SECURE INTEGRATION OF ELECTRIC VEHICLES WITH THE POWER GRID

BY

CHAITRA PRASAD NIDDODI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Dr. Sibin Mohan

ABSTRACT

A wide variety of distributed energy resources (DERs) such as pluggable electric vehicles (EVs), solar arrays, smart buildings, etc. are now being connected to the power grid. Malicious adversaries can use these as entry mechanisms to gain access to the grid with the intention of creating instability in the system. This work focuses on secure integration of DERs with the power grid. To this end, we propose techniques to detect malicious activity when either the DERs or the communication channels between the DERs and the smart grid components are compromised. We propose a *cyber-physical anomaly detection engine* to ensure that critical grid components remain secure, and hence, safe. Specifically, we have focused on the vehicle-to-grid (V2G) system. In this system, aggregators are the critical components through which DERs such as EVs are connected to the grid. We have developed a prototype anomaly detection engine for aggregators that manage/communicate with the EVs. Since the V2G system is time-sensitive, the anomaly detection engine also monitors the timing requirements of the system by checking the frequency constraints on messages at the aggregator apart from monitoring the cyber and physical data constraints to ensure safety of the aggregator.

ACKNOWLEDGEMENTS

The material in this paper is based upon work supported by the U.S. Department of Energy (DoE) under award numbered DOE UTRC Prime award DE-OE0000826[1].

I would like to thank my advisor Dr. Sibin Mohan for giving me an opportunity to be part of this project and for all his guidance along the way.

I would also like to thank Tim Yardley, Devu Manikantan Shila and Hao Zhu for their useful discussions and ideas.

[1] Disclaimer: Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

TABLE OF CONTENTS

Chapter 1: Introduction

The electric grid is one of the critical infrastructures of a nation and its correct and continued operation is of national interest. It is a complex cyber-physical system (CPS)[1], often with safety-critical properties. Any problem that affects the power grid can result in damage to life, property or the environment. With increased automation, technological and communication advances, multiple new components and systems are interacting with the grid, often in real-time. Some notable examples are the distributed energy resources (DERs) such as pluggable electric vehicles (EVs), solar arrays, smart homes and industrial building automation systems. With this increased connectivity and plethora of applications comes more opportunities for malicious entities to gain access to critical systems and potentially wreak havoc with the essential components of the grid. Attackers could use the communication channels between the DERs and the grid to actually take control of or even shut down critical grid components [11].

There exist multiple challenges in securing power grid systems: *(a)* the grid has many legacy systems that may not know how to interact, in a secure manner, with the newer applications such as EVs; *(b)* attacks on edge devices (such as DERs) are difficult to detect at the grid level due to the lack of visibility into their operational details; *(c)* new systems such as the V2G introduce new components such as EVs, aggregators and electric vehicle supply equipments (EVSEs) to the grid, each of which can have additional vulnerabilities; *(d)* these systems also introduce new communication paths that raise new issues dealing with coordination among multiple stakeholders and finally *(e)* new infrastructure must be set up for managing credentials for the new components

[1] Cyber-physical systems (CPS) are engineered systems that are built by integrating computation components with physical components. [1].

and their operators. In fact, it is well documented that the electric grid is vulnerable to a wide range of attacks [21].

With the increasing number of EVs (*e.g.,* Tesla), the communication complexities in the vehicle-to-grid (V2G) system have increased. EVs not only participate in drawing current from the grid to charge themselves but they also act as batteries that can discharge to the grid during periods of high power demand [24]. Hence, EVs are fast becoming an important type of DER that closely interact with the grid. If an attacker can take control of *(a)* either enough EVs in a local area or *(b)* the entity that controls/manages a large number of them then these systems can be used to cause problems to the grid. Hence, it is vital to ensure that EVs (and especially their communication channels with the grid) are monitored and remain secure. In this work we focus on EVs and their connection to the grid via intermediaries such as aggregators.

Figure 1.1 presents a high-level conceptual architecture of an electric grid with multiple EVs connected to it. The ”aggregators” [15] in this figure refer to entities that act as mediators between the end DER systems (the EVs in this case) and the power grid utility system. A single aggregator can manage multiple DERs (usually in close geographic proximity). In the model presented in Figure 1.1, the aggregator can be a prime target for attackers since *(i)* it manages multiple DERs and *(ii)* is also directly connected to the utility system. A successful intrusion at the aggregator level can have serious consequences for the end systems as well as the grid. Therefore, ensuring the security of this critical component (*viz.,*aggregator) is essential to ensure secure integration of DERs with the grid. To this end, we *propose an Anomaly Detection Engine with mechanisms to detect anomalous behavior in aggregators of the V2G system*. For our anomaly detection engine, we rely on both the cyber and physical properties of the system. On the cyber side, we focus on the communication protocol in the V2G system to ensure correct operation of the

controller, *i.e.,* aggregator, while we validate its behavior using the physical side of the system in the form of power measurements.

The main contributions of this work are:

1. An enumeration of the correct sequences of commands in the V2G communication protocol to generate an aggregator state machine.

2. Development of an anomaly detection engine that can differentiate between correct/incorrect communication and data exchange at the aggregator level. Since the V2G system is a real-time system (*i.e.,* a system with timing constraints), the anomaly detection engine also makes use of **timing constraints** of the V2G system apart from data constraints in order to differentiate between correct and incorrect behavior.

