

Analysis of Passive Inference of Attacks on CPS Protocols

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE MINOR PROJECT

By

Rahul Kumar 1906049

Brij Mohan Diwakar 1906044

Kumawat Lakhan Makhanlal 1906055

UNDER THE SUPERVISION OF
Asst. Professor CSE Dept, NIT PATNA
Dr. Kakali Chatterjee



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY PATNA
(An Institute of National Importance)
MAHENDRU, PATNA, BIHAR - 80000



CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY PATNA

This is to certify that **Brij Mohan Diwakar** (1906044), **Kumawat Lakhan** (1906055), **Rahul Kumar** (1906049) have carried out the minor project entitled “**Analysis of passive inference of attacks on CPS communication protocols**” under the supervision of **Dr. Kakali Chatterjee**, Department of Computer Science and Engineering, NIT Patna. This project is bonafide work done by them for the fulfillment of the requirements for the minor project.

Dr. Kakali Chatterjee
Supervisor

Dr. Maheshwari Prasad Singh
Head of Department CSE

DECLARATION

We students of 7th semester hereby declare this project entitled "**Analysis of passive inference of attacks on CPS communication protocols**" has been carried out by us in the department of Computer Science and Engineering of National Institute of Technology Patna under the guidance of **Dr Kakali Chatterjee**, Department of Computer Science and Engineering, NIT Patna. No part of this work has been submitted to any other institute nor copied from any journals or articles or project reports.

Brij Mohan Diwakar
1906044

Kumawat Lakhan Makhanlal
1906055

Rahul Kumar
1906049

Place : -----

Date : -----

ACKNOWLEDGEMENT

We hereby take the privilege to express our gratitude to all the people who were directly or indirectly involved in the execution of this work, without whom this project would not have been a success. We extend our deep gratitude, respect and obligation to our project mentor, **Dr. Kakali Chatterjee**, Department of Computer Science and Engineering, NIT Patna. for being a constant source of inspiration. Our heartiest thanks to our classmates who have supported us in all ways. Words are inadequate to express our gratitude to our parents and friends who have been supportive all the time. We would also like to thank our institution and the faculty members without whom this project would have been a distant reality. Above all we bow before the Almighty for his immense blessings throughout life.

1. Rahul Kumar 1906049
2. Brij Mohan Diwakar 1906044
3. Kumawat Lakhan Makhanlal 1906055

TABLE OF CONTENTS

TABLE OF CONTENTS	5
ABSTRACT	6
1. INTRODUCTION	7
1.1 SYSTEM DIAGRAM	8
2. MOTIVATION	8
3. OBJECTIVE	8
4. RELATED WORK	9
5. PROBLEM STATEMENT	10
6. PROPOSED MODEL	10
6.1. SECURITY PROTOCOLS TARGETED	10
6.2. CPS USED	11
6.3. CIRCUIT CONNECTIONS	12
7. WORKING MODEL	15
7.1. SNIFFING	15
7.2. DOS	16
7.3. AUTHENTICATION	17
7.4. MITM	18
8. FLOW DIAGRAM OF PROPOSED ATTACKS	19
9. MODEL DESCRIPTION	20
9.1. ALGORITHMS	21
10. IMPLEMENTATION RESULTS	22
11. CONCLUSION	23

ABSTRACT

The security of Cyber-Physical Systems (CPS) has been recently receiving significant attention from the research community. While the majority of such attention originates from the control theory domain, few approaches have addressed the problem from the practical perspective. We make no claims in this work that we offer a specific solution to a certain issue regarding CPS security, but rather that we offer a glimpse into potential future directions for these solutions. Indeed, our vision and ultimate goal is to attempt to merge or at least diminish the gap between highly theoretical solutions and practical approaches derived from insightful empirical experimentation, for securing CPS. This article takes a novel way to derive conceptions of CPS maliciousness based on passive measurements and analysis. The motivation for this approach is the dearth of malicious empirical data that can be gathered, inferred, and studied from within operational CPS contexts. Indeed, by scrutinizing unsolicited real traffic targeting routable, allocated used Internet Protocol (IP) addresses, we shed the light on attackers' intentions and actual attacks targeting ample of CPS communication and control protocols. In order to make such analysis possible, we first create and assess a novel probabilistic model that aims to filter out the noise (i.e., misconfiguration traffic) that is incorporated into darknet traffic. Subsequently, a near real-time inference algorithm is designed and implemented to detect CPS probing and denial of service activities. To this end, we characterize such misdemeanors in terms of their types, their frequency, their target protocols and possible orchestration behavior. The results also reveal coordinated groups of uninvited actions and covert probing operations against secret CPS protocols. We concur that the devised approaches, techniques, and methods provide a solid first step towards better comprehending real CPS unsolicited objectives and intents. As a result, we hope that this report will encourage researchers to create secure, specialized CPS models that take use of real-world threats and weaknesses deduced from empirical data in order to provide CPS that is genuinely trustworthy and secure.

1. INTRODUCTION

The analysis of CPS security from a control-theoretic perspective has undoubtedly received considerable attention. The cyber security research community has offered various approaches in an attempt to tackle numerous security aspects of CPS. Such approaches typically put less emphasis on the control system dynamics of CPS by essentially focusing on the cyber (i.e., communication networks, protocols, data, etc.) perspective. We classify a number of such fundamental approaches into four core categories as summarized in Table 1 and we subsequently discuss only a few of them, to maintain the focus of the report.

Table 1

A brief Classification of CPS Security Approaches from a Cyber Security Perspective

Analysis Perspective	Highlights	Ref.
Protocol Vulnerabilities	Modeling CPS protocols to detect anomalies	[2]
Network Measurements	Data-driven approaches to infer CPS cyber attacks	[4]

The employed approach exploits the fact that the generated communication traffic from Modbus is highly periodic. To this end, we designed and implemented an algorithm to initially capture the embedded periodicity in network traffic channels and consequently flag any deviations from it. We evaluated their anomaly detection approach using different real data sets extracted from an operational facility. To accomplish the latter task, we employed a dynamic Bayesian network and a probabilistic suffix tree as the underlying predictive model. Executed evaluations using synthetic data demonstrated that the proposed approach is able to accurately model normal traffic, flag certain deviations, and reduce the false positive rate.

