Consensus based Anomaly detection in Smart Grid

Fault-Tolerant Computer System Design
Final Project Report

Amit Sheoran
Vishwanath Singh

Contents

Introduction

Smart grid has emerged as next generation power grid via the convergence of traditional power grid infrastructure with communication technology which enables the transfer of real time information between the power grid and the end users via smart meters. In this article we describe the general architecture and goals of the smart grid. Following a brief description of the types of security attacks that can be manifested against a smart grid system and approaches that can be used to detect intrusion in the grid infrastructure, we introduce a consensus based algorithm to detect timing attack in Smart grid systems.

Problem motivation

Traditional power grid systems are unable to handle the complex electricity usage patterns and multiple sources of power generation. The centralized management used by the power grids is undoubtedly outdated and is unable to handle the demand for continuous and stable power distribution. A centralized grid management involves massive data exchange which causes huge message latencies and therefore can not satisfy the real time monitoring and control requirement required to handle the renewable energy sources and intelligent customer equipment. With increased focus on utilization of renewable energy resources there is a need for more effective communication capabilities between the grid and end devices that are capable of altering the power usage to suit the power generation patterns. Figure 1 presents a general communication model used by the smart grid systems. As shown in Figure 1 the grid relies on the smart meters at the points to relay information between the consumer and the grid management. One of the major challenges in such massive communication infrastructure is grid security. Since the end nodes are constrained devices which are limited in memory and processing capacities, its not possible to use traditional Intrusion detection mechanism to locate the security breaches in the smart grid networks. In this paper we analyze a specific type of attack called the timing attack and a consensus based distributed algorithm to detect such attacks.

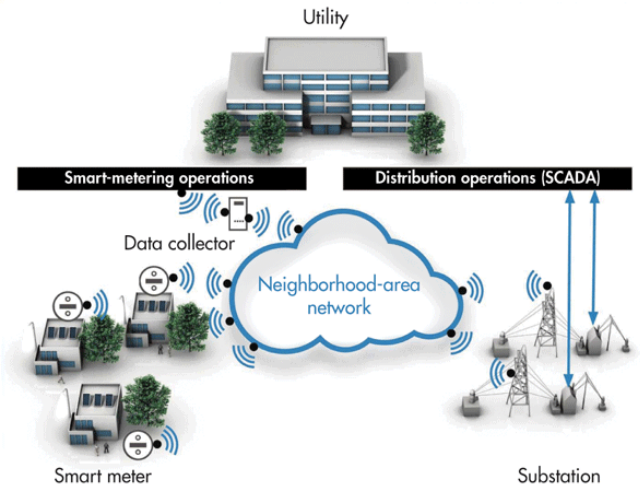


Figure 1 : A general architecture of smart grid systems

Cyber Attacks on Smart grids

Smart grids like any other distributed communication networks are susceptible to various types of cyber attacks [1]. Table 1 presents a basic classification of attacks against the smart grid systems. There are 4 types of attack in this taxonomy: device attack, data attack, privacy attack and network availability attack.

Device attack involves compromising one of the devices in the network. This is usually a first step towards a larger attack. The attacker uses the compromised node to induce malicious messages in the network. The compromised node may suddenly increase the load to cause circuit overload

Table 1 : Taxonomy of cyber attacks on smart grid

Name	Description
Device Attack	Aims to compromise a grid device (Initial step of a sophisticated attack)
Data Attack (False Data Injection)	Insert , Alter or delete network traffic to mislead the smart grid to take wrong decisions
Privacy Attack	Aims to learn users private information by analyzing the usage data
Network Availability Attack	Overwhelm the communication and computational resources of smart grid and to result in delay or failure of communication

A Data attack relies on inserting, deleting or altering the rating messages related commands to disrupt power generation to utilization ratio. Examples of these messages include rating control messages informing the end nodes of a lowering in the power rate or injecting messages to inform the grid of the increased power usage by the end nodes. The compromised node can also be used to disrupt the communication of other nodes which use the compromised node to send and receive messages from the grid. These messages can be used to mislead the smart grid into taking wrong actions like increasing the power generation.

Privacy attacks aim at learning the private information of the user by analyzing the power utilization patterns. Data of current electricity utilization is collected and transmitted several times by the smart meters to keep the power grid updated on the latest power utilization. Although these attacks do not pose a threat to the power infrastructure, such information can be used to infer the occupancy status of the house. A lack of power utilization for equipment's like microwave, stove or heaters can indicate that house is not occupied and this information can be used for physical attacks on the house.

Network availability attacks aim to disrupt the communication capabilities of the network by overwhelming the radio or computational resources of network elements. These attacks may involve flooding the network with spurious packets to introduce delay in the message forwarding and processing capabilities of the individual nodes.

A newer category of attack involves inducing strategic delays in the messages which are transmitted between the grid elements. Unlike network availability attacks this category of attack does not disrupt the overall communication capabilities of the network. Instead it relies on introducing delays in specific types of messages thereby making it harder to detect than a simple flooding attack [2].

It has been demonstrated by experiments that such delays can cause irreparable attacks on the power generation infrastructure. Figure 2 shows the impact of a two second communication delay in the power reporting on the performance of the Automatic Generation Control [3].

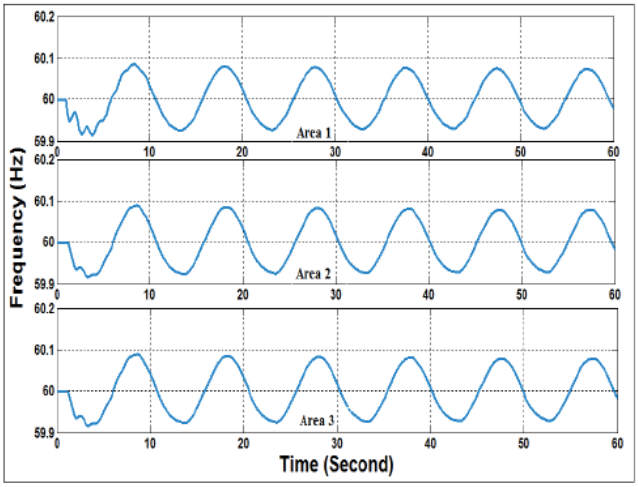


Figure 2 : Change in the operating frequency with a 2 seconds delay

Having demonstrated the impact of the timing attacks on the smart grid networks, we focus on the detection and avoidance of these issues.