3. Implementation and evaluation of a prototype of the anomaly detection engine for aggregator-V2G system.

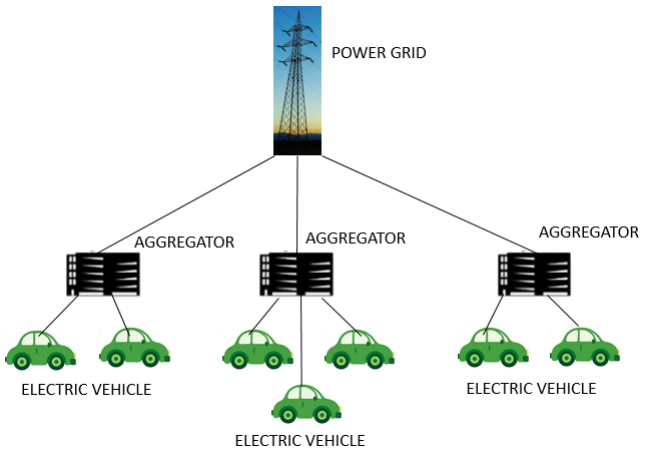

Figure 1.1: Conceptual Architecture of a Vehicle-to-Grid (V2G) system. [22]

Chapter 2: Vehicle-To-Grid System

Power grid systems such as Vehicle to Grid (V2G) system are *time-sensitive*. The correct functioning of the system requires not only functional correctness but also timing correctness. Specifically, the messages in the V2G system must meet certain **timing requirements such as frequency related constraints** apart from data constraints (explained in detail in Chapter).

Fig.1.1 shows the conceptual architecture of the V2G system. The main components in this system include EVs, aggregators and the power grid. EVs not only act as loads but also participate in various other grid operations. The EV to power grid operations include *(a)* charging where EV draws power from the grid, *(b)* discharging where EV supplies power to the grid during times of peak power demand and helps to reduce the load on the grid and *(c)* frequency regulation, an operation that is used to stabilize the grid frequency by performing frequent charging and discharging operations.

Aggregators are entities that act as mediators between end users (*viz.*,EVs) and the utility operator. Aggregators are particularly useful in coordinating the discharging and frequency regulation operations between the EVs and the power grid. This is because individual EVs have very small power capacities in comparison with the scales of power generation and distribution at the power grid. Therefore, for efficient discharging and frequency regulation operations, a large number of EVs are required. An aggregator manages multiple EVs upto 400 in number [19] and helps in efficiently managing these operations [24]. With aggregators acting as intermediaries between the utility power grid operator and the EVs [15], all communication messages between the EVs and the power grid pass through aggregators. This is the system model used in our work.

The requirements and specifications for communication between the EVs and the electric

power grid are established by the SAE communication standards [9]. SAE J2847/1 standard provides specifications for forward power flow *viz.,*charging operation between the EVs and the power grid [17]. Whereas, SAE J2847/3 standard provides specifications for reverse power flow *viz.,*discharging and frequency regulation operations between the EVs and the power grid [18].

Integration of distributed sources of energy such as EVs with the power grid comes with the penalty of making the grid susceptible to a range of cyber-physical attacks. These include large scale attacks if many of these edge devices *viz.,*EVs, are hacked.

The vulnerability of the V2G system to a wide range of attacks is mainly due to the following factors:

- Utilities do not have direct control over EVs and hence are unable to enforce strict policies for secure communication.
- Customer sites lack enough security. This leads to exposure of EVs to physical attacks such as attacks on locks and other anti-theft mechanisms [4]. There is also an additional risk of exposure to cyber attacks due to insufficient knowledge, for instance, in setting up passwords.
- Direct interaction of customers with EVs may be used to unethically manipulate energy consumption data [3]. Compromised grid edge devices (EVs) may also be used to launch attacks that cause grid instabilities and blackouts [11].

The growing number of sophisticated attacks (Chapter), necessitates the need for the *development of an advanced cyber-physical attack detection and resiliency framework.*

Chapter 3: Threat Model

Integrating EVs (that act as variable loads) with the power grid entails many security considerations. Tab. 3.1 shows the possible attacks on a V2G system [12, 25, 26]. The attacks can be broadly classified into Network-based attacks and Component-based attacks.

Network-based attacks are the ones where the V2G communication network is compromised. These include Man-in-the-middle attacks (MITM) that either passively intercept and observe the communication by eavesdropping or actively modify communication by injection, replay or repudiation of traffic. Denial of Service (DoS) attacks on the V2G network include jamming of signals and dropping of packets. For instance, when EVs initially connect to the grid, information such as customer details and location data are exchanged. Eavesdropping on such information compromises customer privacy. As another example, transmitted control commands and updating of firmware, software, drivers *etc.* also affect system stability, safety and reliability. Active modification of such traffic has an impact on the functioning of the V2G components [10].

Component-based attacks are those attacks where one or more of the components in the V2G system are compromised. These include violation of authentication and/or authorization at the utility system (components of the power grid including aggregators), spoofing of utility system components, compromise of end devices (Electric Vehicles) and Denial of Service (DoS) attacks on utility system components by the exhaustion of resources at aggregator and/or Power Grid. These threats make the V2G sub-system of the power grid highly susceptible to attacks. Hence V2G system security is of paramount importance for the safe operation of the power grid.

Table 3.1: Threats in the Vehicle-to-Grid System

Network-based Attacks	**Component-based Attacks**
Passive Man-in-the-middle (MITM) attacks - intercept communication and analyze traffic (eavesdropping) Active Man-in-the-middle (MITM) attacks - modify, inject, replay, repudiate traffic Denial of Service (DoS) attacks - jamming of signals, dropping of packets	Violation of authorization and/or authentication at the utility system (components of the power grid including aggregators) Utility system (components of the power grid including aggregators) spoofing Compromise and spoofing of end devices (Electric Vehicles) Denial of Service (DoS) attacks on utility system components - exhaustion of resources at aggregator and/or Power Grid

Our goal is to *securely integrate EVs with the power grid*. To this end, we monitor and secure the critical component *i.e.,* aggregator, through which EVs are connected to the grid. We focus on the following attacks [12, 25, 26] on the aggregator:

1. **Aggregator Compromise (Component-based attack)** – Aggregator vulnerabilities can be used by attackers to invoke malicious behavior that result in violations of specifications.