Table 2

A brief Categorization of a few Proposed Threat Detectors for CPS.

Inference Approach	Attack Type	CPS Type	Dataset	Ref.
Anomaly-based	DoS attack	SCADA	Real	[1]
Signature-based	Authentication & spoofing attacks	Power utility	Real	[3]
Signature-based	MITM attacks	SCADA	Real	[6]

1.1 SYSTEM DIAGRAM

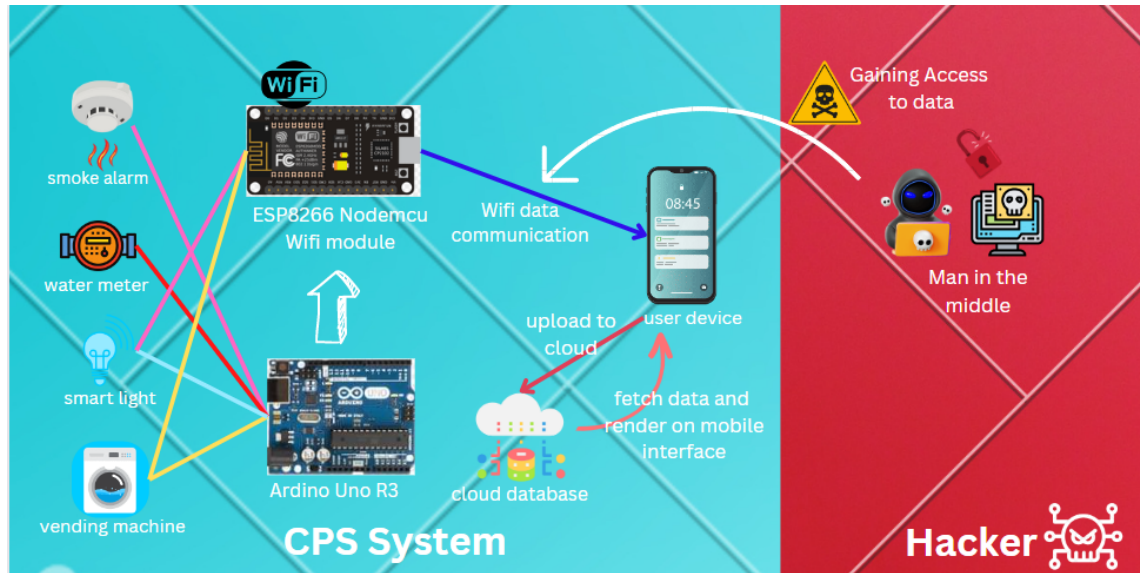


Fig 1.1 An overview of CPS model and Third party interaction

2. MOTIVATION

This study adopts a novel approach to derive conceptions of CPS maliciousness based on passive measurements and analysis. The motivation for this approach is the dearth of malicious empirical data that can be gathered, inferred, and studied from within operational CPS contexts. We actually provide insight on attacker intentions and actual attacks that target a variety of CPS communication and control protocols by closely examining unsolicited genuine traffic that targets routable, allocated but unused Internet Protocol (IP) addresses (i.e., darknet traffic).

3. OBJECTIVE

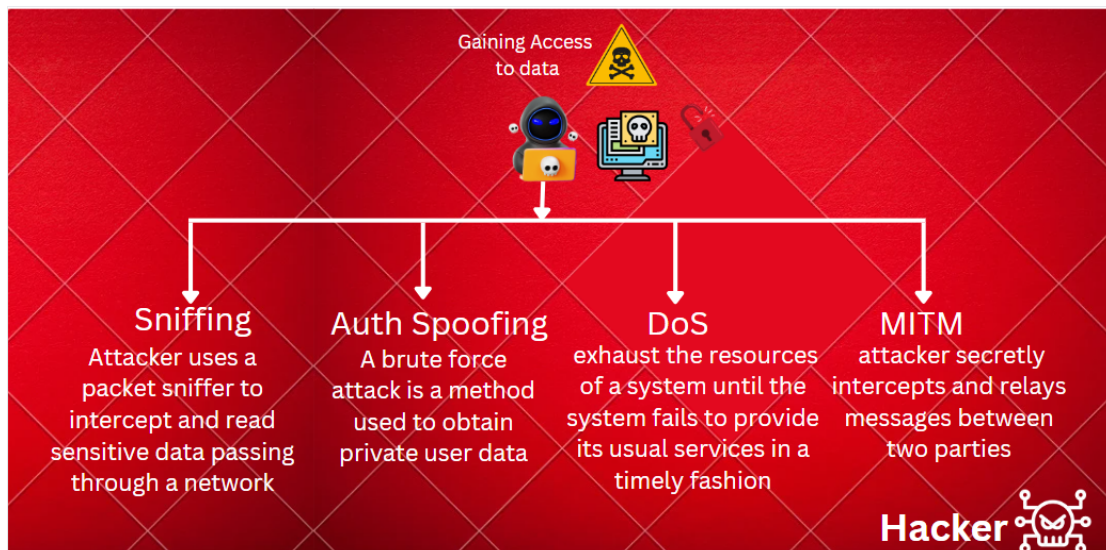


Fig 3.1 Overview of the attacks performed by hacker

4. RELATED WORK

1.Bou-Harb E. Passive inference of attacks on scada communication protocols. In: 2018 IEEE International Conference on Communications (ICC).

We learnt how to deduce, quantify, characterize, and analyze ideas of CPS maliciousness from this research. This article utilized passive measurements and analysis in a novel way to extract probing and denial-of-service (DoS) assaults targeting a variety of CPS communication and control protocols. The motivation for this was the dearth of hostile empirical data from operating CPS systems.

This literature provided a brand-new, incredibly useful probabilistic approach to clean darknet data by fingerprinting and then filtering out misconfiguration traffic. To recognise and decipher such CPS unsolicited traffic, an inference method together with characterisation modules was subsequently developed and put to use.

2.Shin S, Kwon T, Jo G-Y, Park Y, Rhy H. et al. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. Ind Inform, IEEE Trans in 2010.

From this paper we gained the information about intrusion detection for WISNs(Wireless industrial sensor networks) through various experiments on the previous IDSs devised for WSNs, with real motes.