In general, modern communication networks use encryption to secure the messages transmitted between communicating hosts and a cursory observation will reveal that a simple solution to the false data injection attacks like time delay attack can be usage of encryption to secure the messages between the grid and the end nodes. However, research on the messages analysis of smart grid traffic has revealed that encryption alone can not be used a mechanism to counter such attacks. Results from [4] reveal that the attacker does not need accurate system information to be successful and could affect monitoring accuracy by up to 20% even in the presence of encrypted traffic.

Table 2 : Taxonomy AMI traffic

Message Types	Frequency	Encryption	Packet Size
Periodic and continuous traffic (heart beats)	60 seconds	No	92
Periodic AMI Request / Response / Registrations	Reactive	Yes	254
Aperiodic Traffic	NA	No	NA

A taxonomy of the Advanced metering traffic reveal that the communication between the grid elements exhibits a high degree of correlation [5]. Table 2 summarizes the frequency and type of the communication over 2 days. Such high degree of correlation in the AMI traffic makes them vulnerable to attacks despite

the message encryption. A more comprehensive solution to securing the smart grids of the future will be to use Intrusion detection systems (IDS) alongside the encryption communication.

IDS for Smart grid systems using Encrypted communication

As smart grids evolve to encompasses more critical functionality and migrate towards a standard communication stack, encryption becomes an essential tool for preserving the confidentiality of the network communications. Increased use of encryption is expected to prevent eavesdropping and better protect sensitive and private data. However, large scale deployment of encryption will entail a fundamental change in the operation of the IDS deployed for smart grids. Future IDSes must be capable of performing the packet inspection on encrypted traffic. Although it is possible to configure the IDS with encryption keys, to facilitate the real time analysis of the traffic, it would be helpful if the IDSes could use newer techniques to infer the traffic patterns from encrypted traffic.

A variety of techniques can be used by the IDSes [5] to infer traffic flow information from the encrypted traffic. Some of the techniques that can be used by IDS are

- Monitoring the periodicity of the network communication
- Using passive fingerprinting to detect rouge devices or known malware
- Identifying the traffic patterns and anomalies using clustering
- Tracking unknown flows by utilizing the network connectivity graphs.

As shown in previous section the messages send by the smart meters to the power grid exhibit clear patterns in the message frequency. IDSes can use the patterns in the AMI messages to detect anomalies in the traffic. It can be seen from the traffic analysis that AMI registration messages are generated by the Smart meters within a specific range, a sudden increase in the AMI registration messages could indicate a flooding attack on the grid infrastructure.

Fingerprinting is a mechanism by which the IDSes could detect the presence of unknown devices in the network by analysing the message headers. Although encryption hides the data content from the IDSes, the message headers are not encrypted and this information can be used to create a map of valid communication tuples in the network. A drastic change in the message flows that deviate from the maps stored in the IDSes could reveal an attack on the smart grid.

Clustering relies on the utilizing the communication behaviour (packet size, timing and header information) of the participating nodes to develop a traffic model which can be used as a template to detect outliers in the network nodes.

The concept of identifying intrusions using a network connectivity graph refers to learning the relationships among end points in order to flag when a new end point or a new link appears in the graph.

Figure 3 shows a mapping of the techniques described above and the types of malicious behaviour that they can detect.

Attacks	*Period.*	*Fingerpr.*	*Graph*	*Clustering*
Traffic tampering	✓			✓
Traffic injection	✓		✓	✓
Replay attack	✓			✓
Authentication abuse				✓
Spoofing		✓	✓	
Rogue device		✓	✓	
Compromised meter	✓		✓	✓
Resource exhaustion	✓		✓	✓

Figure 3 : Mapping between the IDS techniques and effectiveness despite encryption

Problem formulation

Based upon the analysis in the above sections its clear that the methods used by IDS's are based upon analysis of traffic patterns and will be deployed at a central point in the network. Also, the techniques mentioned above can not protect the malicious commands from being executed on the individual nodes until the grid detects an intrusion. If a distributed algorithm can be developed which can detect in real time the presence of time delay or false data injection attacks, it will have the following benefits

a) Individual smart meters can omit/delay the execution of commands(messages) which are suspected as malicious.
b) Smart meters can inform the grid of messages which are suspected as malicious and the grid IDS can use this information to detect the compromised node(s).

Problem statement

Can a consensus based algorithm be used to detect the presence of malicious/ delayed messages in a smart grid network? If yes, can this information be used to locate the malicious/ compromised node in the network?

Solution approach

Proposed Solution for Time Delay Attacks

Having analyzed the general techniques used by the Smart grid IDS, in this section we present a consensus based algorithm that can be used by a distributed IDS to detect the time delay and false data injection attacks. This algorithm does not address the entire class of cyber attacks that can be mounted against the smart grid infrastructure, instead it focuses on the detection and prevention of time delay and false data injection/omission attacks.

To illustrate the algorithm in detail, consider Figure 4.

Figure 4 has two sample clusters with 5 nodes each:

Cluster 1: (G, H, I, J and N)

Cluster 2: (H, J, K, L and M)

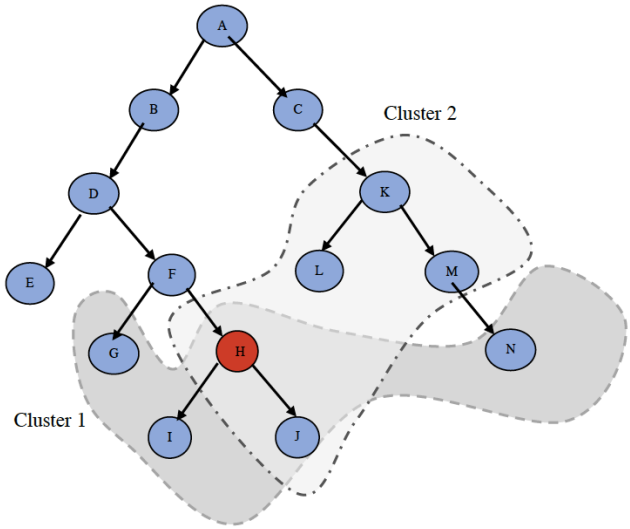


Figure 4: Sample Topology

If a given node is compromised with a timing attack, it will delay the delivery of the rating control messages to its child nodes. In Figure 4 the Node H is compromised and we assume that messages that are delivered to nodes I and J will be delayed by a random value.