2. **Electric Vehicle Compromise (Component-based attack)** – Multiple edge devices (in this case EVs) can be compromised and used to send malicious data to the aggregator.

3. **Network Compromise (Network-based attack)** – The network links connecting the

aggregator to the EVs can be compromised to intercept and / or modify traffic going to the aggregator.

According to our system model [Chapter], the aggregator receives inputs from two sources, the EVs and the power grid. Therefore, secure integration of EVs with the power grid requires monitoring of communication between the aggregator and EVs such that compromise of either the EVs or the network connecting them to the aggregator does not result in aggregator compromise. In other words, our goal is to ensure that the aggregator is *intrusion tolerant*.

Chapter 4: Cyber-Physical Anomaly Detection Engine

Our focus is to secure the aggregator, one of the main components of the V2G system, thereby increasing the reliability and the resiliency of such systems against both cyber and physical attacks. Power grid systems are often time-sensitive. Therefore, our goal is to not only detect anomalies accurately but also to do it in ”real-time” i.e., we detect an anomaly in a malicious network packet *instantaneously* upon its arrival at the aggregator. The system model used in our work is the model adopted from literature [15] where aggregators act as intermediaries between the utility (power grid) operator and the EVs [Chapter]. Hence all communication messages between the EVs and the power grid pass through the aggregators. In order to ensure safe operation of the aggregator, we monitor operations on both sides, *i.e.,* the EV side as well as the power grid side, as explained below.

We have designed an Anomaly Detection Engine for the aggregator that detects unexpected packets during communication. This anomaly detection engine uses information from multiple sources to monitor the system. The system architecture is as shown in Fig.4.1. Anomalous incoming packets to the aggregator are dropped thereby ensuring *intrusion tolerance* at the aggregator.

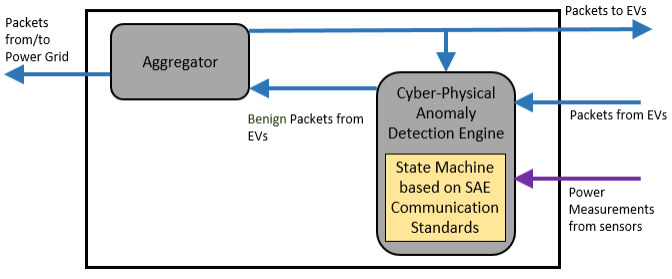

Figure 4.1: Secure Aggregator Architecture.

The anomaly detection engine uses *(a)* two levels of deep packet inspection along with *(b)* timing constraint validations and *(c)* power measurement validations to detect anomalies in system behavior. An anomaly is detected whenever the system deviates from the expected system behavior. Expected system behavior is defined based on the communication standards in the V2G system, as discussed in detail below.

4.1 Deep Packet Inspection

In order to detect anomalies at the aggregator, data present in incoming packets from EVs as well as timing constraints associated with them are monitored. Two levels of deep packet inspection are used for this purpose as explained in detail below.

4.1.1 Deep Packet Inspection Level 1

At this level, the sequences of messages in incoming packets are validated. To do this, an aggregator state machine is created with valid states and state transitions. This state machine is based on valid message sequences established by communication standards between the aggregator and the EVs as well as between the aggregator and the power grid. The requirements and specifications for communication between EVs and the electric power grid are established by the SAE J2847/1 standard for forward power flow that includes charging [9, 17] and SAE J2847/3 standard for reverse power flow that includes discharging and frequency regulation [9, 18]. A list of valid messages or commands between the aggregator and the EVs as well as the aggregator and the power grid is created. Then a list of valid command sequences formed by these messages is created to verify if the packets coming into and going out of the aggregator follow these valid sequences.

These valid messages and message sequences are in accordance with the SAE communication standards

Since the SAE standards are proprietary, complete details are not provided. However, sufficient details on the types and sequences of messages are provided below in Fig. 4.2 for a better understanding of our work. Fig. 4.2 shows the various messages in each of the three EV-grid operations, *(a)* charging, *(b)* discharging and *(c)* frequency regulation.

Table 4.1: SAE J2847/1 and SAE J2847/3 Message Types

Charging and Discharging	**Frequency Regulation**
Flow Reservation, Power Status, Flow Reservation List	DER Setup/Initialization, Power Status, DER Status, DER Availability, DER Control List

Fig. 4.2 shows the principal sub-states in the aggregator state machine that are generated from the list of valid command sequences enumerated for the three EV-grid operations. The state diagram captures the various states and state transitions when EVs are subscribed *i.e.,* connected to the power grid (denoted by 'S') as well as when they are not subscribed to the grid (denoted by 'NS'). When subscribed, EVs can be involved in any one of the three aforementioned EV-grid operations. Each state represents a *sequence* of messages. The state transitions in various cases are as follows:

a) Subscribed and Charging / Discharging - Flow reservation is the process in which the EV is assigned a subscription period (*i.e.,* period when connected to the grid) for charging/discharging. First, a flow reservation is established. This may be followed by one or

more new flow reservations after cancellation of a previously established flow reservation. Once the subscription period begins, the EV starts sending periodic power status updates (*i.e.,* information related to the amount of power drawn) to the grid through the aggregator. In parallel, it also periodically fetches the flow reservation list from the grid, through the aggregator, to check for any updates in the subscription period.

b) Subscribed and Frequency Regulation - first the set up and initialization of the Distributed Energy Resource (DER) is performed where the EV specific information (such as battery capacity, *etc.*) is sent and the process of frequency regulation is initiated. Once the subscription period begins, the EV starts sending periodic updates including power status, DER status and DER availability to the grid. In parallel, it also periodically fetches the DERControl list from the grid to check for any updates in the subscription period. The DERControl List is equivalent to the flow reservation list in case of charging/discharging operations and contains subscription period related information.

c) Not subscribed - when the EV is not subscribed to the power grid (*i.e.,* not engaged in any of the three EV-grid operations), it periodically fetches updates on pricing and load control related information from the grid to make a decision on when to charge / discharge.

