Con-Resistant Trust for Improved Reliability in a Smart-Grid Special Protection System

Crystal M. Shipman, Kenneth M. Hopkinson, *Senior Member, IEEE*, and Juan Lopez, Jr.

***Abstract*—This paper applies a con-resistant trust mechanism to improve the performance of a communications-based special protection system to enhance its effectiveness and resiliency. Smart grids incorporate modern information technologies to increase reliability and efficiency through better situational awareness. However, with the benefits of this new technology come the added risks associated with threats and vulnerabilities to the technology and to the critical infrastructure it supports. The research in this paper uses con-resistant trust to quickly identify malicious or malfunctioning (untrustworthy) protection system nodes to mitigate instabilities. The con-resistant trust mechanism allows protection system nodes to make trust assessments based on the node's cooperative and defective behaviors. These behaviors are observed via frequency readings which are prediodically reported. The trust architecture is tested in experiments by comparing a simulated special protection system with a con-resistant trust mechanism to one without the mechanism via an analysis of the variance statistical model. Simulation results show promise for the proposed con-resistant trust mechanism.**

***Index Terms*—Con-resistant trust, critical infrastructure, reputation-based trust, smart grid, special protection systems.**

I. Introduction

SMART-GRID technologies promise to modernize the power grid, improve efficiency and reliability, and help meet increasing power demands through better communication to facilitate coordination and situational awareness [1], [2], [3]. While full of potential, efforts to modernize the grid, on top of legacy systems, have created a cyber-infrastructure that is susceptible to threats and vulnerabilities [1], [4].

Special protection systems (SPS) detect power grid disturbances and take predetermined actions to counteract them in a controlled manner [5]. Large system disturbances, such as transient instabilities, require an immediate response to prevent cascading power outages. This paper is motivated by the proper SPS response to system disturbances, which can be complicated by malfunctions and malicious entities.

Fadul [6] created a context-specific reputation-based trust mechanism to improve SPS decision making in the presence of failures and disruptions caused by malfunctioning or malicious smart-grid components. This paper proposes an SPS with a con-resistant trust mechanism that can operate more effectively under such conditions. The con-resistant trust enhanced SPS uses load-shedding strategies to mitigate transient instabilities. The core contribution of this paper is in the application of existing trust algorithms, which have been used in cooperative cybersystems, to cyberphysical security. Three main questions are investigated. First, does an SPS with a con-resistant trust mechanism successfully determine and execute an appropriate load-shedding strategy during system-wide disturbances in the presence of untrustworthy (malicious or malfunctioning) agent nodes? Second, does an SPS with a con-resistant trust mechanism keep the system's steady frequency above a 58.8-Hz threshold? Third, can an SPS implemented with a con-resistant trust mechanism perform at least as well as Fadul's reputation-based trust mechanisms?

This paper is divided into seven sections. Section I is the introduction. Section II presents related work. Section III gives a model of the adversary. Section IV describes the proposed con-resistant trust mechanism. Section V covers the methodology and simulated test environment. Section VI presents results and Section VII concludes this paper.

II. Background

A. SPSs

SPSs detect disturbances and take predetermined actions to counteract the conditions in a controlled manner to return to an acceptable equilibrium [5]. Two common types of SPS schemes are based on generation rejection and underfrequency load shedding [7]. Generation rejection involves selectively tripping generating units during severe transmission system disturbances [8]. Load shedding reduces the connected load to a level that can be safely supplied by generation[8].

The communications-based smart grid enables better context awareness and better SPS protection system decisions for system disturbances [6]. Trust systems improve protection systems by estimating and acting on component reliability.

B. Reputation-Based Trust

Reputation-based trust is found in many computing systems where *trust* is based on observations and is often binary [9] in the sense that an entity is entirely trusted or untrusted. This paper uses a context-specific reputation-based trust. In context-specific reputation-based trust, an entity (the truster) trusts another entity (the trustee) with respect to a certain context [9], [10]. Here, *context* is synonymous with *service*. Context-specific trust is based on direct and indirect trust. Direct trust of an entity

Manuscript received April 03, 2014; revised July 27, 2014; accepted September 05, 2014. Date of publication September 15, 2014; date of current version January 21, 2015. Paper no. TPWRD-00377-2014.

The authors are with the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433 USA (e-mail: crystal.shipman@afit.edu; kenneth.hopkinson@afit.edu; juan.lopez@afit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRD.2014.2358074

evolves from direct interaction experiences with other entities [10]. In this model, the term *interaction* denotes an action regarding a context or service. Indirect trust occurs when there is no history of direct interactions between two entities. In this case, recommendations from trusted peers with direct interactions with the entity are considered [10].

C. Multimechanism Trust Model

Duncan's CTMS [11] extends Zhao's earlier work [12] to make trust decisions for satellite telecommand systems.

CTMS uses interactions and credentials to calculate entity trust values. Trust is based on the number of cooperative and defective interactions [13]. Trust is easily lost but hard to gain.

Duncan incorporated Salehi-Abari and White's [14] and Yu and Singh's [13] trust models, which are resistant to con-man (confidence-man) attacks. In such attacks, a con-man conducts a series of consecutive cooperative interactions to elevate its trust value. The con-man then defects, defrauding the victim. The con-man has two choices: 1) never interact with the victim again or 2) regain the lost trust with subsequent cooperative behavior. The con-man can then again con the victim [14].

While this paper concentrates on these trust models, many alternatives exist. Good survey articles on trust applied to computing domains were written by Sherchan, Nepal, and Paris [15] and Grandison and Sloman [16].