3.Premaratne UK, Samarabandu J, Sidhu TS, Beresh R, Tan J-C. et al. An intrusion detection system for iec61850 automated substations. Power Del, IEEE Trans in 2010.

This paper discusses the use of an IDS tailored to counter the threats to an IEC61850 automated substation based upon simulated attacks on IEDs and a new method of evaluating the temporal risk of an intrusion for an electric substation.

4.Lee EA. et al. The past, present and future of cyber-physical systems: a focus on models. Sensors 2015.

From this paper we grasped the idea of better engineering of cyber-physical systems through better models.

5. PROBLEM STATEMENT

Vulnerability assessment of unsolicited real traffic targeting used IP addresses to analyze the impact of passive inference of attacks on CPS communication protocol.

6. PROPOSED MODEL

6.1. SECURITY PROTOCOLS TARGETED

WEP

WEP (Wired Equivalent Privacy) is the oldest and most common Wi-Fi security protocol. It was the privacy component established in the IEEE 802.11, a set of technical standards that aimed to provide a wireless local area network (WLAN) with a comparable level of security to a wired local area network (LAN). WEP uses secret keys to encrypt data. Both AP and the receiving stations must know the secret keys. There are two kinds of WEP with keys of either 64 bits or 128 bits. The longer key gives a slightly higher level of security (but not as much as the larger number would imply). In fact the user keys are 40bits and 104bits long, the other 24 bits in each case being taken up by a variable called the Initialization Vector (IV). When a packet is to be sent it is encrypted using a combination of the IV and the secret key. The resulting packet data looks like random data and therefore makes the original message unreadable to an outsider not knowing the key. The receiving station reverses the encryption process to retrieve the message in clear text.

WPA /WPA-2

WPA (Wi-Fi Protected Access) and WPA2 (Wi-Fi Protected Access 2) are two security standards that protect wireless networks. WPA2 is the second generation of the Wi-Fi Protected Access security standard and so is more secure than its predecessor, WPA. WPA2 (Wi-Fi Protected Access 2) is the second generation of the Wi-Fi Protected Access wireless security protocol. Like its predecessor, WPA2 was designed to secure and protect Wi-Fi networks. WPA2 ensures that data sent or received over your wireless network is encrypted, and only people with your network password have access to it.

WPA/WPA2 CRACKING

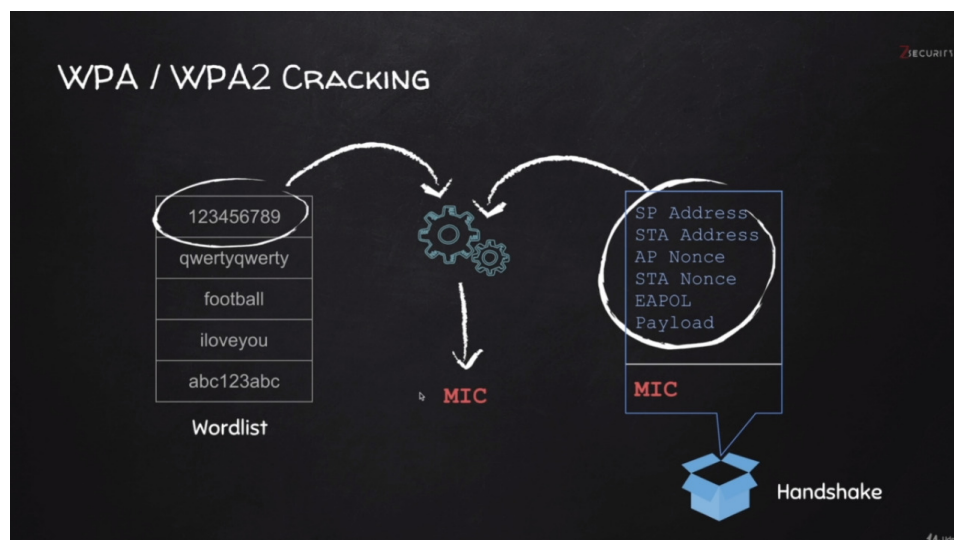


Figure 6.1: WPA/WPA2 Cracking model

6.2. CPS USED

What is CPS ?

A cyber-physical system (CPS) is a computer system in which a mechanism is controlled or monitored by computer-based algorithms.

Functions of CPS:

- Automatically control and monitor different types of industrial, scientific and business measures.
- Operate as a large-scale system and distribute tasks and roles.
- Require inter-disciplinary systems that are highly dependent on each other.
- Improve their performance eventually.
- Can self-adapt and change in progressively with real-time scenarios.
- Need for powerful decision systems.

Use cases of CPS:

Some examples of CPS include automobile frameworks, clinical monitoring, measure control systems, robotics systems, automatic pilot flying, traffic logistics systems and many more. Many instances of CPS surround us in our daily lives. At homes, we have vacuum cleaners, smart lighting systems, smart heating, ventilation, and air-conditioning systems. For transportation purposes we have cars, airplanes, motorized scooters, and electric bicycles. Existing systems like these represent the areas where we can expect to see huge advancement and improvement in future. For instance, while cars have been around for almost 300 years, several new features get added consequently and are now available in vehicle product lines.

Working of our CPS System:

3.1 The basic principle of operation:

A radar is an electromagnetic sensor that is used to detect and locate an object. The radar antenna transmits radio waves or microwaves that bounce off any object in their path. Due to this, we can easily determine the object in the radar range. Radio waves or microwaves are radiated out from the radar into free space. Some waves are intercepted by reflecting objects. These intercepted radio waves hit the target and are reflected in many different directions. Some of these waves can be directed back toward the radar, where they are received and amplified. If these waves are received again at their origin, then it means an object is in the propagation direction.

3.2 Hardware & Software used

- A. Ultrasonic Sensor : An ultrasonic sensor is a proximity sensor that is used to measure the distance of a target or object. It detects the object by transmitting ultrasonic waves and converts the reflected waves into an electrical signal. These sound waves travel faster than the speed of the sound that humans can hear.
- B. Servo motor : The servo motor is a simple DC motor that can be controlled for specific angular rotation with the help of additional servomechanism. This motor will only rotate as much as we want and then stop. The servo motor is a closed-loop mechanism that uses positional feedback to control the speed and position.