4.1.2 Deep Packet Inspection Level 2

At this level, *command or message data* in the packets coming into the aggregator are validated. This next level of analysis of command data in packets is particularly useful for monitoring the aggregator/EV side of communication that involves a highly vulnerable component of the V2G system,*viz.,*the EV (edge device). Two important parameters in the EV to power grid

communication that are most likely to be tampered by adversaries are:

a) the subscription period, that defines the duration of charging/discharging during the charging/discharging/frequency regulation operations and

b) the state of charge (SOC), that defines the percentage of charge in the battery of the connected EV.

Hence, we need to inspect packets to monitor these quantities. During charging and discharging operations, the EV periodically fetches the flow reservation list from the grid while also periodically updating its power status to the grid. Similarly, during frequency regulation operation, the EV fetches the DERControl list from the grid. The flow reservation list and the DERControl list contain the start and end of the subscription period. This data is used to verify that there are no power status updates outside the specified time interval. Power status updates occur only during the subscription period.

The power status update messages contain vehicle SOC related information in terms of amount of power drawn. These power measurements are validated against physical power measurements as discussed below.

4.2 Consistency of Cyber States with Physical States

We propose another level of security check by verifying the consistency of Cyber states with the Physical states. Power measurements are obtained from the power grid network through sensors [6] and compared with the power measurements reported in the cyber messages, *i.e.,* the power status update messages. In order to reduce the possibility of simultaneous tampering of both the cyber messages and the power measurements, the sensors used for cross-validation are located

geographically far away from the EVs.

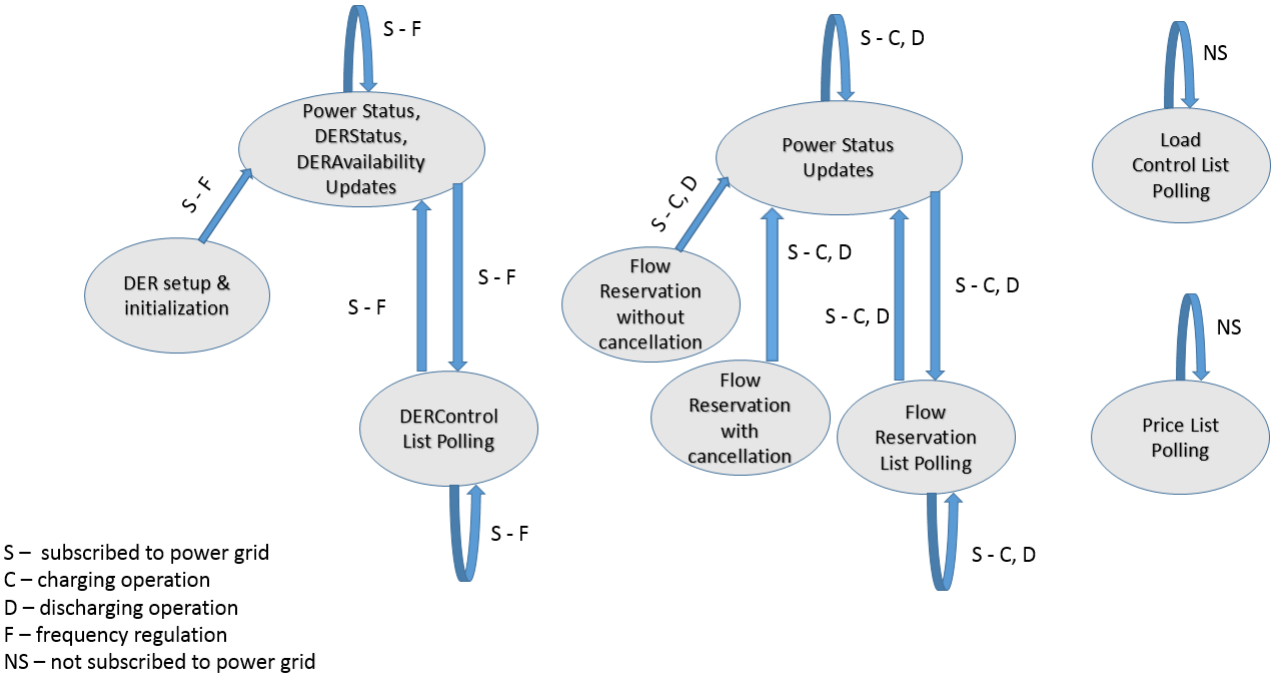

Figure 4.2: Sub-states of the aggregator state machine.

4.3 Timing Requirements Validation

Power grid systems such as V2G system are real-time systems that must satisfy certain time related constraints. Our anomaly detection engine monitors incoming packets to ensure that timing constraints are enforced. Periodic messages have a predefined frequency. The frequency of such periodic messages coming into the aggregator are monitored. At the aggregator, the frequency of a given periodic message is monitored by checking the time elapsed between two occurrences of the message. Therefore, it is not required that the clocks at the aggregator and the EVs be synchronized.

