |\triangle_{min}||) + (|k_s - |s_j| - |\triangle_{min}||)\big)$ **then**
19: $s_i = \{s_i \setminus \triangle_{min}\}$ **and** $s_j = \{s_j \cup \triangle_{min}\}$; // Updates the MST partitions
$count = count + 1$; // It checks for balancing segment sizes
20: **end if**
21: **end if**
22: **end for**
23: **end for**
24: **if** $count = 0$ **then**
25: $\phi = 1$; // Termination condition
26: **end if**
27: **end while**
28: **return** $S = \{s_1, s_2, \ldots, s_K\}$
29: **end**

Algorithm 2 Computation of $\triangle_{min}$

Input: s_i, s_j, and $T(V, E^{MST})$;
Output: $\triangle_{min}$;

1: **begin**
2: **if** $\{e_{u \leftrightarrow v} | u \in s_i, v \in s_j\} \cap E^{MST} = \varnothing$ **then**
3: $\triangle_{min} = \varnothing$;
4: **else**
5: $\delta_{size} = \varnothing, \delta_{set} = \varnothing, , n = 0$; //Initialization
6: $E^{ps_i} = \bigcup_{\forall a,b \in V^{s_i}} \{e_{a \leftrightarrow b} | e_{a \leftrightarrow b} \in E^{MST}\}$;
7: **for all** $e_{u \leftrightarrow z} \in E^{ps_i}$ **do**
8: $E^{ps_i} = \{E^{ps_i} \setminus e_{u \leftrightarrow z}\}$; // Cutting
9: $\{\triangle_u, \triangle_z\} \leftarrow$ **Disjointset**(s_i, E^{ps_i});
10: $n = n + 1$;
11: $\delta_{set}(n) = \triangle_u$;
12: $\delta_{size}(n) = |\triangle_u|$;
13: $E^{ps_i} = \{E^{ps_i} \cup e_{u \leftrightarrow z}\}$; // Restoring
14: **end for**
15: $n* = \arg\min_n \delta_{size}(n)$;
16: $\triangle_{min} = \delta_{set}(n*)$;
17: **end if**
18: **return** $\triangle_{min}$
19: **end**

Algorithm 3 Trust Node Selection

Input: $S = \{s_1, s_2, \ldots, s_K\}$, $G(V, E)$;
Output: V^{Trust};

1: **begin**
2: **for all** $s \in S$ **do**
3: $B(s) = \varnothing$; // Initialize bordering node sets
4: **end for**
5: **for all** $s \neq s\prime$ and $s, s\prime \in S$ **do**
6: $L_{ss\prime} = \varnothing$; // Initialize inter-segment link sets
7: **end for**
8: **for all** $e_{u \leftrightarrow v} \in E$ **do**
9: Find the segment x belongs to node u
10: Find the segment y belongs to node v
11: **if** $x \neq y$ **then**
12: $L_{xy} = \{L_{xy} \cup e\}$;
13: $B(x) = \{B(x) \cup u\}$;
14: $B(y) = \{B(y) \cup v\}$;
15: **end if**
16: **end for**
17: $V^{Trust} \leftarrow$ **Solve LPP2**; // This will select the trust node set
18: **return** V^{Trust}
19: **end**

and return the minimum adjustable subpartition. The set of the links belonging to the oversized partition E^{psi} is extracted (Line 6). The node u is located in the oversized partition and adjacent to the undersized partition. Each time a link belonging to u is chosen as a cut for the oversized partition (Line 8). In a tree graph, any link can be used as a cut. A cut divides the oversized partition into two subpartitions. The Disjointset function identifies those subpartitions (Line 9). One of the subpartitions contains u. This subpartition is eligible for partition adjustment. The sets δ_{size} and δ_{set} are updated (Line 11 and 12). After updating, the cut is restored to the E^{psi} (Line 13). The same process is done for all links of E^{psi} that are belonging to u. Thereafter, the minimum size is identified (Line 15) and the corresponding node set $\triangle_{min}$ is computed (Line 16).