The proposed algorithm works as follows

1. The network will be clustered at bootstrapping and each node in the network will be configured with a list of other nodes in its cluster.
2. All the nodes in the network will maintain a vector of N previous rating messages (Synchronization Vector). **The messages are executed only after its confirmed that other nodes in the cluster have received a similar message**. Each node also maintains a list of the trust weights for other nodes in the cluster (Trust Vector) which is used to detect the anomalies.
3. The consensus algorithm used (RAFT [6]) will detect anomalies at the nodes I and J in cluster 1 and cluster 2 respectively. These anomalies will be reported by the cluster leaders to the (Data concentrator unit) DCU and an action to avoid the attack can be triggered by the DCU.

Terminology

1. **Cluster**: Cluster refers to a set of nodes grouped together for message synchronization. All nodes in the cluster are configured with the list of other nodes in its cluster during initialization of each node. Periodically, the nodes elect a cluster leader which is responsible for anomaly reporting.
2. **Synchronization vector**: Refers to a list of messages that are stored by each node which are used to synchronize among other nodes in its cluster.
3. **Trust vector**: A list of weights maintained by each node for other members in its cluster. These values are used during the selection of a leader.
4. **Trust weight**: The factor by which the trust vector is modified in case of synchronization vector

mismatch. (The values of trust weight are constant in the current code)

5. **Trust Threshold**: Minimum trust vector value after which the leader reports a node as anomaly.

Solution Details:

Clustering: The network will be clustered into multiple (possibly) overlapping clusters. The solution will analyze multiple clustering schemes like random, (Breadth First Search) BFS and (Depth First Search) DFS in the selection of nodes in the cluster.

Consensus Algorithm : The consensus algorithm used is based upon RAFT. However, significant changes are made to the second phase of raft algorithm to suit the project requirements.

The updated consensus algorithm consists of following steps:

- Leader selection: The nodes in a cluster will periodically elect a leader. Each node will start a random timer, upon expiry of which the node nominates itself as the leader and sends a request seeking votes from other nodes in the cluster.

  A node can only become a leader when

  - It has a consistent message vector, that is all other nodes have a weight factor of 100% for the nominated leader.
  - A majority of nodes in the cluster vote for the nominated node.

  After an election round is completed, the elected node will send a leader election announcement to other nodes in cluster notifying them of the elected leader. This step will also conclude the election phase thereby restarting the election timer on all nodes. The randomization threshold is used to ensure that the same node is not elected leader every time. Figure 5 illustrates the steps involved.

- Anomaly detection: After any rating related message is received from the DCU, the receiving node will wait for a brief period and send a synchronization message to other nodes in the cluster. The sender node will supply the synchronization Vector in this message. Upon receipt of this synchronization message each node will process the message and validate the received vector with the local vector. If the received message list is consistent with the senders list a SYNC_SUCCESS (value = True) message is send to the sender. The source node will then update the trust vector in accordance with the response received.

- Anomaly Reporting: The cluster leader will report an anomaly to the DCU when the trust weight value for a given node is below the configured threshold. If the node is present in multiple clusters or multiple nodes share the parent which has been compromised, multiple cluster leaders will generate these anomaly reports. Figure 6 illustrates the process of anomaly detection.

Compromised node detection: After the anomaly reports have been generated by the cluster leaders, the DCU or any other designated node can find the compromised node by finding the least common parent for the anomaly nodes.