In all of the validation techniques above, consistency of data obtained from one source is

verified against data obtained from a different source as explained next. This makes it more difficult to tamper with the data so as to produce false consistencies. In case of Deep Packet Inspection Level 1 and Timing Requirements Validation, packets are verified against information obtained from the communication standards documentation. For Deep Packet Inspection Level 2, *(a)* SOC related data is validated against physical power measurements, *(b)* subscription period related data is validated using messages from the power grid side of communication that contain flow reservation list/DERControl List and messages from the EV side of communication that contain power status updates.

Chapter 5: Implementation

To evaluate the anomaly detection engine, we implemented a prototype in Python2.7 on the Intel i7 NUC platform [5]. Tab.5.1 summarizes the platform details.

Table 5.1: Implementation Platform Details

Platform	**Processor**	**Memory**	**Operating System**
Intel i7 NUC	Intel(R) Core(TM) i7-7567U CPU @ 3.5 GHz, 4 cores	32 GB RAM, 128 GB HDD	Ubuntu 16.04

The algorithm for anomaly detection is as shown in Algorithm 5.1. The engine inspects all the received packets from EVs and filters out the malicious ones. To handle multiple EVs at an aggregator, there are multiple instances of packet inspection that execute in parallel, each instance monitoring a single EV. The maximum number of such parallel instances is same as the maximum number of EVs that can be handled by a given aggregator. The source EV of each received packet is first identified and the packet is then passed onto the corresponding instance of packet inspection. Based on the output from the packet inspection module, the packet is either dropped or output. The algorithm for packet identification is shown in Algorithm 5.2. The source and destination IP addresses of a packet are used to determine the instance i to which the packet must be passed on. If the packet is from a new EV, the packet is either assigned to a free instance (an instance i in idle state) or dropped if all instances are busy (*i.e.,*-1 is returned). An instance moves to idle state if it does not receive a packet from the EV for a specified time period (i.e., connection to the EV is

dropped after a timeout).

Algorithm 5.1 Anomaly Detection Engine

```
Input: All packets from EVs
Output: Benign packets from EVs
Spawn 'n' instances of Packet_Inspection module
for packet in packets do
    instance_i := Packet_Identification(packet)
    if instance_i not equals -1 then
        //Call ith instance of Packet_Inspection
        anomalous := Packet_Inspection(packet)
        if anomalous then
            drop packet
        else
            output packet
        end if
    else
        drop packet
    end if
end for
```

The algorithm for packet inspection is as shown in Algorithm 5.3. The python scapy module [7] is used for packet parsing. Using the specifications provided by the SAE communication standards for forward and reverse power flows [17, 18], a state machine of all possible valid states and state transitions is created. The received packet is parsed and *(a)* two levels of data validations, *(b)* timing constraint validation and *(d)* cyber-physical consistency validation using power measurements from physical sensors embedded in the network are performed to detect anomalous behavior (as discussed in detail under System Architecture). Based on the outcome of this inspection, the state machine is advanced and anomaly notifications are returned accordingly.

As previously mentioned, an instance of the packet inspection module moves to idle state if no packet is received until timeout.

Algorithm 5.2 Packet Identification

```
Input: Packet
Output: Instance of Packet_Inspection module
//Determine to which instance of Packet_Inspection module does packet belong
for i in instances do
  //Checking src and dst IP addresses
  if packet IP equals IP monitored by i then
    return i
  else
    if i is free then
      set monitoring IP of i to packet src IP and return i
    else
      return -1
    end if
  end if
end for
```

Algorithm 5.3 Packet Inspection

Input: Packet
Output: Packet is anomalous or not
if no packet received until timeout **then**
  transition to idle state
else
  //*Deep Packet Inspection 1*
  if packet.payload matches expected packet.payload in current state **then**
    //*Deep Packet Inspection 2*
    if current time in subscription period and packet.payload contains power status update **then**
      pass
    else
      raise anomaly - invalid subscription
    end if
    //*Cyber and Physical States Consistency Validation*
    if (packet.SOC - packet.lastSOC) matches power measurements from sensors in the elapsed time interval **then**
      update packet.lastSOC
    else
      raise anomaly - inconsistent power
    end if
    //*Timing Constraints Validation*
    if (packet.arrivalTime - packet.lastArrivalTime) satisfies frequency constraints **then**
      update packet.lastArrivalTime
    else
      raise anomaly - inconsistent frequency
    end if
    transition to next state
  else
    raise anomaly - unexpected packet
  end if
end if
if anomaly raised **then**
  return anomalous
else
  return not anomalous
end if

Chapter 6: Evaluation

The anomaly detection engine is placed close to the aggregator in the communication path between the aggregator and the EVs as shown in Fig. 4.1. This makes it important to ensure that it does not introduce significant delay to the packet transfer rate at the aggregator. We therefore evaluate the prototype of our anomaly detection engine in terms of both accuracy and performance.

1. **Accuracy** is measured in terms of *false positives* and *false negatives*.

2. **Performance** is measured by comparing the average time taken by the anomaly detection engine to inspect a packet with the minimum packet inter-arrival time on network.

6.1 Test Cases to measure Accuracy

Currently, there are no EVs/EVSEs that support the SAE J2847/1 and SAE J2847/3 standards since these communication standards are still in the process of development.

Therefore, to test our anomaly detection engine, we developed a python script to generate packets with custom HTTP payloads according to specifications provided by currently available SAE communication standards [17, 18]. Details are explained below.

6.1.1 Deep Packet Inspection Level 1

To test Level 1 packet inspection that monitors the sequence formed by incoming and outgoing packets against the list of all possible valid sequences, the following tests were performed:

- **Test for false positives using valid test cases** - We generated packets that form all known valid sequences from the SAE standards [17, 18] for testing. These valid sequences consist of parallel as well as repeating sequences with repeating subsequences [Tab. 6.1]. This

complexity gives rise to the possibility of a lot of valid variations. Hence the large number of possible test cases. This is explained with an example below.