Algorithm 3 is straight forward. Its inputs include the output of Algorithm 1 and the network graph. It computes the trust node set V^{Trust}. At first, it initializes the bordering node

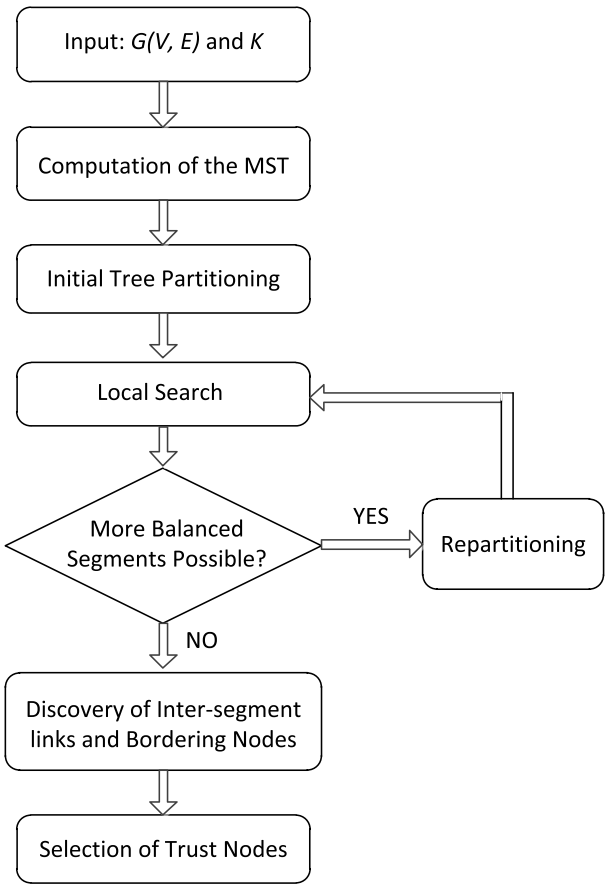

FIGURE 3. Flow diagram of the proposed scheme.

set $B(s)$ for each segment (Line 3). For each pair of segments, inter-segment link set $L_{ss'}$ is initialized (Line 6). For each link in the SCADA network graph, segments are identified for both nodes (Line 9 and 10). If nodes are belonging to different segments, the link is an inter-segment link (Line 11). Thus the inter-segment link set and bordering node sets are updated (Line 12-14). Once all bordering nodes are identified, LPP2 is solved to compute the trust node set (Line 17).

The Kruskal and the Disjointset algorithms are not shown here since they are well known in the literature and can be found in [21].

Figure 3 shows the complete flow diagram of the proposed trust system placement scheme. The worst case computational complexity of the proposed scheme is $\sim O(N^2)$, where N is the number of nodes in a SCADA network [see Appendix].

Figure 4 shows an illustrative example of a trust system placement problem for a small SCADA network. It precisely explains the proposed scheme with major steps. A given network graph with nine nodes and twelve undirected links to be segmented into three segments. The average segment size is three. The MST comprises eight links. To create initial partitions, two links need to be eliminated. Link (6, 7) has the highest minimum degree in the MST which is three. Therefore, it is eliminated. For the second link to eliminated there are three links with the minimum degree of