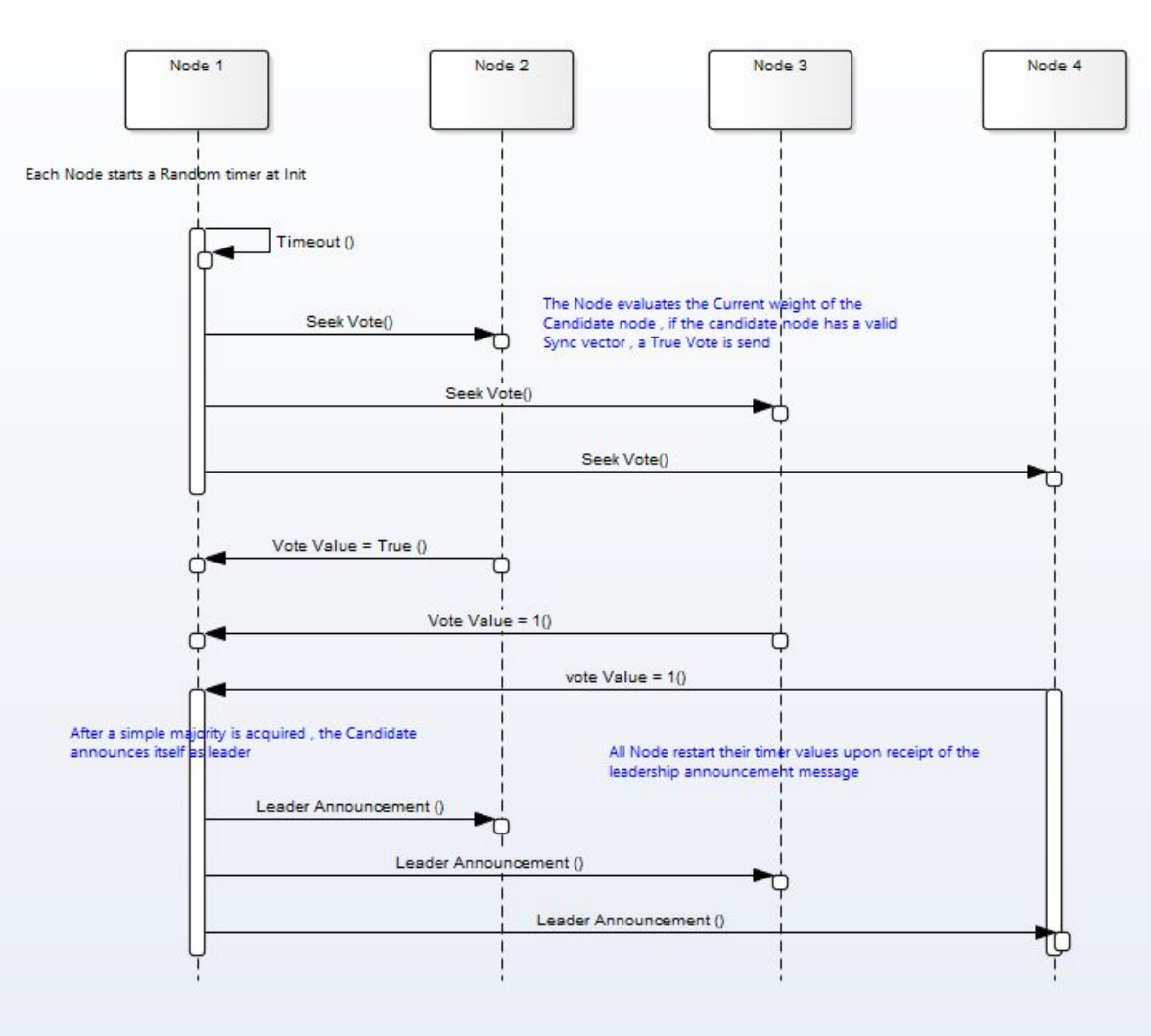


Figure 5 : Leader Selection

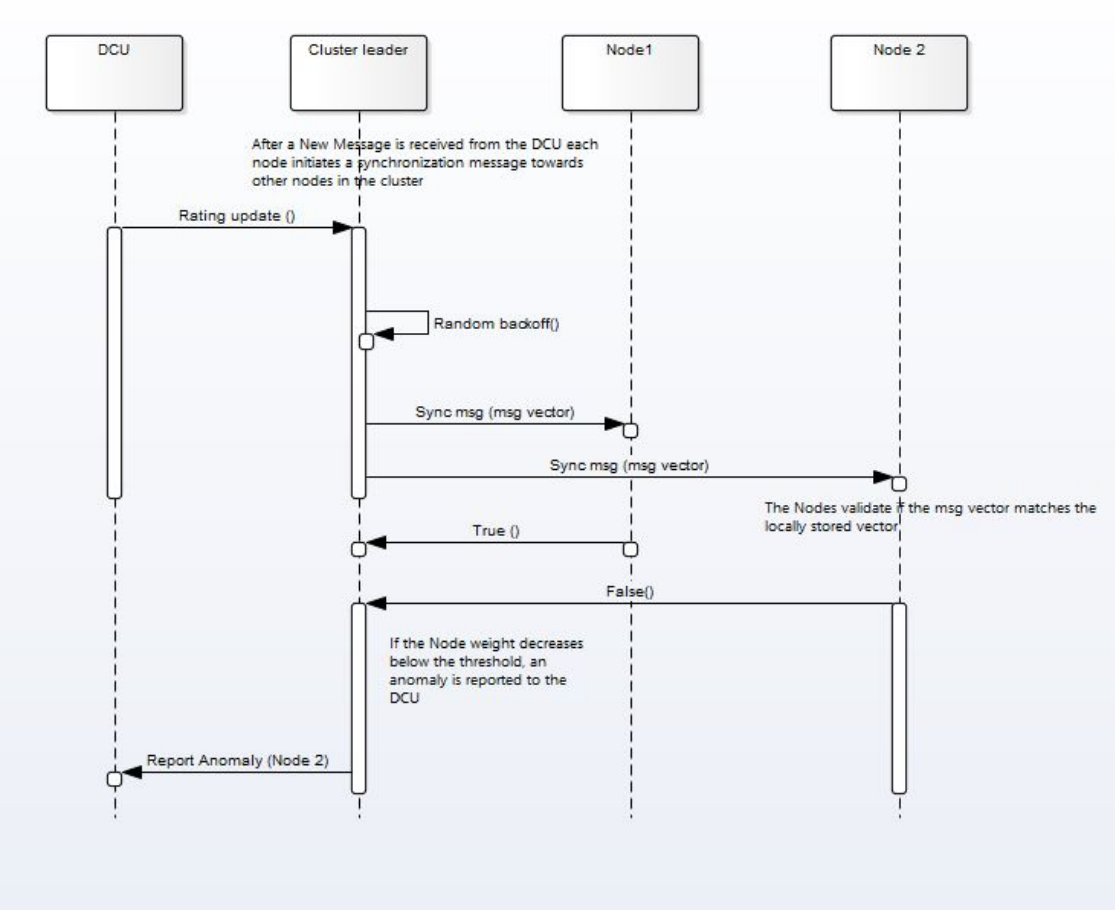


Figure 6 : Anomaly detection

Implementation

We extended Sec-AMI to implement the consensus based attack detection model. The simulation consists of the following components,

- **Topology Generator:** We used the matrix-topology generator module in the existing Sec-AMI model. This gives an adjacency list of the graph which models the Neighborhood Area Network. The node 0 in this graph is treated as a Distribution Control Unit (DCU) and this is the node which would trigger the message flow in the graph.
- **Cluster Generator:** The cluster generator module takes the graph topology as an input and groups the nodes into different clusters based on the clustering algorithm being tested. The currently implemented clustering algorithm are random clustering, Breadth First Search and Depth First Search. We put a soft limit on the number of nodes that can be added in a cluster. A node may be part of more than one cluster.
- **Attack Simulator:** This is an event based attack simulator model which is part of the existing Sec-AMI model. It provides an event loop for scheduling events in the network. We used this loop as the backbone for message transmission between neighbors.