Tab.6.1 gives the number and type of valid sequences for the three EV-grid operations. For instance, consider the charging operation. As shown in Fig. 4.2, first a flow reservation with or without cancellation is performed (note the existence of two possibilities already). Then the EV starts sending periodic power updates to the grid during its subscription period for charging. In parallel, the EV also periodically fetches the flow reservation list from the grid. Periodic messages give rise to repeating sequences and increase the number of possible valid variations. Similarly, parallel sequences of messages (power updates and fetching of flow reservation list in this case) also increase the number of possible valid variations. This is because one or more messages from a parallel sequence (say, power updates in this case) can arrive anywhere between messages in a related parallel sequence (fetching of flow reservation list in this case). The sequence to which the message belongs is identified using the message data.

We tested all known valid sequences as shown in Tab. 6.1 including variations due to parallel and repeating sequences. Our prototype did not signal an anomaly in any of the valid cases. In other words, *no false positives or false alarms*, *i.e.,* signaling anomaly in case of valid sequences, were encountered.

- **Test for false negatives using invalid test cases** - We generated various types of invalid test cases including well crafted and random test cases. These test cases consisted of invalid packets being randomly placed among packets forming valid sequences (python's random module was used to randomly choose the location). In the well crafted test case, the payload

of an invalid packet followed the communication standard specifications, *i.e.,* the payload contained one of the valid messages but was placed so as to form an invalid sequence. The only difference in the random test case was that the payload of an invalid packet did not follow the communication standard specifications. The number of possible test cases here is much larger than in case of known valid sequences due to the fact that a large number of random variations to the given set of valid sequences is possible. It was therefore important to generate a large set of invalid test cases to verify that the state machine implemented in the prototype did not have any invalid transitions. Our prototype detected anomalies in each of these test cases, *i.e.,* no false negatives were encountered. These results are summarized in Tab. 6.2.

Table 6.1: Valid Command Sequences

Grid Operation	**Number of Parallel Sequences**	**Number of Repeating Sequences and Subsequences**	**Total number of Valid Sequences(including parallel, repeating and non-repeating sequences)**
Charging	2	6	150
Discharging	2	3	48
Frequency Regulation	2	2	28

6.1.2 Deep Packet Inspection Level 2

To test Level 2 packet inspection (that monitors the SOC and subscription period), the following tests were performed:

- **Test for false positives using valid test cases** - We generated packets with consistent SOC

and subscription period related data for testing. Again, our prototype did not detect anomalies in any of the valid cases, *i.e.,* no false positives were encountered.

- **Test for false negatives using invalid test cases** - We generated various types of invalid test cases. These test cases consisted of packets with inconsistencies. With respect to the subscription period, the arrival time of packets containing power status updates were modified so as to be inconsistent with the time intervals specified in packets containing the flow reservation list. With respect to SOC, the power information in packets containing power status updates were modified so as to be inconsistent with physical power measurements. Our prototype successfully detected anomalies in all of the above cases, *i.e.,* no false negatives were encountered. These results are summarized in Tab. 6.2.

Currently, the physical power measurements are read from a file filled with crafted power measurements. The crafted power measurements consist of values that are consistent with the values reported in cyber messages as well as values that are different from those reported in cyber messages. The above crafted power measurements file can easily be replaced by a CSV spreadsheet file containing the actual physical power measurements, for instance, that generated by the PQube3 sensors [8] and will not impact the accuracy evaluation of the anomaly detection engine.

6.1.3 Timing Requirements Validation

To test the timing based validation that monitors the frequency of periodic messages, the following tests were performed:

- **Test for false positives using valid test cases** - We generated packets with valid periodicities as specified in the SAE standards [17, 18] for testing. As one of the examples,

power status updates occur every 5 minutes during discharging and frequency regulation. Once again, our prototype did not signal anomalies in any of the valid cases, *i.e.,* no false positives were encountered.

- **Test for false negatives using invalid test cases** - We generated various types of invalid test cases. These test cases consisted of packets containing messages with periodicities different (*i.e.,* periodicities lower and higher than expected values) from expected values as specified in the SAE standards [17, 18]. Our prototype successfully detected anomalies in all of the above cases, *i.e.,* no false negatives were encountered. These results are summarized in Tab. 6.2.

6.2 Performance measurement

We compare the average time taken by the anomaly detection engine to inspect a packet with the minimum inter-arrival time between two packets on the network to determine whether or not the anomaly detection engine introduces significant delay.

In this evaluation, the total number of EVs handled by the aggregator were varied up to a maximum of 400 based on literature [19]. The average time taken by the anomaly detection engine to inspect a packet is approximately 0.0082 seconds as shown in Fig. 6.1. According to the smart grid communication requirements specified by the Department of Energy, the range of network bandwidth in case of Electric Transportation applications is 9.6 to 56 kbps [2]. The average packet size in this application is 0.5 KB, average being computed over the set of messages defined in SAE standards. Thus, the network bandwidth in terms of number of packets is approximately 2 to 14 packets per second and the inter-arrival time between packets on the network is 0.07 to 0.5 seconds. Tab. 6.3 summarizes these results.