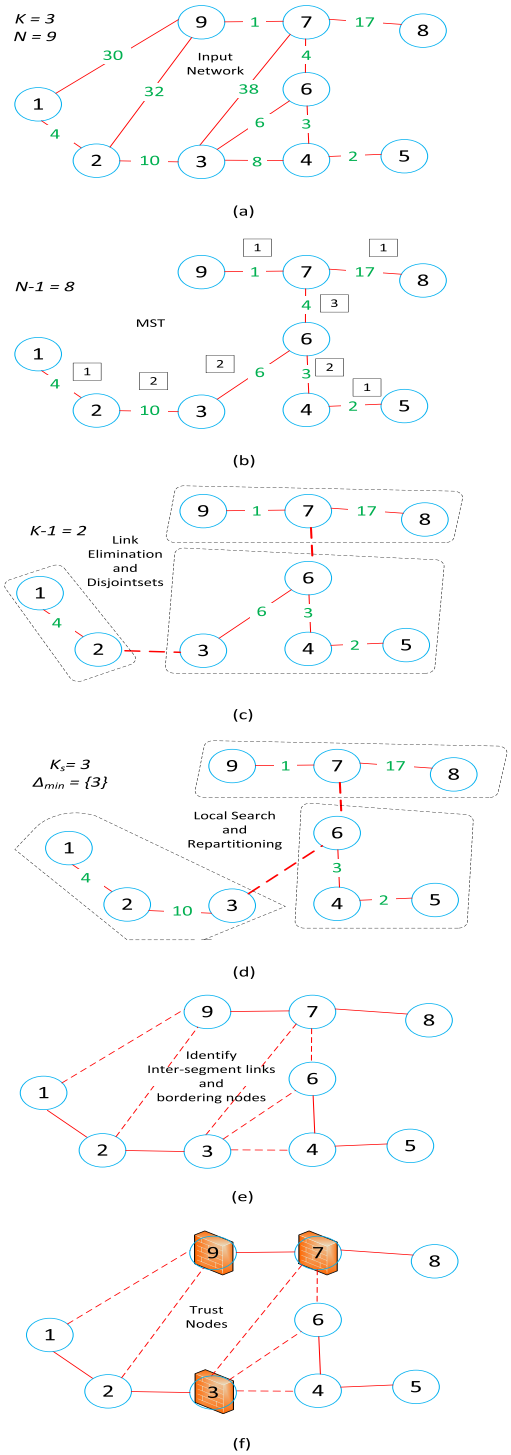

FIGURE 4. An illustrative example to explain the proposed scheme. (a) A SCADA network graph to be segmented into three pieces. The link weights are representing propagation delays. (b) The minimum spanning tree is computed using Kruskal. The minimum degree of each link is shown in the square box. (c) Initial tree partitions from LPP1. Links (6, 7) and (2, 3) are eliminated for their higher minimum degrees and weights. (d) Local search to adjust segment sizes. Repartitioning tries to balance between oversized and undersized segments. (e) Returning from the MST to the network graph. Discovery of Inter-segment Links and bordering nodes. There are six inter-segment links: (1, 9); (2, 9); (3, 4); (3, 6); (3, 7); and (6, 7). There are seven bordering nodes: 1, 2, 3, 4, 6, 7, and 9. (f) The required number of trust systems is minimized such that at least one trust system is placed per inter-segment link. LPP2 selected trust nodes are: 3, 7, 9.

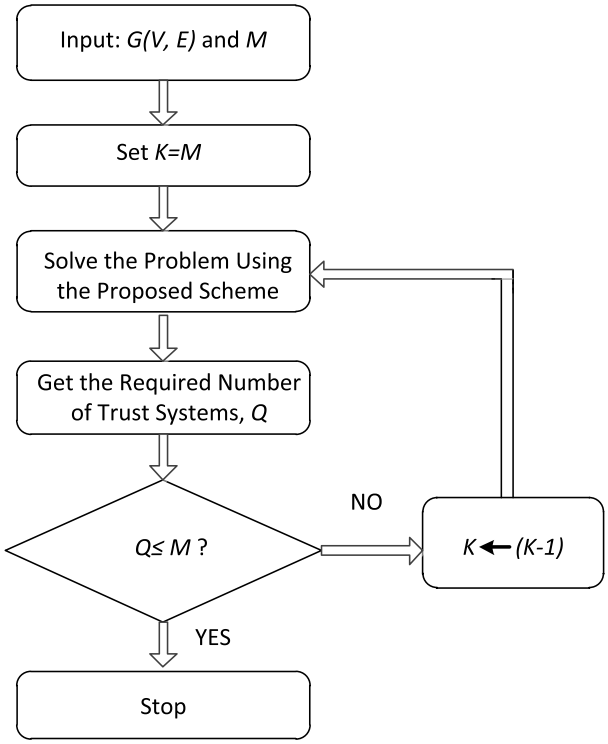

FIGURE 5. Converge with a different planning approach.

two: (2, 3);(3, 6); and (4, 6). Link (2, 3) has the highest weight among them. Therefore, it is eliminated. The partitions are identified using the Disjointset algorithm. Node sets for initial partitions are: {1, 2}; {3, 4, 5, 6}; and {7, 8, 9}. As the average segment size is three, the first partition is undersized and the second partition is oversized. These two partitions are adjacent. Node 3 belongs to the oversized partition. It is adjacent to the undersized partition. It becomes the subpartition to be adjusted for size balancing. As a result, node 3 is added to the undersized partition and removed from the oversized partition. The partitions are now balanced. After repartitioning, node sets for partitions become {1, 2, 3}; {4, 5, 6}; and {7, 8, 9}. These sets are the node sets for the corresponding segments. Inter-segment links and bordering nodes are identified with the help of the network graph. Inter-segment links are: (1,9); (2,9); (3,4); (3,6); (3,7); and (6,7). Bordering nodes are: 1, 2, 3, 4, 6, 7, and 9. Finally, the minimum number of trust nodes are selected from the bordering nodes such that each inter-segment link is monitored by at least one trust node. The trust nodes are: 3, 7, and 9.

Our proposed trust system placement scheme can also be applied in a different cyber security planning approach where the number of trust system (M) is given. Figure 5 shows the flow diagram for such convergence.

The main idea is to search an appropriate number of segments from an iterative procedure. If the estimated number of trust system (Q) is greater than M then new iteration with lower value of K is required. This continues unless $Q \leq M$. This proves the versatility of our proposed scheme.

TABLE 2. Summary of experimental parameters.