D. Reputation-Based Trust for SPSs

Fadul's reputation-based trust-management toolkit (TMT) [6] uses communication to improve decision making in the presence of failures and disruptions attributed to malfunctioning or malicious components. The TMT was applied to an SPS where it used reputation-based trust to improve fault-response times and resiliency to component faults/failures and communication errors. The TMT consists of three major modules that calculate entity trust values. The Trust Assignment Module uses context-sensitive information, such as frequency data from individuals, to determine trust values. The fault detection module uses error signals from frequency disturbance monitoring devices to detect faults. The decision module analyzes current power conditions and assigns trust values to decide the most reliable corrective action to contain a disturbance. The TMT uses a majority-rule algorithm where trust values are assigned based on information from multiple entities. Only entities that agree with the trusted majority are trusted. The TMT greedily selects trusted nodes to load shed.

III. Model of the Adversary

The malicious adversary in this paper's experiments has the ability to subvert five, ten, or fifteen loads. A subverted node intentionally misreports its frequency reading and will not shed load on command. This is a relatively simple adversary. It allows an investigation into the use of trust mechanisms to detect and respond to failures, which may be caused by an adversary with limited resources. This mimics scenarios ranging from actual malicious actors to merely adverse conditions.

Experiments focus on the first 0.5 s after a disturbance. If an SPS is successful, if it can shed enough load to keep the frequency at or above 58.8 Hz. The 58.8-Hz threshold is derived from physical generator constraints [17], [18].

The frequency is evaluated after 50 s of simulation time to judge the SPS's success or failure. The goal of the adversary was to disrupt frequency readings during the first 0.5 s to trick the SPS into shedding the wrong amount of load. This attack works because the SPS tested uses the generators' frequency measurements to determine the load to shed after a disturbance. The nodes do this by subtracting a random amount, up to 3.33% of the real frequency reading, at each reporting opportunity. The trust system in the experiments considers a reading to be faulty and, hence, malfunctioning, whenever its reported frequency is not within 3.6% of one standard deviation of the mean reported frequency from all nodes. As discussed in Section V-E, $3.6\% = C/10$, where $C = 1/e$. This is a simple marker for malicious behavior. In a real system, there are many potential trust markers, which could be used to cross-validate trust measurements and to look for a much wider range of behaviors and conditions. The intent of the simulations in this paper is to give a proof of concept prototyping a trust method by using just one simple type of trust marker with the implicit acknowledgement that a real system would involve additional trust markers to enhance robustness.

IV. Con-Resistant Trust

A. Trust Implementation

The trust-management system in this paper is primarily derived from Yu and Singh's work in reputation management in electronic interacting communities [13]. The goal is to avoid interactions with undesirable entities. The interaction trust (I-Trust) mechanism calculates and maintains I-Trust values, based on a particular interaction marker, for each of the agent nodes in the system. The interaction marker used in this scenario is the reported frequency for each agent node. The I-Trust value is calculated based on this interaction marker. The I-Trust value is compared to its peers to see if its reading is within 3.6% of one standard deviation of the mean reported frequency from all nodes. Agent nodes with an I-Trust value within this threshold are considered trusted and are otherwise untrusted. The method for calculating I-Trust is as follows.

B. How Trust is Calculated

To enforce the previously described trust implementation, an I-Trust value is defined below.

- **DEFINITION 1:** T_{jx} is the trust value assigned by the I-Trust mechanism to node j for interaction marker x. $-1 < T_{jx} < 1$ and T_{jx} is initialized to zero [13].

The I-Trust mechanism calculates a trust value for agent node j based upon the interactions involving agent node j affecting marker x. Positive and negative interactions can be defined in game theory as cooperation and defection, respectively [14]. The simulations in this paper only look at one sample trust marker based on frequency reporting to illustrate the concepts in this paper. In these simulations, an agent node is cooperating when it is reporting a frequency value within 3.6% of one standard deviation of the mean of all nodes' reported frequencies. An agent node is defecting when it is reporting a frequency value

TABLE I
SIMPLE INTERACTION TRUST ALGORITHM [11], [13]

T_{jx}	Cooperation Interaction by j	Defection Interaction by j
> 0	$T_{jx+1} = T_{jx} + \alpha_j(1 - T_{jx})$	$T_{jx+1} = \frac{T_{jx} + \beta_j}{1 - min(\lvert T_{jx}\rvert, \lvert\beta_j\rvert)}$
< 0	$T_{jx+1} = \frac{T_{jx} + \alpha_j}{1 - min(\lvert T_{jx}\rvert, \lvert\alpha_j\rvert)}$	$T_{jx+1} = T_{jx} + \beta_j(1 + T_{jx})$
$= 0$	α_j	β_j

TABLE II
CON-RESISTANT INTERACTION TRUST ALGORITHM [14]

Cooperation Interaction by j	Defection Interaction by j
$\alpha_{j+1} = \min(\alpha_j + \gamma_{cj}(\alpha_0 - \alpha_j), \alpha_0)$	$\alpha_{j+1} = \alpha_j(1 - \lvert\beta_j\rvert)$ $\beta_{j+1} = \beta_j - \gamma_{dj}(1 + \beta_j)$
$\gamma_{cj+1} = 1 - \lvert\beta_j\rvert$	$\gamma_{dj+1} = C \times \lvert T_{jx}\rvert$

equal to or below 58.8 Hz. Cooperation interaction by agent node j generates a positive evidence α_j, and a defection interaction by agent node j generates a negative evidence β_j. This requires $\alpha_j \geq 0$ and $\beta_j \leq 0$. Values for α_j and β_j can be either static or dynamic. Generally, trust relationships are set so that trust is easy to lose and hard to gain[13]. This relationship is achieved in DEFINITION 2 by requiring that $|\alpha_j| < |\beta_j|$.

- **DEFINITION 2:** After an interaction, the trust value T_{jx+1} is calculated by the algorithm in Table I which considers the previous trust value T_{jx}[13]. Once the calculations in Table I are complete, updates are made to α, β, and/or γ as in Table II. After these updates, a boundary check is executed so that $\alpha_{j+1} = min(\alpha_{j+1}, |\beta| - \epsilon)$, where ϵ is a small positive value to ensure that $|\alpha| < |\beta|$. Also, $\alpha_{j+1} = max(0, \alpha_{j+1})$ and $\beta_{j+1} = min(0, \beta_{j+1})$ to ensure that $\alpha \geq 0$ and $\beta \leq 0$.