- C. Wifi Module : It is used to connect small devices like mobile phones using a short-range wireless connection to exchange files. It uses the 2.45GHz frequency band. The transfer rate of the data can vary up to 1Mbps and is in the range of 10 meters.
- D. Android Studio : Android Studio provides a unified environment where you can build apps for Android phones, tablets etc. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug
- E. Arduino IDE : The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++.
- F. Breadboard and Jumper Wires: A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a perfboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable.
- G. Buzzer: An arduino buzzer is also called a piezo buzzer. It is basically a tiny speaker that you can connect directly to an Arduino. You can make it sound a tone at a frequency you set. The buzzer produces sound based on the reverse of the piezoelectric effect.

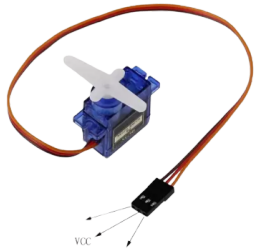


6.3. CIRCUIT CONNECTIONS


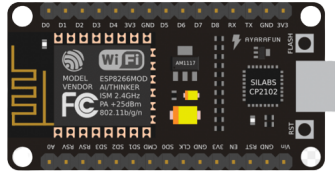
Arduino Pin Connections :

Voltage : Connect arduino 5V to power rail (long horizontal row) positive line in breadboard.

Ground : Connect arduino GND to power rail (long horizontal row) negative line in breadboard.

Digital Pins : The pins 4, 6, 8, 9, 10, and 12 are used as a digital input or output for the Arduino board.

S. No.	Name	Device Image	Pins	Connections
1.	Servo Motor Micro Servo SG-90		A servo motor has 3 pins: 1 red (5V), 1 brown or black (Ground), and 1 yellow (Control).	Connect brown to arduino ground, red to arduino Voltage and yellow to digital pin 12.
2.	Ultrasonic Sensor		An ultrasonic sensor has 4 pins : VCC, Trig (signal output pin), Echo (signal input pin) and GND.	Connect GND to Ground, Vcc to Voltage, Trig with ~10 and Echo with ~11.
3.	Buzzer		Buzzer has 2 pins : one Positive and other Negative.	Connect the buzzer positive node with digital pin 8 and negative pin via a resistor to ground.

4.	Bluetooth Module HC-05		The HC-05 Bluetooth module has 6 pins: enable, Vcc, Ground, Tx, Rx, and State, and Vcc can be powered from 3.3V – 6V.	Connect Vcc with Voltage, Ground with Ground, Tx with Arduino Rx and Rx with Arduino Tx.
5.	WiFi Module Node MCU ESP2866		A typical NodeMCU board has 30 pins . In this, 8 pins are related to power and 2 are reserved.	Connect Vcc with Voltage, Ground with Ground, Tx with Arduino Rx and Rx with Arduino Tx.

3.2 Step by Step Working :

- i) Hardware Connection : Connection of hardware resources like ultrasonic sensor, servo motor, jumper wires, Arduino and breadboard, bluetooth module, battery.
- ii) Software and Code installation : Installation of necessary softwares like Arduino IDE and uploading the codes in the Arduino.
- iii) Working : The system consists of a basic ultrasonic sensor placed upon a servo motor which rotates at a certain angle and speed. This ultrasonic sensor is connected to Arduino digital input output pins and servo motors also connected to digital input output pins. After uploading the code, the servo motor starts running from 0 to 360 degrees and again back to 0 degrees. The ultrasonic sensor also rotates along with the servo as it is mounted on the motor. The Radar is then connected with the Application with the Help of bluetooth. Then, there is the graphical representation of data from the Ultrasonic Sensor which is represented in a radar type display. If an ultrasonic sensor detects any object within its range, you can see the object graphical representation on the sonic app . The buzzer also creates sound in case the sensor detects any object within its range.

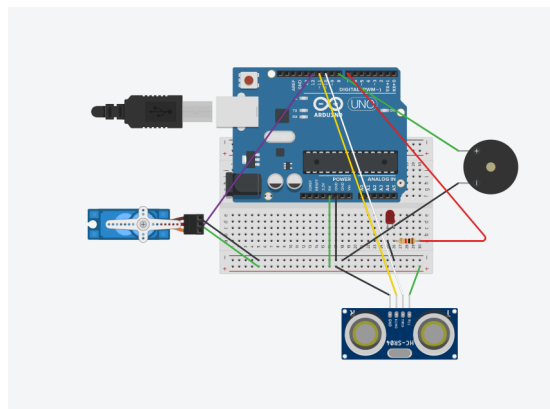


Fig 6.5 CPS System Circuit Diagram

Code Sketch For Arduino

```
// Includes the Servo library
#include <Servo.h>
// Defines Trig and Echo pins of the Ultrasonic Sensor
const int echoPin = 11, trigPin = 10, buzzerPin = 8, ledPin = 7;
// Variables for the duration and the distance
long duration;
int distance;
Servo myServo; // Creates a servo object for controlling the servo motor
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
  myServo.attach(12); // Defines on which pin is the servo motor attached
}
void loop() {
  // rotates the servo motor from 15 to 165 degrees
  for(int i=10; i<=165; i++){
    digitalWrite(ledPin, LOW);
    myServo.write(i);
    delay(30);
    distance = calculateDistance(); // Calls a function for calculating the distance
    measured by the Ultrasonic sensor for each degree
    if(distance < 40){
      digitalWrite(ledPin, HIGH);
      tone(buzzerPin, 1000, 100);
    }
    Serial.print(i); // Sends the current degree into the Serial Port
    Serial.print(","); // Sends addition character right next to the previous value
    needed later in the Processing IDE for indexing
    Serial.print(distance); // Sends the distance value into the Serial Port
    Serial.print("."); // Sends addition character right next to the previous value
    needed later in the Processing IDE for indexing
  }
  // Repeats the previous lines from 165 to 15 degrees
  for(int i=165; i>10; i--){
```

```

digitalWrite(ledPin,LOW);
myServo.write(i);
delay(30);
distance = calculateDistance();
if(distance<40){
digitalWrite(ledPin,HIGH);
tone(buzzerPin,1000,100);
}
Serial.print(i);
Serial.print(",");
Serial.print(distance);
Serial.print(".");
}
}
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave
travel time in microseconds
distance= duration*0.034/2;
return distance;
}

```

7. WORKING MODEL

7.1. SNIFFING

Sniffing is the act of intercepting and monitoring traffic on a network. This can be done using software that captures all data packets passing through a given network interface or by using hardware devices explicitly designed for this purpose. A sniffing attack occurs when an attacker uses a packet sniffer to intercept and read sensitive data passing through a network. Common targets for these attacks include unencrypted email messages, login credentials, and financial information.