Table 6.2: Accuracy of Anomaly Detection Engine

Component of Anomaly Detection Engine Tested	**Test Description**	**Test Criterion**	**Test Outcome**
Deep Packet Inspection Level 1	Tested against set of all known valid sequences including variations due to parallel and repeating sequences [Tab.6.1]	Test for False Positives	No false positives encountered
Deep Packet Inspection Level 1	Tested against set of partially valid sequences including variations due to parallel and repeating sequences [Tab.6.1]	Test for False Negatives	No false negatives encountered
Deep Packet Inspection Level 1	Tested against set of invalid sequences including variations due to parallel and repeating sequences [Tab.6.1]	Test for False Negatives	No false negatives encountered
Deep Packet Inspection Level 2	Tested against consistent SOC related data - during charging, discharging and frequency regulation operations	Test for False Positives	No false positives encountered
Deep Packet Inspection Level 2	Tested against consistent subscription period related data - during charging, discharging and frequency regulation operations	Test for False Positives	No false positives encountered
Deep Packet Inspection Level 2	Tested against inconsistent SOC related data - during charging, discharging and frequency regulation operations	Test for False Negatives	No false negatives encountered
Deep Packet Inspection Level 2	Tested against inconsistent subscription period related data - for each of the 3 EV-grid operations, only the start time, only the end time and both start, end times were made inconsistent and tested	Test for False Negatives	No false negatives encountered
Timing Requirements Validation	Tested against packets with valid periodicities	Test for False Positives	No false positives encountered
Timing Requirements Validation	Tested against packets with invalid periodicities, periodicities lower and higher than expected values were tested	Test for False Negatives	No false negatives encountered

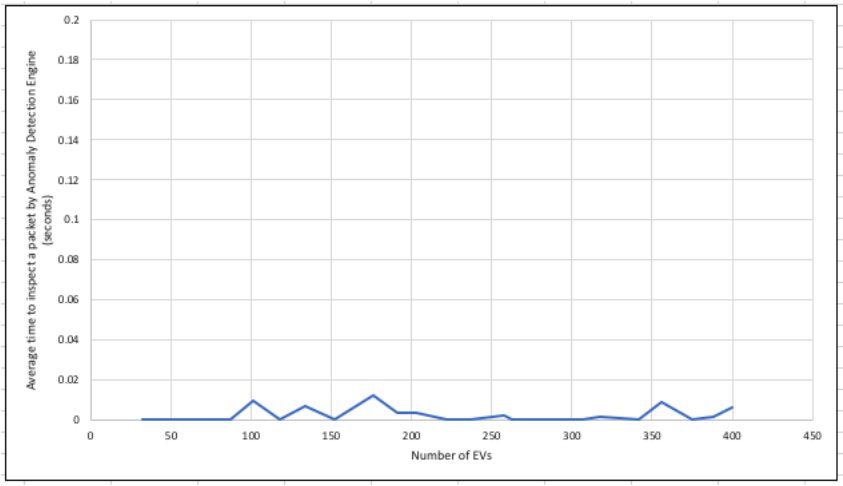

Figure 6.1: Performance of the Anomaly Detection Engine

Table 6.3: Performance of Anomaly Detection Engine

Average packet size	**Network Bandwidth**	**Inter-arrival time of packets**	**Average time taken by Anomaly Detection Engine to inspect a packet**
0.5 KB	9.6 to 56 kbps OR 2 to 14 packets per second	0.07 to 0.5 seconds	0.0082 seconds

6.3 Discussion

We have tested for false positives and false negatives to validate the accuracy of our prototype. The lack of false positives and false negatives show that the state machine is correctly implemented. It is important to note that this evaluation is carried out based on the previously stated assumption that the physical state of the system is not compromised, i.e, power measurements from sensors are genuine. If this is not true, then an attacker can tamper with the cyber and physical states of the system so as to obtain consistency and conceal attack from the anomaly detection engine. We intend to address the possibility of compromised physical state of the system in our future work.

The performance of our prototype has been evaluated by comparing the time taken for packet inspection with inter-arrival time of packets on network. As seen from the results, the average time for packet inspection by the anomaly detection engine : 0.0082 seconds, is lower than the minimum inter-arrival time of packets: 0.07 seconds. Therefore, our anomaly detection engine does not introduce significant delay into the system.

The network we have considered so far in the evaluation is the network connecting EVs and the aggregator. We now discuss the impact of the other network *i.e.,* the network connecting the anomaly detection engine and the power sensors. This network is used to obtain power measurements to validate consistency between the cyber and physical states of the system and constitutes the 'Physical' side of our Cyber-Physical anomaly detection engine. This network does not impact the currently reported performance of the anomaly detection engine when it has high bandwidth in gbps (as compared to the bandwidth of the EV network in kbps).

Currently, the prototype of the anomaly detection engine does not handle encryption of packets. In future work, we intend to handle encrypted packets by using mechanisms to share decryption keys between the aggregator and the anomaly detection engine.

Chapter 7: Related Work

There has been a wide range of attacks on the power grid that has triggered research towards securing its various components such as the advanced metering infrastructure (AMI) and smart inverters. Examples of intrusion detection systems for cyber-physical systems include the work on securing the advanced metering system by using specification based intrusion detection [12]. This IDS monitors the cyber state of the system by observing traffic among access points and meters at various layers to ensure expected behavior. We use similar techniques to monitor the cyber state of a V2G aggregator. In addition, we also check for consistency of cyber and physical states of the system.

Apart from the various components of the power grid, the distribution networks have also received attention. The paper by Liao et al [20] focuses on enhancing power grid security by using micro-synchrophasors as a tool to monitor and manage distribution networks. The high fidelity, time-synchronized phase angle and voltage magnitude data obtained from micro-synchrophasors helps track events originating at local distribution. This work is similar to our work in that it uses data from sensors for monitoring. In other words, the physical component of the cyber-physical system is being monitored.

With increase in number of DERs being integrated with the power grid, there has been effort in the direction of securing these DERs [23]. The paper discusses the architecture of cyber-physical power system with penetration of DERs, analyzes related cyber security challenges, summarizes important attack scenarios and proposes a DER resilience analysis methodology to prevent, detect and respond to attacks. The paper provides a generalized analysis for DERs. However, specific DERs have their own challenges. Our work focuses on a specific DER, i.e., EVs.