The Simple Intraction Trust Algorithm[11], [13] in Table I calculates I-Trust values. In [14], Salehi-Abari and White tested this algorithm against a con-man attack, as defined in Section II-C. The authors showed that Yu and Singh's trust algorithm displayed con-resistance when its input parameters were chosen so that trust increments were small after a defection, and penalties for each defection were high. We apply this idea to create an SPS con-resistant trust algorithm.

To make Yu and Singh's [13] simple interaction trust algorithm resistant to a con-man attack, Salehi-Abari and White proposed adding two characteristics [14].

- **Cautiously increment trust with each cooperation:** An agent's corresponding trust value should be increased more slowly by each perceived consecutive cooperation.
- **Larger punishment after each defection:** An agent's corresponding trust value should be dropped more sharply by each perceived defection.

These characteristics are implemented by dynamically adjusting α_j and β_j based on agent interactions with node j. The modified trust value is defined in DEFINITION 3 below.

- **DEFINITION 3:** α_j and β_j are determined for Con-Resistant trust calculations by the algorithm in Table II, where C is a constant $0 < C \leq 1$ for node j.

The con-resistant interaction trust algorithm (I-Trust) [11], [14] in Table II extend Yu and Singh's simple interaction trust algorithm. Here, α_j is the positive reward for cooperation and β_j is the negative punishment for defection, as in the simple interaction trust algorithm. However, a defection will decrease α_j and will increase the absolute value of β_j based on the characteristics listed before. Forgiveness is slower when several defections have occurred, and punishments are bigger for those who defect more often [14].

The con-resistant trust algorithm introduces variables in its I-Trust value calculation. The initial value for α_j is preserved as α_0. Based on the equations presented in Table II, α_j will increase for each cooperation; however, it will never exceed α_0[14]. Furthermore, α_j is decreased at the rate of $1-|\beta_j|$ which results in a large decrement for α_j for a high value of $|\beta_j|$ and a small decrement of α_j for a low value of $|\beta_j|$[14]. Discounting factors, γ_{dj} and γ_{cj} as well as a constant C are also introduced. γ_{dj} is the discounting factor for a defection with respect to node j and is proportional to the absolute value of the previous I-Trust value T_{jx}. The authors hypothesized that the discounting factor γ_{dj} should be high when the target agent's I-Trust value is close to 1 (trusted) or -1 (untrusted) since, "Trust is hard to earn but easy to lose" [14]. Furthermore, if an agent has a high value of β because of previous defections, its α value should be increased more slowly when it is cooperating; thus, a node's γ_c value should decrease as the magnitude of its β value increases [14].

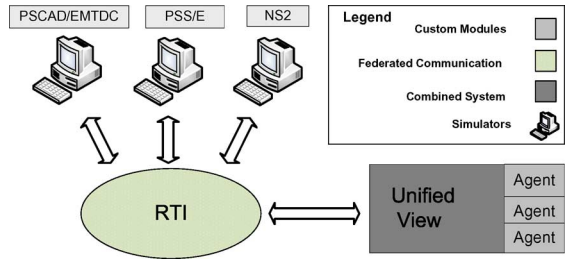

Fig. 1. EPOCHS simulation system [19].

V. METHODOLOGY AND SIMULATED TEST ENVIRONMENT

A. Simulation Environment

Article simulations make use of the electric power and communication synchronizing simulator (EPOCHS) [19]. EPOCHS (see Fig. 1) combines the PSS/E electromechanical transient simulator [20], the PSCAD/EMTDC electromagnetic transient simulator [21], and the NS2 network simulator [22]. The AgentHQ presents a unified environment to agents [19]. The RTI synchronizes timing between the simulators [19].

EPOCHS allows users to simulate wide-area smart-grid environments, such as the one depicted in Fig. 2. Such environments consist of wide-area communication networks interconnecting control centers, power generation plants, substations, and customers, traditional protection and control systems, and smart protection and control systems residing in smart remote terminal units (RTUs) and intelligent electronic devices (IEDs). EPOCHS has three agent types: control, load, and generator agents. There is one control agent node, 30 load agents, and 50 generator agents in the simulation scenario, which is based on the IEEE 50-Generator test case [23]. Each of these agent nodes

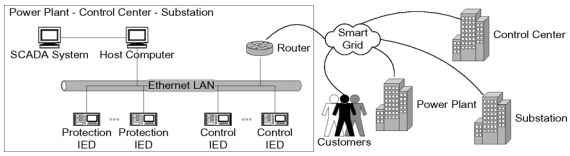

Fig. 2. Abstract representation of a smart-grid wide-area network [19].

has the ability to directly access and modify its corresponding power component's data and to use network communication. The communications network is based on 100-Mb/s lines.

B. Simulation Scenario

The simulation scenario uses a modified IEEE 50-generator/145-bus test case by Hopkinson *et al.* [19] to demonstrate the benefits of the con-resistant trust mechanism in an SPS.

The SPS monitors the system frequency for disturbances indicative of imminent fault and attempts to mitigate the fault by using two of the most common types of SPS schemes: 1) generation rejection and 2) load shedding [7]. Generation rejection reduces power to transfer over critical transmission interfaces [8]. Load shedding reduces the load to a level that can be safely supplied by available generation [8].

The scenario begins with two high-capacity transmission lines down, resulting in a transiently unstable system requiring SPS action. Generator 93 was preselected for generation rejection and commanded to trip or go offline by the SPS. Contingencies resulting from generator tripping cause an imbalance between generation and load [24]. Depending on power loss, the frequency can reach low levels. When this occurs, other generating units can trip, resulting in cascading events leading to system blackouts unless SPS actions are taken [24]. The frequency of 58.8 Hz is chosen as the frequency threshold for SPS success based on previous research [6], [19], [25]. Operating too far below this threshold can cause an increase in generator turbine vibrations, ultimately damaging the generator causing it to fail [8].