Types of Sniffing Attacks

There are two primary sniffing attack types:

Passive Sniffing

In a passive sniffing attack, the hacker monitors traffic passing through a network without interfering in any way.

Active Sniffing

Active sniffing is a type of attack that involves sending crafted packets to one or more targets on a network to extract sensitive data. Active sniffing can also involve injecting malicious code into target systems that allows attackers to take control of them or steal sensitive information.

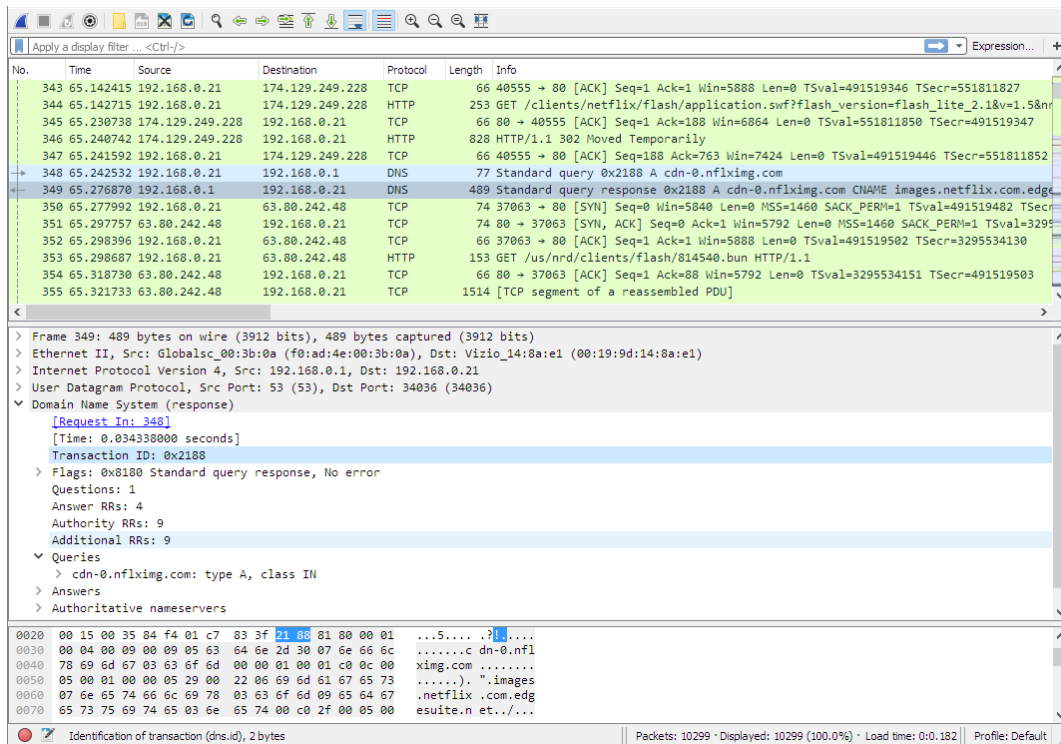


Fig 7.1: Sniffing data using Wireshark

7.2. DOS

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected. There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop. Popular flood attacks include:

- Buffer overflow attacks – the most common DoS attack. The concept is to send more traffic to a network address than the programmers have built the system to handle.
- ICMP flood – leverages misconfigured network devices by sending spoofed packets that ping every computer on the targeted network, instead of just one specific machine. This attack is also known as the smurf attack or ping of death.
- SYN flood – sends a request to connect to a server, but never completes the handshake. Continues until all open ports are saturated with requests and none are available for legitimate users to connect to.

Table N

Packet Features of DoS Activities.

Abused Protocol	Reply Packet Feature
TCP	SYN/ACK, RST ACK, etc.
UDP	ICMP Port Unreachable
ICMP	ECHO, Port Unreachable
DNS	DNS reply

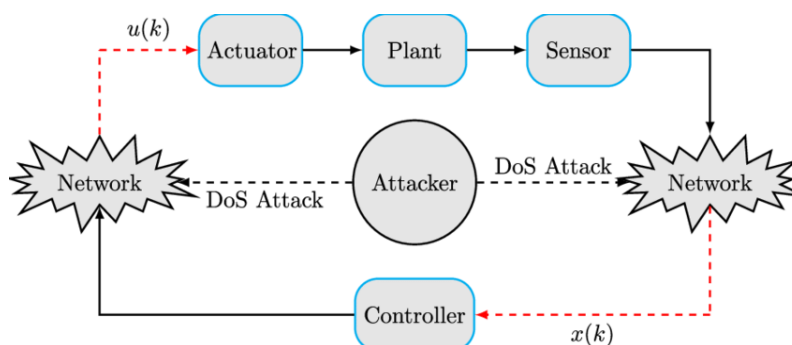


Fig 7.2. Working diagram of DoS attack

7.3. AUTHENTICATION

This type of attack targets and attempts to exploit the authentication process a web site uses to verify the identity of a user, service, or application.

The following types of attacks are considered authentication attacks:

Attack types	Attack description
Brute Force	Allows an attacker to guess a person's username, password, credit card number, or cryptographic key by using an automated process of trial and error.
Insufficient Authentication	Allows an attacker to access a web site containing sensitive content or functions without having to properly authenticate with the web site.
Weak Password Recovery Validation	Allows an attacker to access a web site that provides them with the ability to illegally obtain, change, or recover another user's password.

```

root@diwakar-ROG-Strix-G531GT-G531GT:/home/diwakar

Aircrack-ng 1.6

[00:00:54] 666400/4782969 keys tested (12365.39 k/s)
Time left: 5 minutes, 32 seconds                                13.93%

KEY FOUND! [ 123456789 ]

Master Key   : 1D D6 24 32 8B A8 A3 9A 95 48 85 46 A3 1A 91 D7
              94 4C 77 00 6E 88 33 8A A1 4A BD 8D F0 3A CF 2D

Transient Key : 56 3E 9C CC 26 AA 4B FB BD C2 E6 98 97 09 08 08
              AE FA DA FC A0 CC A3 49 71 E3 60 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC   : C5 A4 2B 5B 03 13 EE 8F 2B 8D E0 8C E4 67 37 CB

t@diwakar-ROG-Strix-G531GT-G531GT:/home/diwakar#

```

Fig 7.3: Authentication gaining using brute force approach

7.4. MITM

A man in the middle (MITM) attack is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway. The goal of an attack is to steal personal information, such as login credentials, account details and credit card numbers. Targets are typically the users of financial applications, SaaS businesses, e-commerce sites and other websites where logging in is required. Successful MITM execution has two distinct phases: interception and decryption.