- **Message Generator:** The message generator was integrated into the Sec-AMI event loop to model the communication between nodes starting from DCU. Each message represents a value which a station intends to broadcast into the system. On receiving a message, a node broadcasts it to all the other connected nodes. If the node is one of the attacked nodes, this message may be dropped or delayed with a certain probability.
- **Raft Process:** The raft model was implemented as an independent module. This module is responsible for leader election, propagating synchronization vectors and anomaly detection.

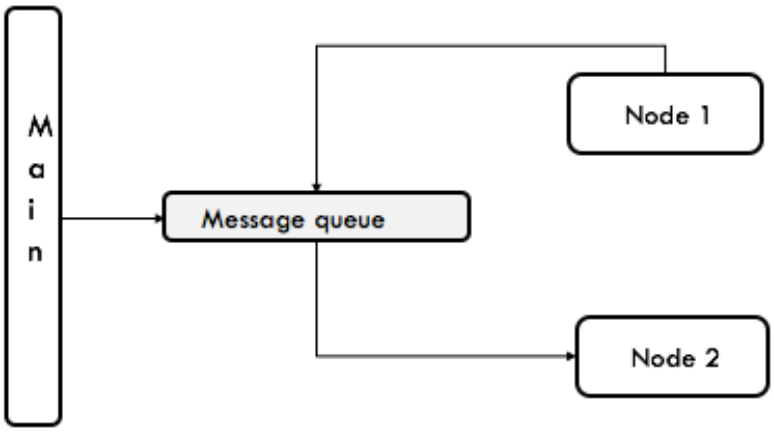


Figure 7 : Event Model for RAFT

Each Raft node is implemented as a thread and each thread is associated with a message queue. The messages are communicated to each node by posting a message to the respective node's queue.

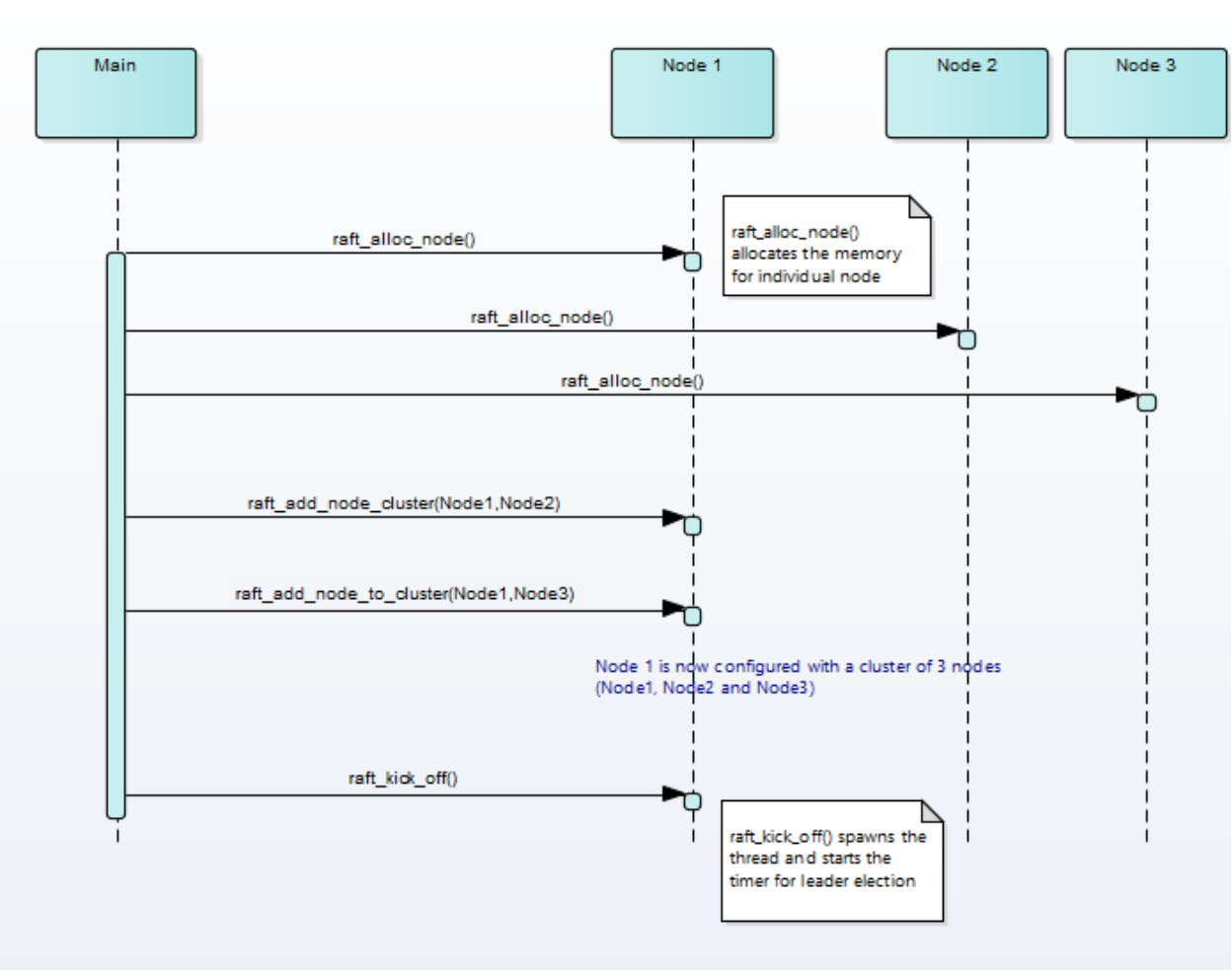


Figure 8: Initiation sequence for RAFT

The Message processing behaviour of the RAFT node is shown in Figure 9 and Figure 10.