The additional SPS action taken is load shedding. The goal is to shed enough load to keep the system's frequency above a preset level following a system disturbance, such as generation loss [19]. The SPS uses an algorithm to estimate the system's disturbance size and the amount of load shedding required to maintain the system frequency above 58.8 Hz [19] using

$$P_d = P_a + \Delta P_e(\omega_{0+} - \omega_{0-}, v_{0+} - v_{0-}). \quad (1)$$

In (1), the size of the disturbance P_d is equal to the accelerating power P_a, which is proportional to the change in the system's frequency, plus the change in electrical power demand ΔP_e due to the variation in frequency and voltage. P_d is the key to finding the amount of lost generation. $0-$ and $0+$ denote the time immediately before and after the disturbance. P_a and ΔP_e can be obtained from wide-area measurements of the generators' operating status and system frequency samples before and after the disturbance, but measurements must be taken simultaneously throughout the region. [19].

The maximum load shed amount is set to 20% of the available load, which is a typical value [8], [26]. Once the load shed amount is determined, it is imposed on selected loads. A sorting algorithm determines which nodes are selected for load shedding. In the original SPS, loads are sorted based on their available load-shed amounts. In the SPS with con-resistant trust, loads are sorted by their assigned trust values and available load-shed amounts. The *goal* is to load shed a calculated amount to keep the frequency above 58.8 Hz [19].

C. Abuse Case

The abuse case is based upon the SPS's ability to detect untrustworthy agent behavior during system updates. Each simulation runs for 50 s to ensure that the system has stabilized. Throughout the simulation, the SPS's control agent node receives updates from load and generator agent nodes every 2o ms. These updates include the load agent's current operating frequency level. The con-resistant trust mechanism also calculates and reports the I-Trust values during each interaction. At 0.18 s, generator 93 is commanded to trip and at a time of 0.184 s, it goes offline. Four milliseconds later at time 0.192 s, the SPS makes the determination of which loads to shed based on trust values.

D. Performance Metrics

The primary metric used to evaluate the SPS's performance with and without con-resistant trust is the system frequency. Success is determined by the SPS's ability to accurately identify which load agents are trustworthy and untrustworthy and its ability to select a sufficient number of trustworthy nodes to shed load to keep the steady-state frequency above 58.8 Hz. The critical system frequency threshold is 58.8 Hz because operating below this threshold can cause an increase in damaging generator turbine vibrations or can lead to cascading events and blackouts through protection actions [8].

E. System Parameters

Systems parameters are characteristics that can affect the performance of the SPS with con-resistant trust. They include the frequency tolerance α_j and β_j, and the constant C.

In this paper, untrustworhy nodes have a tolerance value subtracted from their reported frequency, which is randomized to simulate realistic fluctuations due to inherent noise.

Abari and White's simple interaction trust algorithm used α and β values based on a 1 to 10 penalty ratio of cooperative to defective interactions [14]. In this paper, a 1 to 3 penalty ratio is utilized for the cooperative and defective interactions, specifically 0.15 for α and -0.45 for β.

One goal of the con-resistant trust system is to detect an untrustworthy node even if it only begins defective behavior at the start of a fault that requires an SPS response. The total time allowed to calculate the load to shed and shed load to prevent a frequency drop below 58.8 Hz is 0.5 s.

C is a multiple used to calculate the defection discounting factor γ_{dj} with respect to node j, as in Table II. Abari and White [14] kept C between zero and one with a specific value of $1/e$. This paper also uses a constant $C = 1/e$.

F. Experimental Design and Evaluation

The SPS with con-resistant trust is evaluated via simulation. An analysis of variance (ANOVA) and a comparison of con-

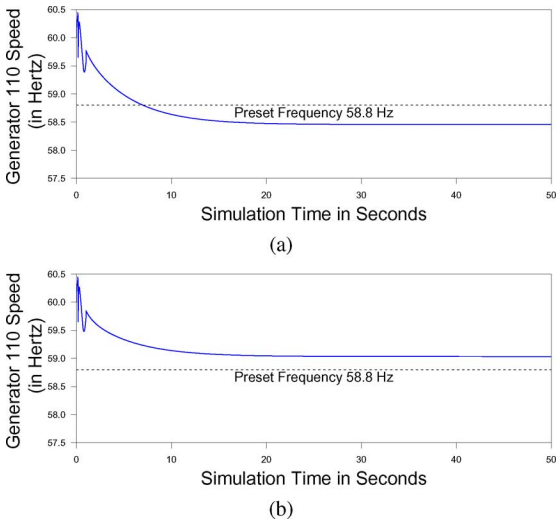

Fig. 3. Comparison of SPS behavior with and without con-resistant trust. (a) The original SPS is unable to keep the frequency above 58.8 Hz. (b) The SPS with con-resistant trust keeps the frequency above 58.8 Hz.

fidence intervals via the R statistical package [27], [28] are used to determine the simulation results' statistical significance. ANOVA tests whether the means among two or more groups are equal, assuming normally distributed populations [29].

SPS experiments have two factors: 1) the number of untrustworthy nodes and 2) whether the SPS uses con-resistant trust. There are three treatment levels of untrustworthy nodes: five, ten, or fifteen. Untrustworthy nodes are chosen at random from 30 loads. Each simulation is replicated 36 times. Thirty-six replicated trials per simulation were chosen because the central limit theorem generally takes hold once the number of trials rises above 30, yielding an approximately normal data distribution [30]. A final set of simulations compares the SPS with con-resistant trust to Fadul's reputation-based trust[6]. The data collected during each run is the minimum system frequency.