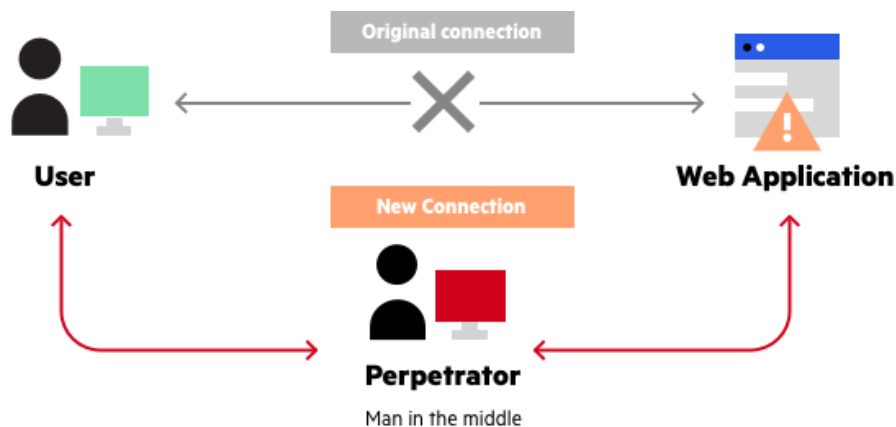


Fig 7.4: Man in the middle attack overview

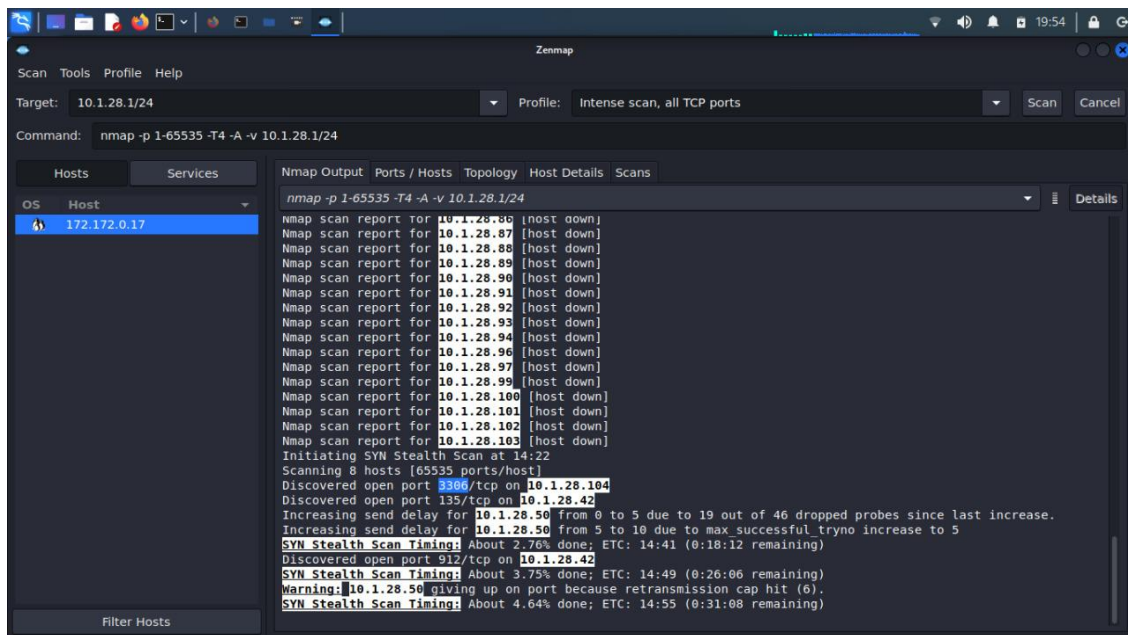


Fig 7.5: Using nmap Scanner for the man in the middle

8. FLOW DIAGRAM OF PROPOSED ATTACKS

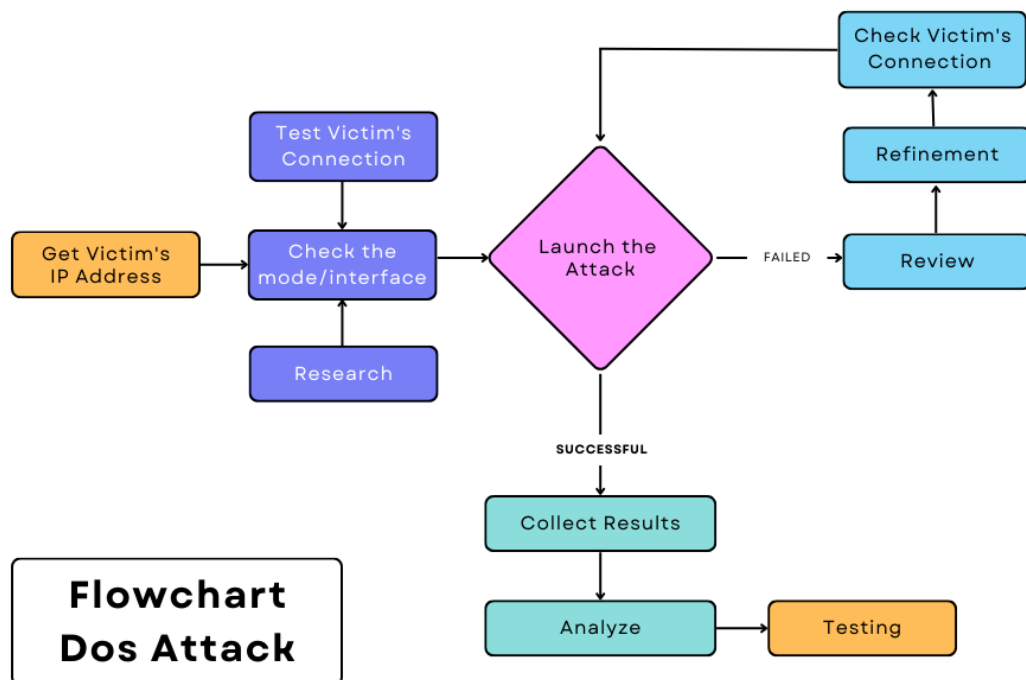


Fig 8.1: Flowchart for DoS attack

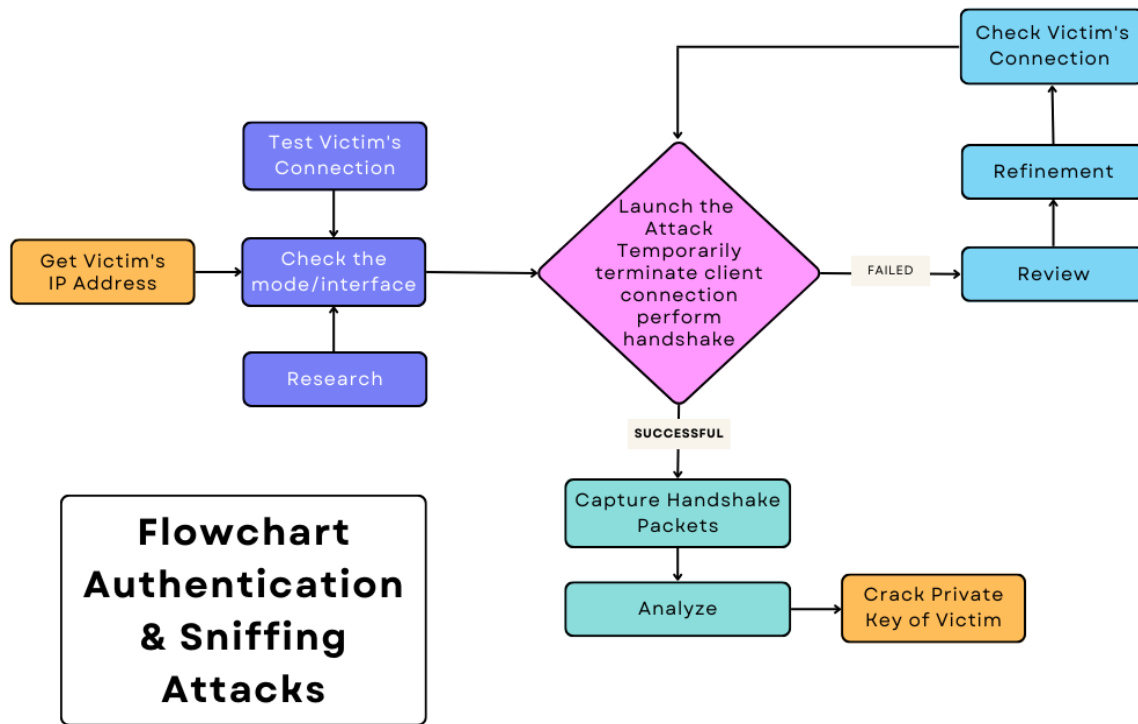


Fig 8.2: Flowchart for Brute Force Authentication break and Sniffing attacks

9. MODEL DESCRIPTION

The passive inference of scanning/probing efforts and Denial of Service (DoS) attacks directed at CPS resources is the main subject of this article. Targeted attacks have their roots in probing actions. Attackers often use these intelligence gathering strategies to characterize the scanned hosts and determine whether their services are open to exploitation. DoS attacks, on the other hand, try to overwhelm a service such that it becomes inaccessible. While attempts to scan CPS communication and control protocols may be an important sign that attackers are trying to identify the active services to conduct future crimes, DoS assaults may destroy CPS assets and do significant harm. After preprocessing darknet data using the previously suggested model, we present Algorithm, which makes use of both flow-based parameters and packet header information to infer CPS probing and DoS actions. It uses darknet flows, which are groups of packets with the same source IP address that are sent in succession. Each flow is carefully examined to see if any service ports in the related packets can be used to accurately identify CPS darknet activities. The algorithm determines that a particular flow is unrelated to CPS operations if no CPS ports were detected.