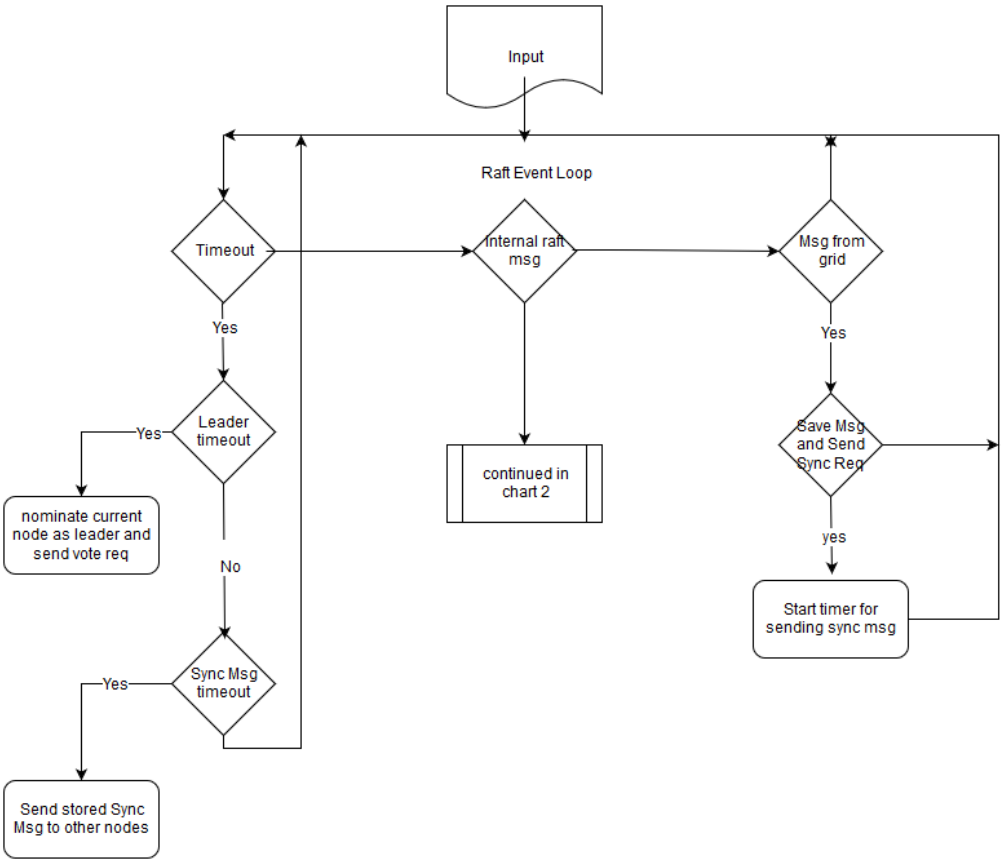


Figure 9: Event loop RAFT

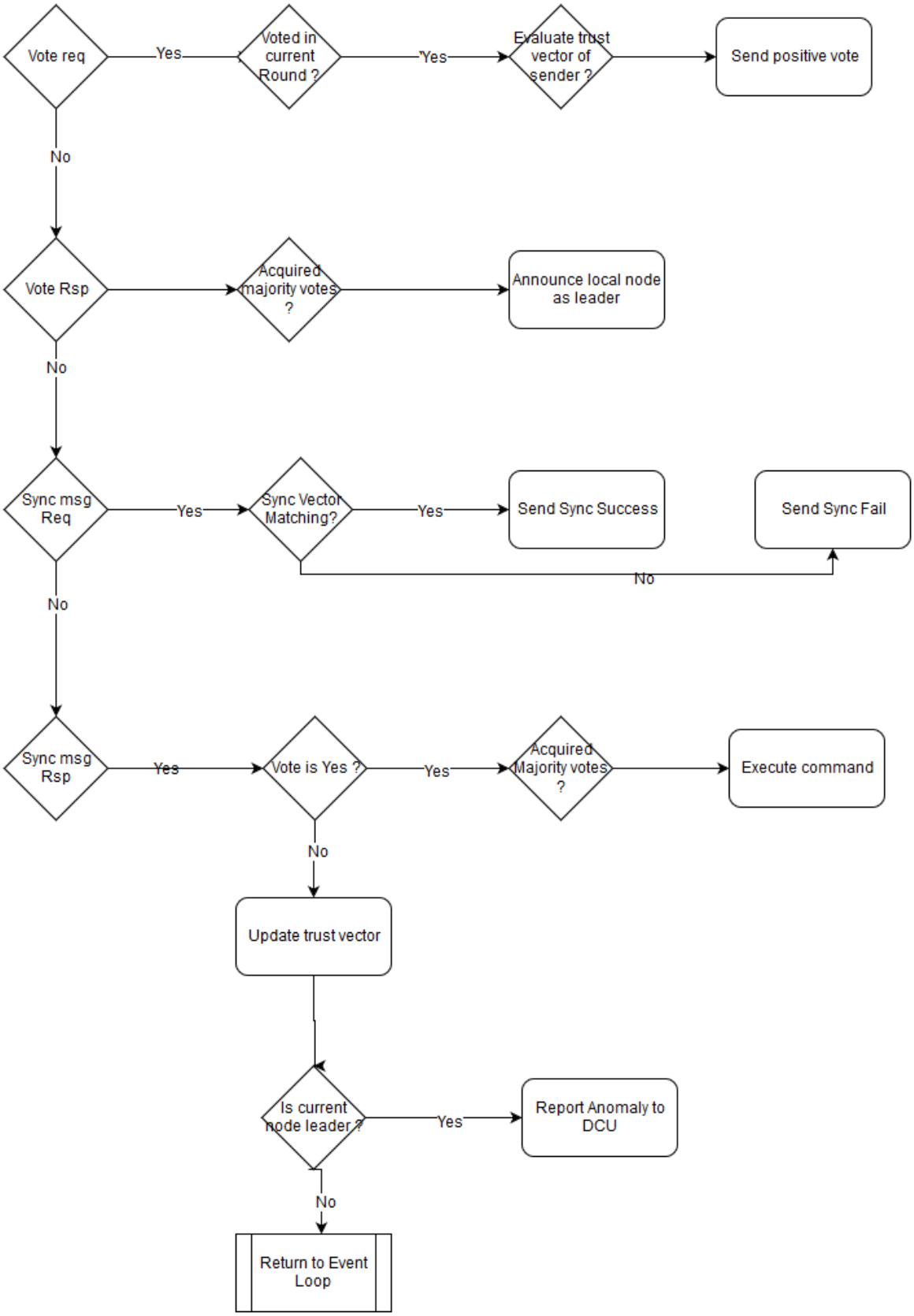


Figure 10 : RAFT Internal Message handling

Results and insights

We ran the experiments on a set of nodes and verified that the Raft module was successfully able to flag the anomalies. The leader election process also successfully rotated the leaders among the various nodes.