VI. Results and Analysis

Simulation results support the use of an SPS with con-resistant trust over a traditional SPS. Fig. 3(a) and (b) shows original SPS frequency levels versus an SPS with con-resistant trust, respectively. Fig. 3(b) shows that the SPS with con-resistant trust can successfully keep the system's steady-state frequency above 58.8 Hz whereas the original SPS without a trust implementation in Fig. 3(a) does not. These plots show the frequency at generator 110 from time 0 to 50 s. Real protection systems would take equipment offline near 58.8 Hz. These plots show the frequency at steady state.

The remaining experiments will use the final stable frequency at time 50 s without intermediate dynamics. The results in Fig. 3(a) and (b) map closely to the average performance across many runs in Fig. 10. Fig. 3(a) corresponds to Fig. 10 when using an SPS without trust mechanism with 15 untrustworthy nodes. Fig. 3(b) corresponds to Fig. 10 when an SPS with con-resistant trust is used in this scenario.

Figs. 4–6 represent the mean con-resistant interaction trust (I-Trust) values as determined by the SPS con-resistant trust mechanism during 36 simulation runs for each of the three treatment levels. At time 0.180 s, Generator 93 is commanded to trip. At time 0.184 s, Generator 93 goes offline. At time 0.192 s, the SPS makes the determination of which load agent nodes are trusted and untrusted. The error bars represent a 95% confidence interval.

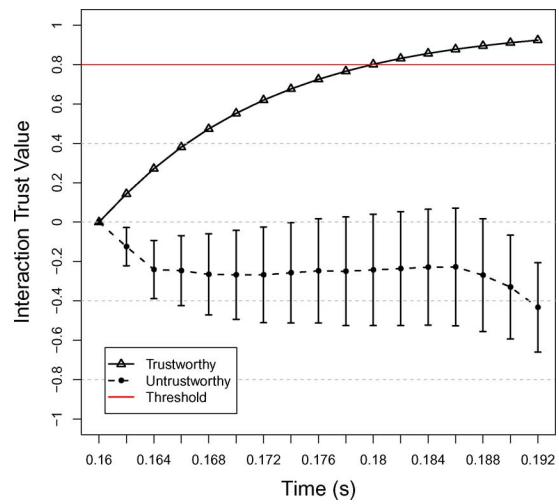

Fig. 4. Mean con-resistant trust results in five untrustworthy nodes.

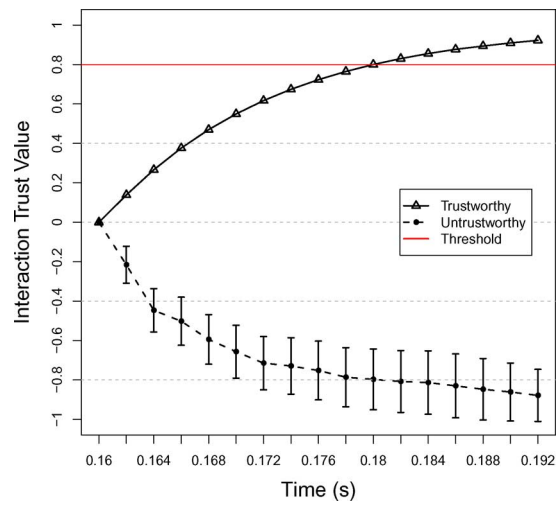

Fig. 5. Mean con-resistant trust results in 10 untrustworthy nodes.

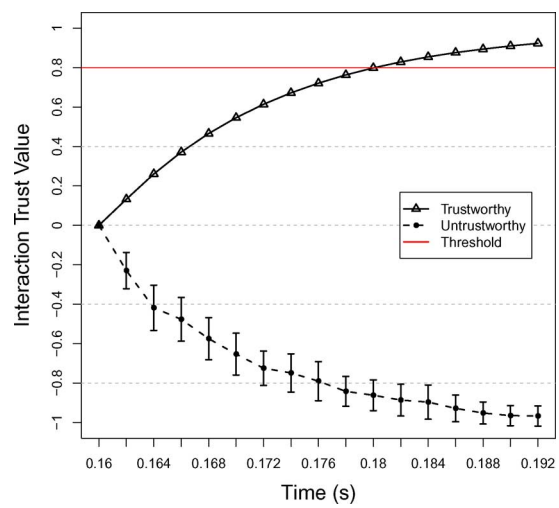

Fig. 6. Mean con-resistant trust results in 15 untrustworthy nodes.

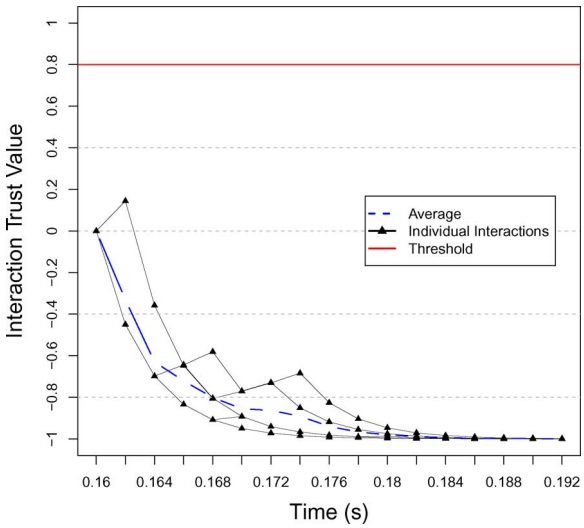

Fig. 7. Individual cooperative and defective interactions for five untrustworthy nodes during one simulation run.

The SPS with con-resistant trust is tuned to minimize identifying unreliable nodes as trusted. Empirical data showed that the mean error associated with a 95% confidence interval for the trustworthy nodes is negligible (<0.0027). The system is capable of identifying nodes that exhibit cooperative behaviors with a high degree of certainty, but it is possible for the trust mechanism to classify a node with cooperative behavior as *untrusted* for a short interval, if its frequency reading deviates significantly from the mean frequency of all nodes. A significant frequency deviation is one that is more than 3.6% outside one standard deviation from the mean reported frequency for all nodes, which would make the untrusted node classification error greater than that for the trusted node.