IEEE Test System Topology	Number of Nodes (Network Size)	Number of Active Links	Link Weight Mean (μs)	Link Weight Standard Deviation (μs)
BUS 14	14	20	19.55	18.84
BUS 30	30	42	24.1	24.67
BUS 57	57	78	22.33	34.41
BUS 118	118	179	8.35	6.22
BUS 300	300	409	14.59	39.21

VI. NUMERICAL RESULTS

Case studies are conducted for the IEEE test system topologies [22]. Table 2 shows the summary of experimental parameters. Propagation delays are calculated using the methodology introduced in [7]. The proposed scheme is implemented using the MATLAB optimization toolbox. The IEEE test system topologies are used as SCADA network graphs. We categorize the topologies based on their sizes: (i) small networks and (ii) large networks. Small networks include BUS 14, BUS 30, and BUS 57; and large networks include BUS 118 and BUS 300 systems. The number of segments (K) is varied to observe the performance of the proposed scheme. For small networks, the value of K is varied from 3 to 6 with increments of 1. For large networks, the value of K is varied from 5 to 30 with increments of 5. These values are chosen considering SCADA network sizes and the feasibility of segment sizes. For small networks, the average segment size is varied between 2.33 and 19. For large networks, the average segment size is varied between 3.93 and 60. All experiments are run on a desktop machine with Intel core i3 3.30 GHz CPU and 4 GB RAM. Numerical results are obtained from 100 runs for each combination of inputs. For a given combination of inputs, each run generates the same results but processing time varies depending on the instance of the desktop machine. This is why 100 runs are observed to obtain an average processing time.

A. PERFORMANCE OF THE SEGMENTATION METHOD

The performance of the segmentation method is evaluated based on segment sizes and geographic dispersion. For segment sizes, we choose the coefficient of variation as the metric. It refers to the ratio between standard deviation and average. It is also known as the normalized standard deviation. For a given number of segments, the average segment size in a network is a fixed number. The average remains the same for all methods. In this type of cases, the standard deviation is an important metric. As our experiments are conducted for different network sizes, normalized standard deviation is chosen. Figure 6(a) and 6(b) show the coefficient of variation of the computed segment sizes. The former is showing results for small networks and the latter is showing results for large networks. In both cases, the coefficient of variation is less than unity. Therefore, the computed segment sizes

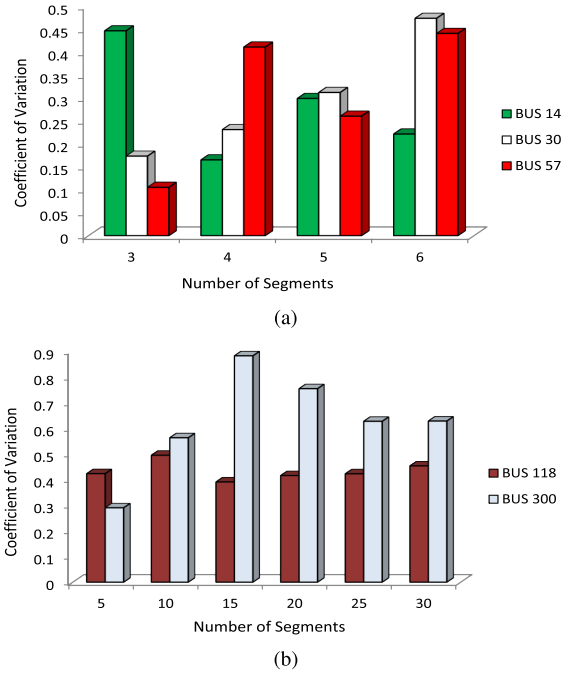

FIGURE 6. Coefficient of variation of the computed segment sizes. (a) Small Networks. (b) Large Networks.

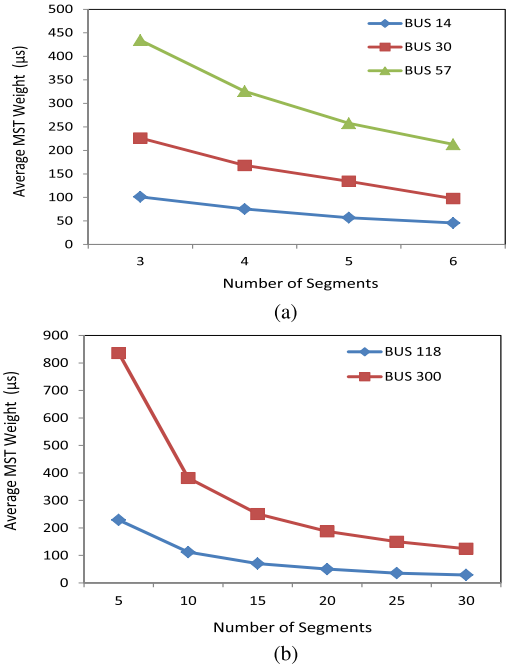

FIGURE 7. Average MST weight of the computed segments. (a) Small Networks. (b) Large Networks.

are low-variance. Large networks in 6(b) exhibits higher coefficient of variation than small networks in 6(a). This is because of the network size and the topological limitations of the MST. Segments of large networks are less balanced in size compared to that of small networks. As a result, the variation is larger.