1. **Leader Rotation:** The results we observed by running the experiments showed that the leader was being correctly rotated among all the active nodes. Also it was validated that a node under attack was not able to procure the votes required and hence it was not elected as leader in any round.

2. **Anomaly Detection**: As seen in the figures below the leader correctly displays the anomaly in the graphs. This is then used to identify the lowest common ancestor of the nodes which are marked to have inconsistent data and is flagged by the algorithm.

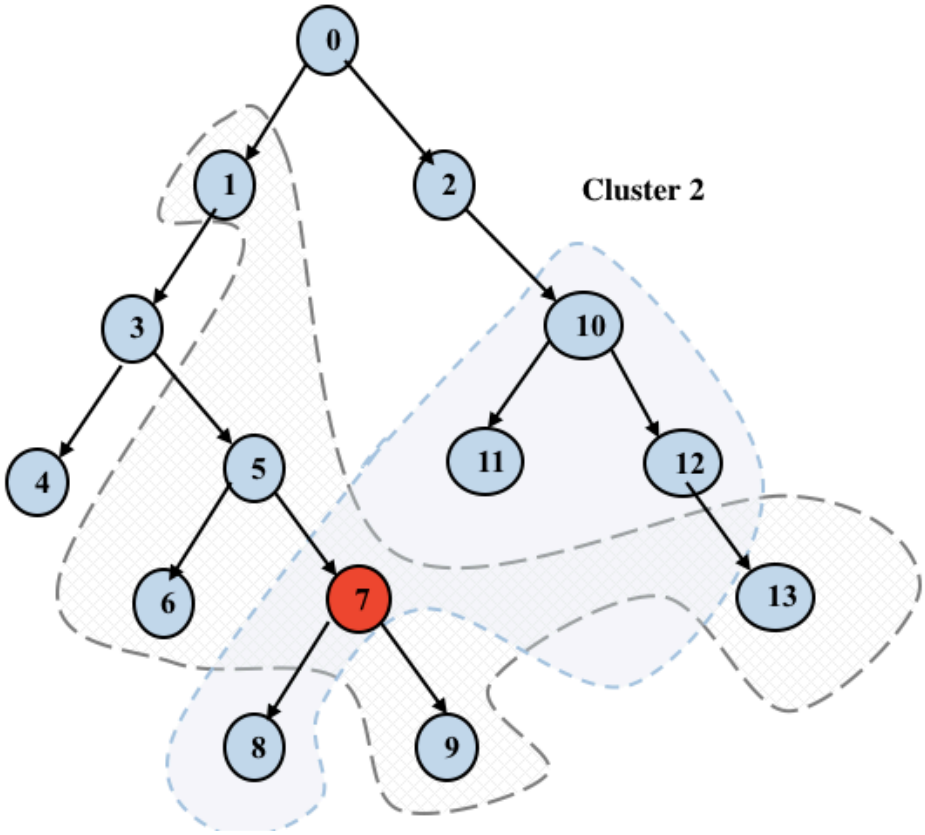
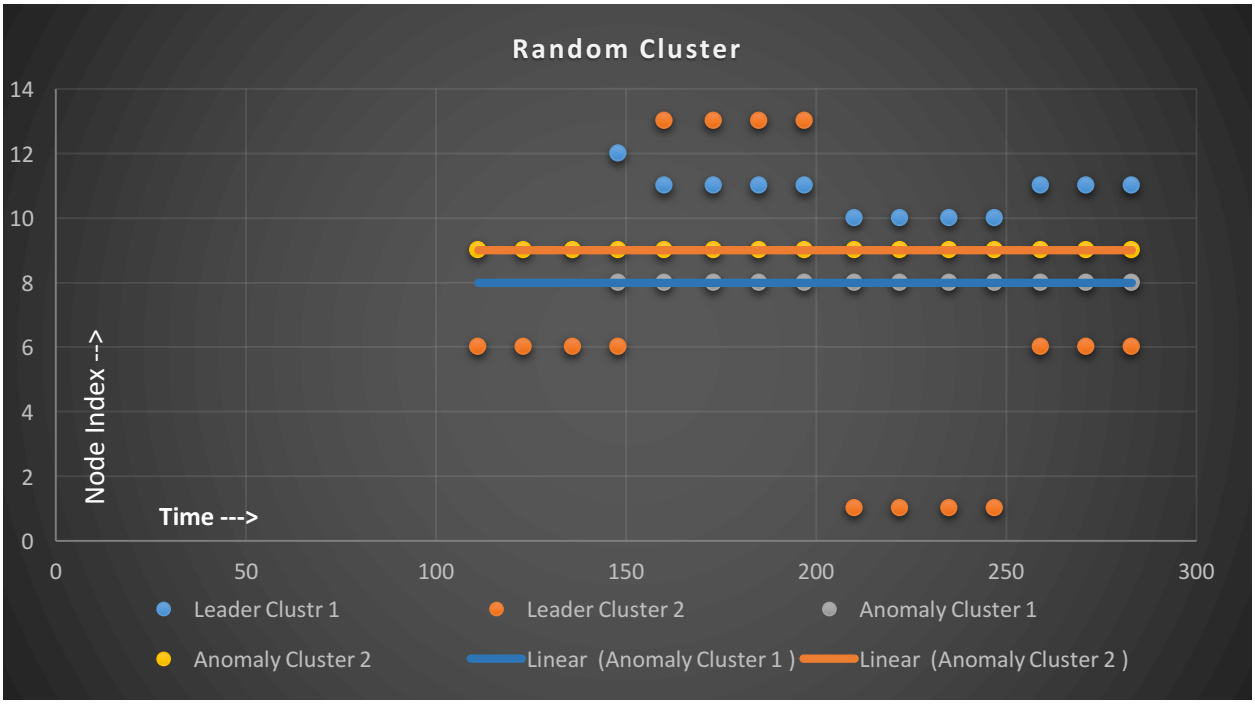
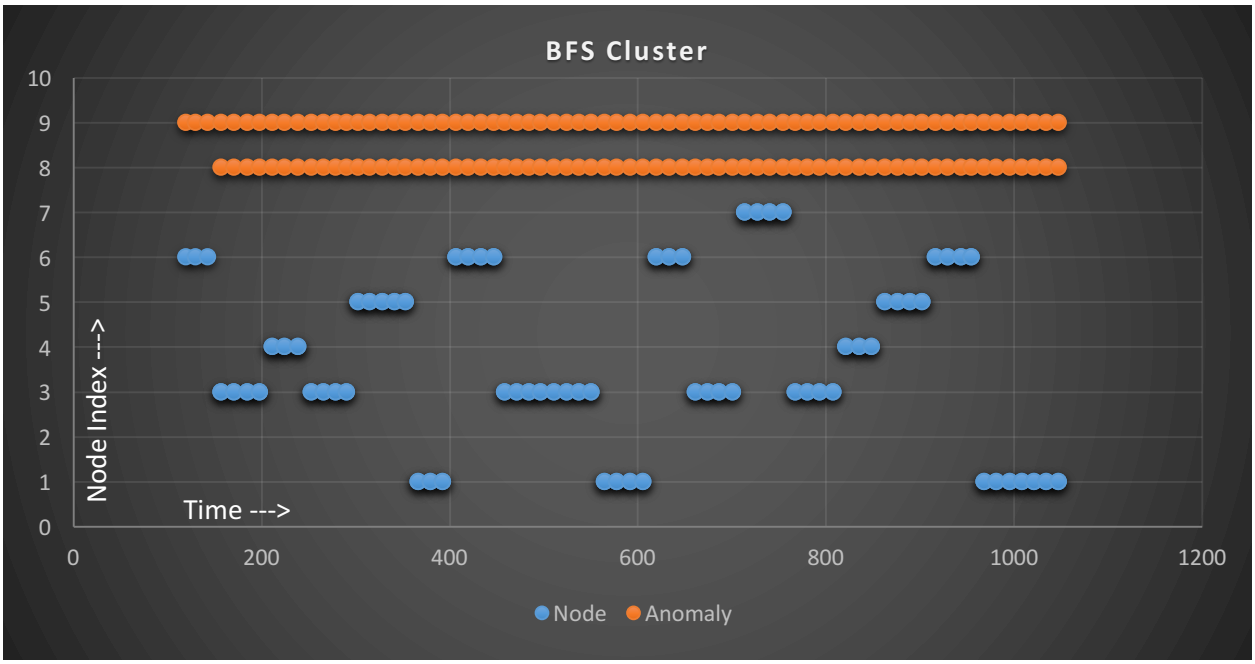