Fig. 4 shows that the error associated with a 95% confidence interval for five untrustworthy nodes increases during the simulation. The larger error signifies false negatives, that is, untrustworthy nodes reported as trusted. However, at time 0.192 s, when the final trust determination is made, the high and low interaction trust (I-Trust) values representing a 95% confidence interval for the experiment are -0.20614 and -0.66014, respectively. These values fall well below the trust threshold and would not be selected for load shedding.

As the number of untrustworthy nodes increases, the number of false negatives decreases. This is evident in the ten and fifteen untrustworthy node experiments in Figs. 5 and 6, respectively. The high and low I-Trust values representing a 95% confidence interval for ten untrustworthy nodes at time 0.192 s are -0.74631 and -1.01031. Similarly, the high and low I-Trust values representing 95% confidence intervals for 15 untrustworthy nodes at time 0.192 s are -0.916 and -1.018. In both cases, the I-Trust values fall well below the threshold to be selected for load shedding. By comparison, the false positive rate is nearly zero in all cases, as evidenced by tight error bars around the trustworthy nodes.

Fig. 7 depicts the individual cooperative and defective interactions of five untrustworthy nodes as determined by the SPS con-resistant trust mechanism during one simulation run. The

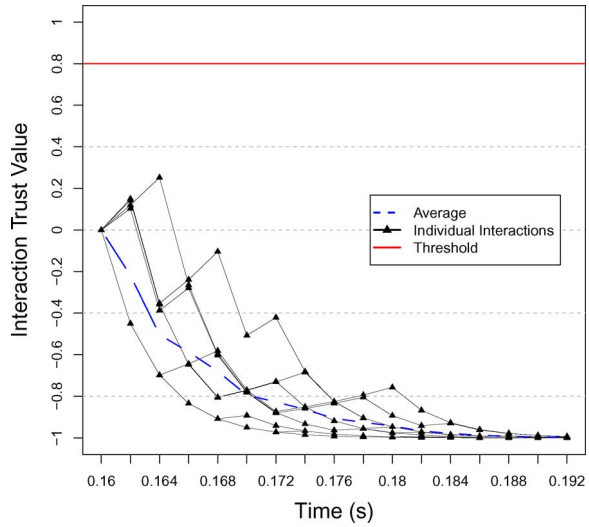

Fig. 8. Individual cooperative and defective interactions for 10 untrustworthy nodes during one simulation run.

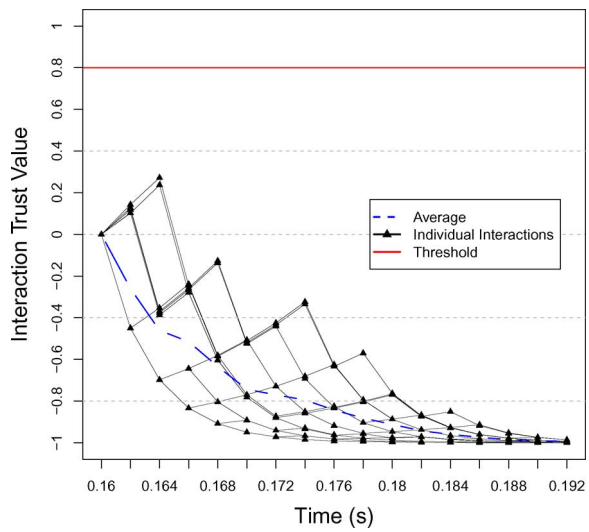

Fig. 9. Individual cooperative and defective interactions for 15 untrustworthy nodes during one simulation run.

untrustworthy nodes' interaction trust (I-Trust) values are severely impacted by their defection activity. Four of the five untrustworthy nodes exhibited cooperative behaviors, but none of their interaction patterns converged to high I-Trust values. For an untrustworthy node to be trusted, it would take a significant number of cooperations and considerable time.

Fig. 8 depicts the individual cooperative and defective interactions of ten untrustworthy nodes as determined by the SPS con-resistant trust mechanism during one simulation run. As with the five untrustworthy node interactions case, the ten untrustworthy nodes' interaction trust (I-Trust) values are severely impacted by the defection activity. Ninety percent of the untrustworthy nodes exhibit cooperative behaviors. As in the five individual untrusted node interactions, none of the ten untrustworthy nodes converge to a high I-Trust value. All of the untrustworthy nodes converge to a -1.0 I-Trust value.

Fig. 9 depicts the individual cooperative and defective interactions of 15 untrustworthy nodes as determined by the SPS con-resistant trust mechanism during one simulation run. As

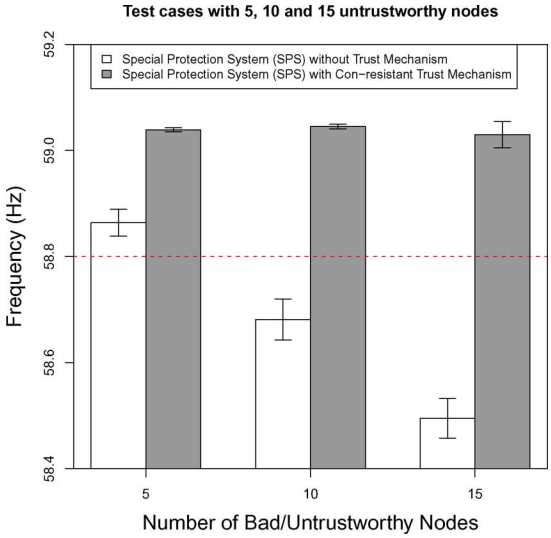

Fig. 10. Comparison of tests with 5, 10, and 15 untrustworthy nodes.