For geographic dispersion, the average MST weight of the computed segments is the metric. Figure 7(a) and 7(b)

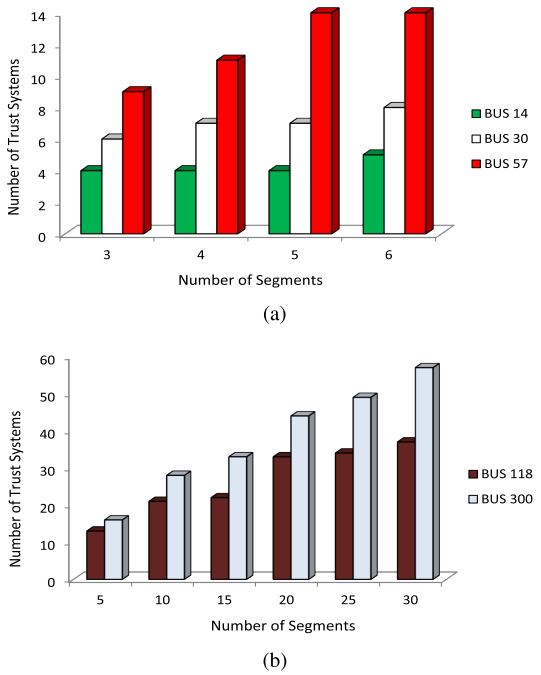

FIGURE 8. The required number of trust systems. (a) Small Networks. (b) Large Networks.

show the average MST weights for small networks and large networks respectively. It is clear that in both cases the average MST weights follow a decreasing trend as the number of segments increases. Higher values of K cause decrements in segment sizes. Thus less number of MST links are required for a segment. As a result the average MST weight is reduced. This reduction rate is higher for large networks.

B. ANALYSIS OF RESOURCE REQUIREMENTS

The metric for resource requirements is the required number of trust systems. Figure 8(a) and 8(b) show the required number of trust systems for the proposed scheme. We observed that the required number of trust systems follows an increasing trend with the number of segment increases. This is observed in all cases. The more number of segments in a SCADA network causes the more number of inter-segment links. As at least one trust system is placed to monitor an inter-segment link, the required number is increased. The amount of increment is higher for larger networks.

C. ANALYSIS OF PROCESSING TIMES

Figure 9(a) and 9(b) show the average processing time for the proposed scheme. For each combination of inputs, the average is obtained using 100 runs. We observe that the processing time mainly depends on the SCADA network size. It is clear that the number of segments has no significant impact on the processing time. The proposed scheme exhibits small processing time. Even for large networks such as the BUS 300, it can be run in a few minutes. This reveals the main advantage of the proposed scheme.

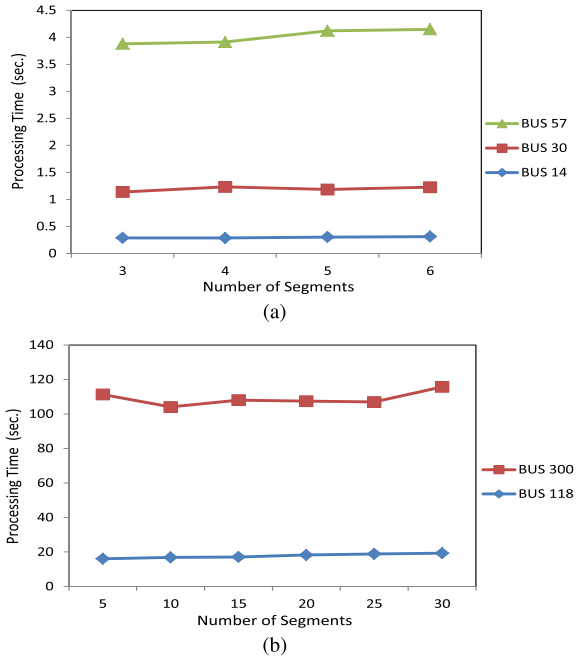

FIGURE 9. Average processing time of the proposed scheme. (a) Small Networks. (b) Large Networks.