On the other hand, if a service port was discovered, Algorithm considers that flow to be suspect and proceeds to try to prove that suspicion. The technique logs the flow packet count (pkt cnt) in order to infer CPS probing actions. In order to achieve this, we take the packet count threshold. Please take note that the probing inference process would normally have defined and required a rate threshold (Rth). To allow the system to infer very low frequency, potentially covert probing actions, we do not require one here.

9.1. ALGORITHMS

CPS Probing & DoS Inference Algorithm.

```
1: Input: A set (F) of unique darknet flows (f),
2: Each flow f contains packet count (pkt_cnt) and rate (rate)
   SP: CPS Service Port
   Tw: Time window
   Pth: Packet threshold
   Rth: Rate threshold,
   Tn: Time of packet number n in a flow
   pkt: Packet
3: Output: CPS flag, CPS_flag
for Each f in F do
  while pkt in f do
    if pkt.contains() != SP then
      CPS_flag() ← 0
    end if
    if pkt.contains() = SP then
      CPS_flag() ← 1
    end if
12: end while
13:
14: pkt_cnt ← 0
15: T1 ← pkt_gettime()
16: Tf ← T1 + Tw
17: while pkt in f do
18: Tn = pkt_gettime()
19: if Tn < Tf then
20: pkt_cnt ← pkt_cnt + 1
21: end if
22: end while
23: rate ← pkt_cnt
   Tw
24: if pkt_cnt < Pth || rate < Rth then
25: CPS_flag() ← 0
26: end if
27: end for
```