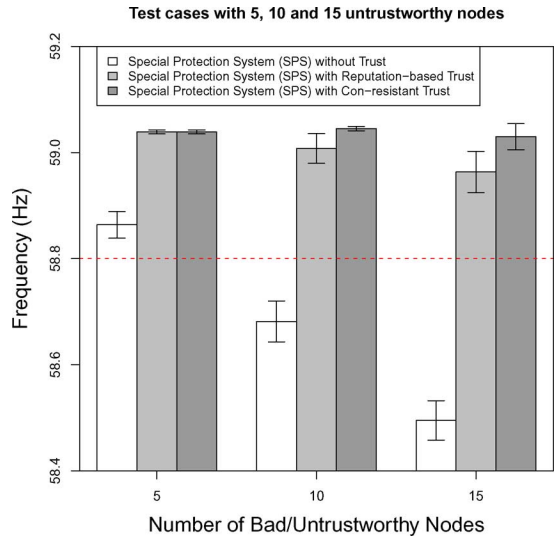

Fig. 11. Comparison of test data against prior reputation-based work [6].

TABLE III
ANOVA NUMERICAL CALCULATION RESULTS BETWEEN SPS WITH NO TRUST AND SPS WITH CON-RESISTANT TRUST

```
Analysis of Variance Table

Response: Frequency
                Df Sum Sq Mean Sq  F value    Pr(>F)
Treatment        1 6.9154  6.9154 1069.523 < 2.2e-16 ***
Levels           2 1.2876  0.6438   99.572 < 2.2e-16 ***
Treatment:Levels 2 1.1653  0.5827   90.114 < 2.2e-16 ***
Residuals      210 1.3578  0.0065
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

TABLE IV
ANOVA NUMERICAL CALCULATION RESULTS BETWEEN SPS WITH REPUTATION-BASED TRUST AND SPS WITH CON-RESISTANT TRUST

```
Analysis of Variance Table

Response: Frequency
                Df  Sum Sq  Mean Sq F value    Pr(>F)
Treatment        1 0.06463 0.064630 13.9985 0.0002361 ***
Levels           2 0.06820 0.034099  7.3856 0.0007948 ***
Treatment:Levels 2 0.03993 0.019964  4.3240 0.0144454 *
Residuals      210 0.96955 0.004617
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

with the five and ten untrustworthy node cases, the 15 untrustworthy nodes' interaction trust (I-Trust) values are also severely impacted by the defection activity. Of all the untrustworthy nodes, 93.3% exhibit cooperative behaviors. Just as in the 5 and 10 untrustworthy node cases, none of the 15 untrustworthy nodes converge to a high I-Trust value. All of the untrustworthy nodes converge to a -1.0 I-Trust value.

Fig. 10 compares experiments at each treatment level. Each bar plot represents the mean steady-state frequency reported at the end of each simulation run. Error bars represent a 95% confidence interval. The nonoverlapping confidence intervals show a statistically significant difference between the original SPS without the trust mechanism and the SPS with con-resistant trust. This indicates that the SPS with con-resistant trust performed better than the one without a trust mechanism and that it is extremely unlikely that this occurred by chance.

ANOVA calculations in Table III indicate a significant statistical difference between the two factors (with and without the trust mechanism). The p-value, $\Pr(>\mathrm{F})$ is less than $2.2x10^{-16}$, which is smaller than an alpha value of 0.05 associated with a 95% confidence interval. The small p-value is convincing evidence of a statistical difference between the two factors.

Fig. 11 compares the trust implementations conducted at each of the treatment levels, including the original SPS without any trust implementation, Fadul's [6] SPS with majority-rules reputation-based trust, and the con-resistant trust SPS introduced in this paper. Each bar plot represents the mean steady-state frequency reported at the end of each simulation run. The error bars represent a 95% confidence interval. A visual analysis shows that the SPS with either reputation-based or con-resistant trust is able to successfully keep the steady-state frequency above 58.8 Hz across all three treatment levels.

The ANOVA analysis in Table IV indicates a statistical difference between the SPS with con-resistant trust and Fadul's SPS with reputation-based trust [6]. The p-value $\Pr(>\mathrm{F})$ is approximately 0.0002, which is smaller than an alpha value of 0.05 associated with a 95% confidence interval. The small p-value is convincing evidence of a statistical difference between the SPSs with con-resistant and reputation-based trust.

A pairwise t-test indicates that a significant statistical difference between the reputation-based trust and con-resistant trust enabled 5 and 15 untrustworthy node results with a p-value of 0.0004 and in the 10 and 15 untrustworthy node cases with a p-value of 0.0123. These p-values are smaller than an alpha value of 0.05 associated with a 95% confidence interval which is convincing evidence of a statistical difference between the 5 and 15 untrustworthy node cases and the 10 and 15 untrustworthy node cases. The pairwise t-test results indicated no statistical difference between 5 and 10 untrustworthy node cases with a p-value being 0.29361. To summarize, the SPS with con-resistant trust is better than the alternative with reputation-based trust. The difference becomes more significant as the number of bad/untrusted nodes increases.

These experiments demonstrate that the SPS with the proposed con-resistant trust mechanism outperformed the alternatives tested when faulty or malicious nodes were present.

VII. Conclusion

This paper presented an SPS with a con-resistant trust mechanism, which can function in the presence of untrustworthy (malicious or malfunctioning) protection nodes using load shedding to mitigate transient instabilities. Success was determined by the system's ability to accurately identify which load agent nodes are trustworthy and its ability to select a sufficient number of trustworthy nodes to shed load to keep the steady-state frequency above 58.8 Hz. The results showed that the SPS with a con-resistant trust mechanism was able to keep the steady-state frequency above the 58.8-Hz threshold. The SPS with a con-resistant trust mechanism also successfully identified nodes that exhibit cooperative behaviors as trusted and defective behaviors as untrusted with a high degree of certainty. An analysis of the conducted experiments suggests that the benefits of the con-resistant trust mechanism are statistically significant.