D. IMPACTS OF TOPOLOGY AWARENESS: A COMPARATIVE ANALYSIS

As our proposed scheme offers a topology aware approach to trust system placement, it is worthwhile to investigate the impacts of topology awareness. To investigate the impacts of topology awareness, we compare the proposed scheme with a greedy approach. The greedy approach is developed by resetting the weighting factors, α and β. For the greedy approach, $\alpha = 0$ and $\beta = 1$. This value of α nullifies the impact of the minimum degree of MST links in the segmentation method. As a result, the segments are only created based on the propagation delays. The objective of the greedy approach is to deploy the minimum number of trust systems regardless topologies. It does not go for balancing of the segment sizes. We define an additional performance metric to compare between the greedy and the proposed scheme. The metric is named tolerance factor (η). It is defined as follows.

$$\eta = \frac{\text{Total number of unmonitored links in a network}}{\text{Total number of monitored links in a network}}. \tag{18}$$

We consider the tolerance factor as the quality of protection for SCADA networks. Less protected networks expose higher tolerance and *vice versa*. To obtain comparative performance, we only consider BUS 118 and BUS 300 topologies. Only large networks are chosen to demonstrate the difference between two approaches clearly. Figure 10(a) shows the comparative coefficient of variation of segment sizes. The greedy approach exhibits much higher coefficient of variation in all cases. As coefficients of variation for the greedy approach are greater than unity, the computed segment sizes

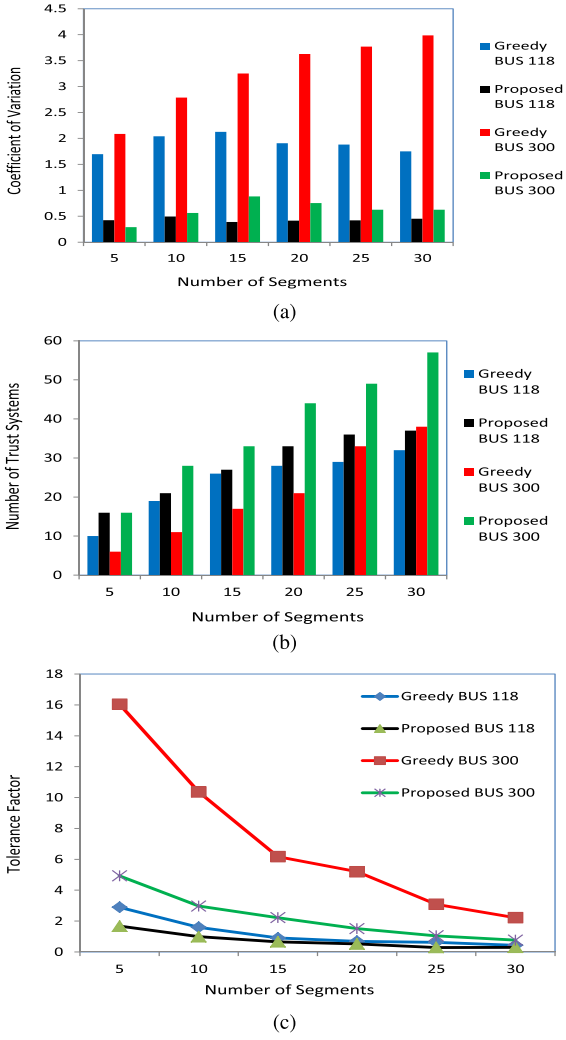

FIGURE 10. Comparative performance. (a) Comparative Coefficient of Variation. (b) Comparative Number of Trust Systems. (c) Comparative Tolerance Factor.

are high-variance. It means most of the segments are either heavily oversized or heavily undersized. Heavily oversized segments are more vulnerable to spreading cyber-attacks. Figure 10(b) shows the comparative number of required trust systems. It is clear that the greedy approach requires less number of trust systems in all cases. This seems to be an advantage of the greedy approach. This happens because of less number of bordering nodes are caused by heavily oversized segments. There is a trade-off between the amount of deployed security resources and the quality of protection. This is why it is necessary to investigate the quality of protection for both approaches. Figure 10(c) shows the comparative tolerance factor. In general, the tolerance factor is lower for higher value of K and *vice versa*. This is because of the number of deployed trust systems increases with the increment of K. We observe that the greedy approach exhibits higher tolerance in all cases. In particular, for the BUS 300 topology, the greedy approach exhibits much higher tolerance factor. It reveals that the proposed scheme offers better protection to SCADA networks. For topology awareness, trust nodes are

computed in such a way that the number of monitored links is as much as possible.

E. GIST OF RESULTS

The following important facts are extracted from the numerical results presented in this section.

- The proposed trust system placement scheme computes low-variance segment sizes. Thus it is effective in limiting the spreading of malicious activities. In particular, internal attacks from compromised nodes will be confined in a small portion of the SCADA network.
- The results on the coefficient of variation do not follow any particular trend. This is because of the sparsity of MSTs and topological constraints of network graphs. Our proposed scheme is a heuristic approach based on the MST. Despite the fact, consistent behavior is observed in the results on geographic dispersion of segments, resource requirements, and tolerance factor.
- The proposed scheme exhibits very small computational time. It is capable of computing trust nodes for the BUS 300 topology in a few minutes. Its computational time mainly depends on the SCADA network size. This implies that the proposed scheme can also be used as an estimation tool.
- The proposed scheme uses topology awareness to offer better quality of protection to SCADA networks.