Figure 11 : Topology used in the Sample below



In the above graph, Node 8 and 9 are detected as anomaly by cluster 1 and 2 respectively.



In the graph above, the parent node 7 is compromised and 8 and 9 are grouped in the same group by the BFS algorithm and only a single cluster is generating anomalies.

We observed that false positives increase with higher cluster size and if a node at higher level is compromised.

Conclusion and future work

As the results indicate, we are successfully able to utilize a consensus based solution to detect anomalies due to time delay attacks in the smart grid networks. However, several issues need to be addressed in order to successfully integrate this solution to a deployable IDS.

Issues

1. Anomaly detection rate: detection of a compromised node depends upon the the arrival rate of messages, in case the message arrival rate is less – the algorithm should dynamically adapt the Trust weight and Trust threshold to reliably detect the anomalies.
   That is, if the message arrival rate is high - the value of Trust weight should be decreased and if message arrival rate is low, Trust weight should be increased so that the algorithm can detect the anomaly with fewer messages.

2. Detection of compromised node: compromised node detection depends upon the presence of a centralized node which stores the network topology. If no such node node exits the algorithm can not detect the the compromised. In the present implementation SecAMI does not store the information about the node parents. SecAMI code must be updated to execute the Compromised Node Detection step of the algorithm.

References

[1] Xu Li; Xiaohui Liang; Rongxing Lu; Xuemin Shen; Xiaodong Lin; Haojin Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges," in Communications Magazine, IEEE , vol.50, no.8, pp.38-45, August 2012

[2] Sargolzaei, A.; Yen, K.; Abdelghani, M., "Delayed inputs attack on load frequency control in smart grid," in Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES , vol., no., pp.1-5, 19-22 Feb. 2014 doi: 10.1109/ISGT.2014.6816508

[3] Rahimi, Kaveh; Parchure, Abhineet; Centeno, Virgilio; Broadwater, Robert, "Effect of communication Time-Delay attacks on the performance of Automatic Generation Control," in North American Power Symposium (NAPS), 2015 , vol., no., pp.1-6, 4-6 Oct. 2015

[4] Torrisi, N.M.; Vukovic, O.; Dan, G.; Hagdahl, S., "Peekaboo: A gray hole attack on encrypted SCADA communication using traffic analysis," in Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on , vol., no., pp.902-907, 3-6 Nov. 2014

[5] Berthier, R.; Urbina, D.I.; Cardenas, A.A.; Guerrero, M.; Herberg, U.; Jetcheva, J.G.; Mashima, D.; Huh, J.H.; Bobba, R.B., "On the practicality of detecting anomalies with encrypted traffic in AMI," in Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on , vol., no., pp.890-895, 3-6 Nov. 2014

[6] Diego Ongaro and John Ousterhout. 2014. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference* (USENIX ATC'14), Garth Gibson and Nickolai Zeldovich (Eds.). USENIX Association, Berkeley, CA, USA, 305-320.

[7] Shawly, T.; Jun Liu; Burow, N.; Bagchi, S.; Berthier, R.; Bobba, R.B., "A risk assessment tool for advanced metering infrastructures," in Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on , vol., no., pp.989-994, 3-6 Nov. 2014

[8] Steven R. Snapp, James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che lin Ho, Karl N. Levitt, Biswanath Mukherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal, and Doug Mansur. DIDS (distributed intrusion detection system) - motivation, architecture, and an early prototype. In Proceedings of the 14th National Computer Security Conference, pages 167–176, 1991.