With respect to the V2G system, implementation and optimization have so far received a lot of focus. The paper by Guille et al [16] discusses a framework to integrate EVs with the power grid. The paper by Mal et al [14] focuses on optimizing the charging operation in the V2G system and thereby efficiently balancing the load on power grid. However, attention has recently shifted towards the security of EVs in the power grid. Chen et al [13] propose an efficient and secure authentication scheme for V2G networks that preserves privacy. The authentication scheme provides anonymity, dynamic management and aggregation in the of V2G network. The paper focuses on securing the communication of EVs in the V2G system of power grid. On the other hand, our work focuses on securing the aggregator, an important component of the V2G system, by increasing its resiliency to attacks.

Chapter 8: Conclusion

In this work, we have presented the architecture of an Anomaly Detection Engine for the aggregator. It is an important component of the Vehicle-to-Grid System that integrates Electric Vehicles with the Power Grid. The Anomaly Detection Engine uses *(a)* two levels of deep packet inspection along with *(b)* timing constraint validations and *(c)* power measurement validations to detect unexpected system behavior. In all the validation techniques used for anomaly detection, consistency of data obtained from one source is verified against data obtained from a different source such as in Deep Packet Inspection level 2, SOC related data in cyber messages is validated against physical power measurements. This makes it difficult for adversaries to tamper with these data so as to produce false consistencies.

References

[1] Cyber-physical systems. URL: https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286.

[2] Doe smart grid communication requirements. URL: https://energy.gov/sites/prod/files/gcprod/documents/Smart_Grid_Communications_Requirements_Report_10-05-2010.pdf.

[3] Ev cyber attacks. URL: https://csrc.nist.gov/CSRC/media/Presentations/.

[4] Ev physical attacks. URL: http://www.theiet.org/sectors/transport/.

[5] Intel i7 nuc. URL: https://www.intel.com/content/www/us/en/products/boards-kits/nuc/kits/nuc7i7bnh.html.

[6] Pqube3. URL: http://www.powersensorsltd.com/PQube3.php.

[7] Python scapy. URL: https://scapy.readthedocs.io/en/latest/.

[8] Recording pqube 3 readings. URL: https://www.powerstandards.com/product/pqube-3/sample-output/.

[9] Sae v2g standards. URL: http://www.sae.org/standardsdev/.

[10] *Security Considerations for the EV Charging Infrastructure*. URL: https://www.iaria.org/conferences2012/filesSECURWARE12/Electric_Vehicle_Charging_Infrastructure_KeynoteRainerFalk.pdf.

[11] Ukraine power grid attack. URL:
https://www.technologyreview.com/s/603262/ukraines-power-grid-gets-\
hacked-again-a-worrying-sign-for-infrastructure-attacks/.

[12] R. Berthier, W. H. Sanders, and H. Khurana. Specification-based intrusion detection for advanced metering infrastructures. In *2011 IEEE 17th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2011.

[13] J. Chen, Y. Zhang, and W. Su. *An anonymous authentication scheme for plug-in electric vehicles joining to charging/discharging station in vehicle-to-grid (v2g) networks*. China Communications, 2015.

[14] S. Mal et al. *Electric vehicle smart charging and vehicle-to-grid operation*. International Journal of Parallel Emergent and Distributed Systems, 2012.

[15] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis. The role of aggregators in smart grid demand. *IEEE Journal on Selected Areas in Communications*, 31:1247–1257, 2013.

[16] C. Guille and G. Gross. A conceptual framework for the vehicle-to-grid (v2g) implementation. *Energy Policy*, 2009.

[17] Sae International. Sae j2847-1 communication for smart charging of plug-in electric vehicles using smart energy profile 2.0. 2013.

[18] Sae International. Sae j2847-3 communication for plug-in vehicles as a distributed energy resource. 2013.

[19] A. Y.S. Lam, K.C Leung, and V. O.K. Li. Capacity estimation for vehicle-to-grid frequency regulation services with smart charging mechanism. *arXiv:1410.1282v4 [cs.SY]*, 2015.

[20] A. L. Liao, E. M. Stewart, and E. C. Kara. *Micro-synchrophasor data for diagnosis of transmission and distribution level events*. in Transmission and Distribution Conference and Exposition (T and D), IEEE/PES, 2016.

[21] Nescor. *" Electric sector failure scenarios and impact analyses version 3.0"*. 2015.

[22] C. Niddodi, S. Mohan, and T. Yardley. Securing electric vehicles in the power grid. In *Proceedings of 29th Euromicro Conference on Real-Time Systems (ECRTS17) Work-in-Progress Session*, 2017.

[23] J. Qi, A. Hahn, X. Lu, J. Wang, and C.C Liu. Cybersecurity for distributed energy resources and smart inverters. *in IET Cyber-Physical Systems: Theory and Applications, pp*, pages 28–39, 2016.

[24] M. A. Rahman, F. Mohsen, and E. Al-Shaer. A formal model for sustainable vehicle-to-grid management. *SEGS '13 Proceedings of the first ACM workshop on Smart energy grid security*, pages 81–92, 2013.

[25] N. Saxena, S. Grijalva, V. Chukwuka, and A. V. Vasilakos. Network security and privacy challenges in smart vehicle-to-grid. *IEEE Wireless Communications, pp*, pages 2–12, 2016.

[26] Y. Zhang, S. Gjessing, H. Liu, H. Ning, L. T. Yang, and M. Guizani. Securing vehicle-to-grid communications in the smart grid. *IEEE Wireless Communications, pp*, pages 66–73, 2013.