VII. CONCLUSION

We have proposed and evaluated a lightweight trust system placement scheme for smart grid SCADA networks. We have introduced a heuristic algorithm based on the MST partitioning problem to segment SCADA networks in a smart grid environment. Our proposed scheme offers better quality of protection using topology-aware trust node selection. We have also shown that the scheme is compatible with both types of cyber security planning approaches: (i) optimal placement for a given number of trust systems while the number of segments is unknown and (ii) optimal location for a given number of segments while the number of trust systems is unknown. The scheme can be a used in developing interactive cyber security planning tool.

Edge routers are used as a gateway to SCADA networks for collecting and exchanging data for IoT services. Therefore, edge routers can be chosen from the trust nodes to detect external cyber-attacks. Though an inter-domain firewall is placed between the control network and IoT service providers, it will build a second line of defense.

Our proposed scheme is well-suited for the IEEE test system topologies. It is able to avoid the segments comprising stand-alone or singleton nodes. For other types of topologies, the feasibility of solution depends on the amount of star-connections. As star-connected portions cannot be segmented, each star connected portion becomes a segment [23]. If a network is heavily star-connected, the desired number of segments may not be computed without having singleton nodes. In such case, constraint (11) needs to be relaxed.

In the future work, we intend to develop an expansion planning approach to deal with response time issues and fault tolerance. The current approach only selects bordering nodes for hosting trust systems. In the expansion planning, additional trust nodes may require to serve internal nodes in a timely manner. A trust system can be unavailable for some reasons such as capacity outage, system failure, and link failure. This is why fault tolerance is another important consideration for the expansion planning. In addition to the expansion planning, consideration of different properties of trust systems can be another direction of the future research.

APPENDIX

The worst case can be explained as follows. The **while** loop is the heaviest part in the proposed scheme. The other parts such as Kruskal, Disjointset, LPP1, and LPP2 have lower complexity. The complexity of the Kruskal is $\sim O(|E|log(N))$. The complexity for the Disjointset is $\sim O(Nlog(N))$. The complexity of an LPP is roughly proportional to the number of constraints and logarithm of the number of variables [24]. Thus the complexity of LPP1 is $\sim O(Nlog(N-1))$ and the complexity of LPP2 is $\sim O(\sum_{s,s' \in S, s \neq s'} |L_{ss'}| log(\sum_{s \in S} |B(s)|))$. The **while** loop runs maximum N times for the slowest case. The minimum amount of update in a partition is a change of an MST link. At least one update is required per run to continue the loop. It will require the first $(N-1)$ runs for updating $(N-1)$ number of MST links and the last run for the termination condition. For each run, it will need to compute $\triangle_{min}$ for maximum N times. As power grid networks are very low density graphs, $|E| << N^2$ [25]. Therefore, the worst case computational complexity is $\sim O(N^2)$.

REFERENCES

[1] G. N. Ericsson, "Cyber security and power system communication—Essential parts of a smart grid infrastructure," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1501–1507, Jul. 2010.
[2] E. D. Knapp and R. Samani, *Applied Cyber Security and the Smart Grid*. Waltham, MA, USA, Elsevier, 2013.
[3] *The Smart Grid Interoperability Panel—Cyber Security Working Group, Guidelines for Smart Grid Cyber Security*, document NISTIR 7628, 2010.
[4] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1244–1253, Sep. 2013.
[5] Y. W. Law, M. Palaniswami, G. Kounga, and A. Lo, "WAKE: Key management scheme for wide-area measurement systems in smart grid," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 34–41, Jan. 2013.
[6] Y. Zhang, L. Wang, and W. Sun, "Trust system design optimization in smart grid network infrastructure," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 184–195, Mar. 2013.
[7] J. Gonzalez *et al.*, "Optimization of trust system placement for power grid security and compartmentalization," *IEEE Trans. Power Syst.*, vol. 26, no. 2, pp. 550–563, May 2011.
[8] H. Li, A. Dimitrovski, J. B. Song, Z. Han, and L. Qian, "Communication infrastructure design in cyber physical systems with applications in smart grids: A hybrid system framework," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1689–1708, Aug. 2014.
[9] A. A. Khan, M. H. Rehmani, and M. Reisslein, "Cognitive radio for smart grids: Survey of architectures, spectrum sensing mechanisms, and networking protocols," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 860–898, Oct. 2016.