10. IMPLEMENTATION RESULTS

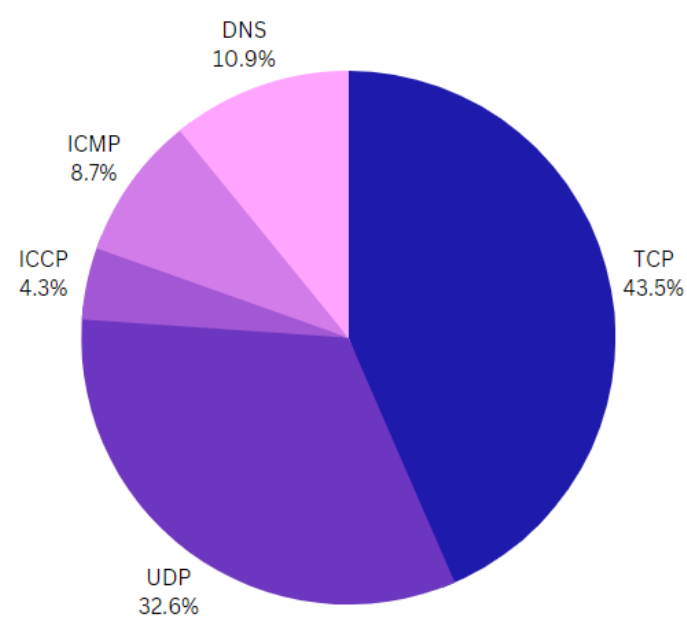


Fig. 10.1. The distribution of DoS Attacks targeting the CPS Protocols.

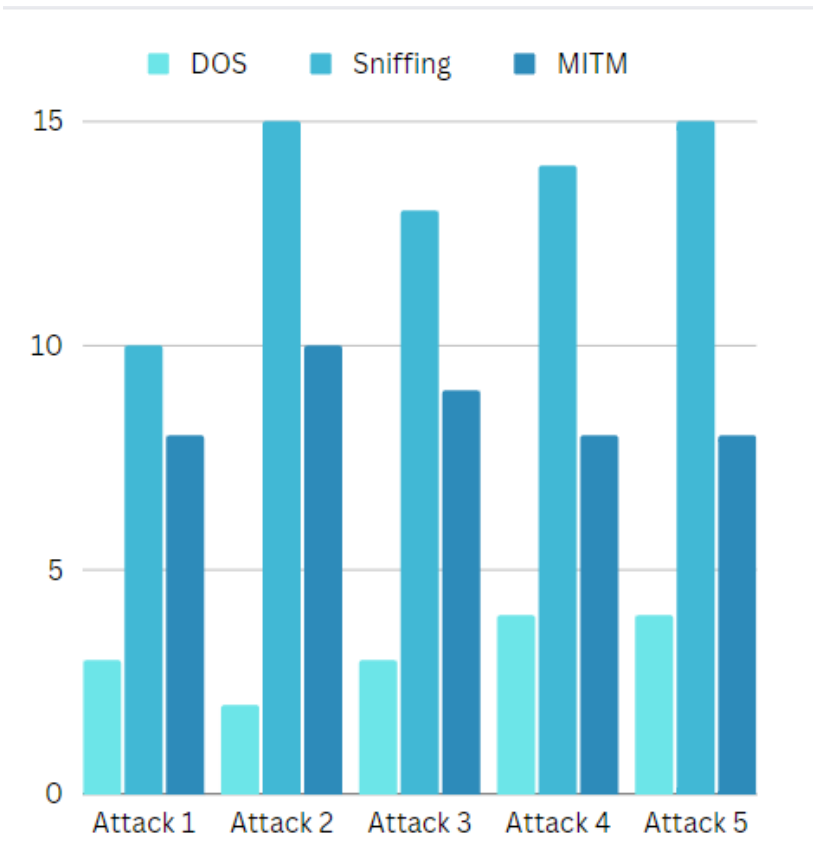


Fig. 10.2. Time taken by various types of attacks

11. CONCLUSION

Given the influence these technologies have on our modern societies and crucial infrastructure, research and development efforts addressing the security of Cyber-Physical Systems (CPS) are unquestionably of paramount importance. We tried to deduce, measure, characterize, and investigate ideas of CPS malice in this work. This article utilized passive measurements and analysis in a novel way to extract probing and denial-of-service (DoS) assaults targeting a variety of CPS communication and control protocols. The motivation for this was the dearth of hostile empirical data from operating CPS systems. In order to achieve this, this article provided a brand-new, incredibly useful probabilistic approach to clean darknet data by fingerprinting and then filtering out misconfiguration traffic. To recognise and decipher such CPS unsolicited traffic, an inference method together with characterisation modules was subsequently developed and put to use. We also confirmed the precision and distributed cluster performance of the proposed darknet preprocessing module through a number of experiments.

12. REFERENCES

- [1]Shin S, Kwon T, Jo G-Y, Park Y, Rhy H. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. *Ind Inform, IEEE Trans* in 2010.
- [2]Goldenberg N, Wool A. Accurate modeling of modbus/tcp for intrusion detection in scada systems. *Int J Crit Infrastruct Prot* 2013.
- [3]Premaratne UK, Samarabandu J, Sidhu TS, Beresh R, Tan J-C. An intrusion detection system for iec61850 automated substations. *Power Del, IEEE Trans* in 2010.
- [4]Düssel P, Gehl C, Laskov P, Bußer J-U, Störmann C, Kästner J. Cyber-critical infrastructure protection using real-time payload-based anomaly detection. In: *Critical information infrastructures security*. Springer; 2009.
- [5]Li X, Lu R, Liang X, Shen X, Chen J, Lin X. Smart community: an internet of things application. *Commun Magazine, IEEE* 2011.
- [6]Verba J, Milvich M. Idaho national laboratory supervisory control and data acquisition intrusion detection system (scada ids). In: *Technologies for homeland security, 2008 IEEE Conference on IEEE*.
- [7]Lee EA. The past, present and future of cyber-physical systems: a focus on models. *Sensors* 2015.
- [8]Rajkumar RR, Lee I, Sha L, Stankovic J. Cyber-physical systems: the next computing revolution. In: *Proceedings of the 47th design automation conference. ACM*; 2010. p. 731–6.
- [9]Kim K-D, Kumar PR. Cyber–physical systems: a perspective at the centennial. *Proc IEEE* 2012.
- [10] Conti M, Das SK, Bisdikian C, Kumar M, Ni LM, Passarella A, et al. Looking ahead in pervasive computing: challenges and opportunities in the era of cyber–physical convergence. *Pervasive Mob Comput* 2012.