Acknowledgment

The views expressed in this document are those of the authors and do not reflect the official policy or position of the U.S. Air Force, Department of Defense, or the U.S. Government.

References

[1] T. Flick and J. Morehouse, *Securing the Smart Grid: Next Generation Power Grid Security*. Burlington, MA, USA: Syngress, 2010.
[2] What is the Smart Grid? U.S. Department of Energy. Washington, DC, USA, Apr. 2012. [Online]. Available: http://www.smartgrid.gov/the_smart_grid#smart_grid
[3] V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. Hancke, "Smart grid technologies: Communications technologies and standards," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.
[4] M. Brandle and M. Nadele, "Security for process control systems: An overview," *IEEE Security Privacy*, vol. 6, no. 6, pp. 24–29, Nov./Dec. 2008.
[5] P. Anderson, *Power System Protection*. New York, USA: McGraw-Hill, 1999.
[6] J. E. Fadul, "Using reputation based trust to overcome malfunctions and malicious failures in electric power protection systems," Ph.D. dissertation, Dept. Elect. Comput. Eng., Air Force Institute of Technology, Dayton, OH, USA, 2011.
[7] P. M. Anderson and B. K. LeReverend, "Industry experience with special protection schemes," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1166–1179, Aug. 1996.
[8] P. Kundur, *Power System Stability and Control*. New York, USA: McGraw-Hill, 1994.
[9] I. Ray, I. Ray, and S. Chakraborty, "An interoperable context sensitive model of trust," *J. Intell. Inf. Syst.*, vol. 32, no. 1, pp. 75–104, 2009.
[10] S. I. Ahamed, M. M. Haque, and N. Talukder, "A formal context specific trust model (FTM) for multimedia and ubiquitous computing environment," *Telecommun. Syst.*, vol. 44, pp. 221–240, Aug. 2010.
[11] M. C. Duncan, "Trust management and security in satellite telecommand," M.Sc. degree, Air Force Institute of Technology, Dayton, OH, USA, 2011.
[12] W. Zhao and V. Varadharajan, "An approach to unified trust management framework," in *Collaborative Computer Security and Trust Management*. Hershey, PA, USA: IGI Global, 2009, pp. 111–130.
[13] B. Yu and M. Singh, "A social mechanism of reputation management in electronic communities," *Cooperative Inf. Agents IV-The Future of Inf. Agents Cyberspace*, pp. 355–393, 2000.
[14] A. Salehi-Abari and T. White, "Towards con-resistant trust models for distributed agent systems," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 272–277.
[15] W. Sherchan, S. Nepal, and C. Paris, "A suvey of trust in social networks," *ACM Comput. Surveys*, vol. 45, no. 4, Aug. 2013.
[16] T. Grandison and M. Sloman, "A suvey of trust in internet applications," *IEEE Commun. Surveys Tutorials*, vol. 3, no. 4, pp. 2–16, 2000.
[17] N. N. Bengiamin and W. C. Chan, "Variable structure control of electric power generation," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 2, pp. 375–380, Feb. 1982.
[18] J. Barsom and S. Rolfe, "Fracture and fatigue control in structures," in *Application of Fracture Mechanics*, 3rd ed. Philadelphia, PA, USA: Butterworth-Heinemann, 1999.
[19] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-Shelf components," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 548–558, May 2006.
[20] *"PSS/E 30 Users Manual,"* Siemens Energy, Schenectady, NY, USA, Jul. 2011. [Online]. Available: www.energy.siemens.com/us/en/services/power-transmission-distribution/power-technologies-international/software-solutions/pss-e.htm
[21] *"PSCAD/EMTDC Manual Getting Started,"* Manitoba HVDC Research Centre, Winnipeg, MB, Canada, 1998.
[22] *"The NS Manual, The VINT Project,"* K. Fall and K. Varadhan, Nov. 4, 2011.
[23] V. Vittal, "Transient stability test systems for direct stability methods," *IEEE Trans. Power Syst.*, vol. 7, no. 1, pp. 37–43, May 1992.
[24] P. Cote and M. Lacroix, "Benefits of special protection systems in competitive market," in *Proc. 22nd IEEE Power Eng. Soc. Int. Conf. Power Ind. Comput. Appl.*, 2001, pp. 192–195.
[25] L. A. Oquendo-Class, K. M. Hopkinson, X. Wang, T. R. Andel, and R. W. Thomas, "A robust communication-based special protection system," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1314–1324, Aug. 2010.
[26] H. Lokay and V. Burtnyk, "Application of underfrequency relays for automatic load shedding," *IEEE Trans. Power App. Syst.*, vol. PAS-87, no. 3, pp. 776–783, Mar. 1968.
[27] The R Project for Statistical Computing. R Foundation for Statistical Computing, 2012. [Online]. Available: http://www.r-project.org/
[28] R. I. Kabacoff, *R in Action*. Shelter Island, NY, USA: Manning Publications, 2011.
[29] *"NIST/SEMATECH e-Handbook of Statistical Methods,"* Apr. 2012. [Online]. Available: http://www.itl.nist.gov/div898/handbook/index.htm
[30] W. Mendenhall, R. Beaver, and B. Beaver, *Introduction to Probability and Statistics*, 14th ed. Boston, MA, USA: Cengage Learning, 2012.

Crystal M. Shipman received the M.S. degree in cyberoperations from Air Force Institute of Technology, Wright-Patterson AFB, OH, USA, in 2012.

Her interests include cybersecurity and the smart grid.

Kenneth M. Hopkinson (SM'10), is an Associate Professor at Air Force Institute of Technology, Wright-Patterson AFB, OH, USA.

His interests include networks, protection, and security.

Juan Lopez, Jr. is a Research Scientist at the Air Force Institute of Technology's Center for Cyberspace Research, Wright-Patterson AFB, OH, USA.

His interests include network security and supervisory-control-and-data-acquisition systems.