[10] M. H. Rehmani, M. E. Kantarci, A. Rachedi, M. Radenkovic, and M. Reisslein, "Smart grids: A hub of interdisciplinary research," *IEEE Access*, vol. 3, pp. 3114–3118, 2015.
[11] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Distributed Internet-based load altering attacks against smart power grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 667–674, Dec. 2011.
[12] T. Sommestad, G. N. Ericsson, and J. Nordlander, "SCADA system cyber security—A comparison of standards," in *Proc. IEEE Power Energy Soc. General Meeting*, Jul. 2010, pp. 1–8.
[13] A. Shevtekar and N. Ansari, "Is it congestion or a DDoS attack?" *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 546–548, Jul. 2009.
[14] Y. Haxhimusa, W. G. Kropatsch, Z. Pizlo, A. Ion, and A. Lehrbaum, "Approximating TSP solution by MST based graph pyramid," in *Proc. 6th IAPR-TC*, 2007, pp. 295–306.
[15] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.
[16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surv. Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
[17] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A survey on Internet of Things from industrial market perspective," *IEEE Access*, vol. 2, pp. 1660–1679, 2014.
[18] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung, "Recent advances in industrial wireless sensor networks toward efficient management in IoT," *IEEE Access*, vol. 3, pp. 622–637, 2015.
[19] R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.
[20] G. M. Coates *et al.*, "A trust system architecture for SCADA network security," *IEEE Trans. Power Del.*, vol. 25, no. 1, pp. 158–169, Jan. 2010.
[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
[22] University of Washington, Seattle, WA, USA. (2016). *Power Systems Test Case Archive*. [Online]. Available: http://www.ee.washington.edu/research/pstca/
[23] M. Laszlo and S. Mukherjee, "Minimum spanning tree partitioning algorithm for microaggregation," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 7, pp. 902–911, Jul. 2005.
[24] V. Chvatal, *Linear Programming*. New York, NY, USA: Freeman, 1983.
[25] P. Hines, S. Blumsack, E. C. Sanchez, and C. Barrows, "The topological and electrical structure of power grids," in *Proc. 43rd HICSS*, Jan. 2010, pp. 1–10.

MD. MAHMUD HASAN received the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 2009. He is currently pursuing the Ph.D. degree with the University of Ottawa. His industrial experience includes operation and maintenance of multi-vendor telecom equipment. His current research area is optimization and cyber security for smart grid operations.

HUSSEIN T. MOUFTAH (F'90) joined the School of Electrical Engineering and Computer Science, University of Ottawa in 2002, as a Tier 1 Canada Research Chair Professor, where he became a Distinguished University Professor in 2006. He has been with the Electronics and Communication Department, Queen's University (1979-2002), where he was prior to his departure a Full Professor and the Department Associate Head. He has six years of industrial experience mainly with Bell Northern Research of Ottawa (Nortel Networks). He has authored or co-authored over ten books, 72 book chapters and over 1400 technical papers, 14 patents, six invention disclosures and 144 industrial reports. He is the joint holder of 20 Best/Outstanding Paper Awards. He is a fellow of the Canadian Academy of Engineering (2003), the Engineering Institute of Canada (2005) and the Royal Society of Canada RSC Academy of Science (2008). He has received numerous prestigious awards, such as the 2016 R.A. Fessenden Medal in Telecommunications Engineering of the IEEE Canada, the 2015 IEEE Ottawa Section Outstanding Educator Award, the 2014 Engineering Institute of Canada K. Y. Lo Medal, the 2014 Technical Achievement Award of the IEEE Communications Society Technical Committee on Wireless Ad Hoc and Sensor Networks, the 2007 Royal Society of Canada Thomas W. Eadie Medal, the 2007-2008 University of Ottawa Award for Excellence in Research, the 2008 ORION Leadership Award of Merit, the 2006 IEEE Canada McNaughton Gold Medal, the 2006 EIC Julian Smith Medal, the 2004 IEEE ComSoc Edwin Howard Armstrong Achievement Award, the 2004 George S. Glinski Award for Excellence in Research of the University of Ottawa Faculty of Engineering, the 1989 Engineering Medal for Research and Development of the Association of Professional Engineers of Ontario, and the Ontario Distinguished Researcher Award of the Ontario Innovation Trust. He served as the Editor-in-Chief of the IEEE *Communications Magazine* (1995-97) and the IEEE ComSoc Director of Magazines (1998-99), the Chair of the Awards Committee (2002-03), the Director of Education (2006-07), and a member of the Board of Governors (1997-99 and 2006-07). He has been a Distinguished Speaker of the IEEE Communications Society (2000-2007).