Sub:	IoT&WSN					Sub Code: 18EC741
Date:	28/12/2021	Duration:	90 Minutes	Max Marks:	50	Sem / Sec: A,B,C,D

Answer any FIVE FULL Questions

1 Explain about vulnerabilities in WSN. [10]

Insecure web interface

Insecure web interface can lead to problems like SQL injection, cross-site scripting, cross-site request forgery, command injection, weak passwords.

Insecure network services

- It includes debugging services, diagnostics sevices, testing services.
- In secure network services can lead to problems like insecure storage of sensitive data (location, images, PII, PHI etc).

Ineffective authentication/authorization

Authentication vulnerabilities includes:

User enumeration weak passwords, Brute force attacks, weak session management.

An absence of transport layer encryption

If your IoT device sends private data over an insecure protocol, it is in cleartext, and as a result, anybody can read it. This only underscores the importance have secure communications protocols with these devices.

Privacy issues

If the data on the IoT device is not encrypted, and other individuals have access to it, this makes your data vulnerable to covert hijacking and theft.

Inadequate security features

- Many times, IoT device may not have the security features built into them as it is claimed to be.
- As a result, there are no encryption options or features to detect and monitor any potential Cyber-attacks.

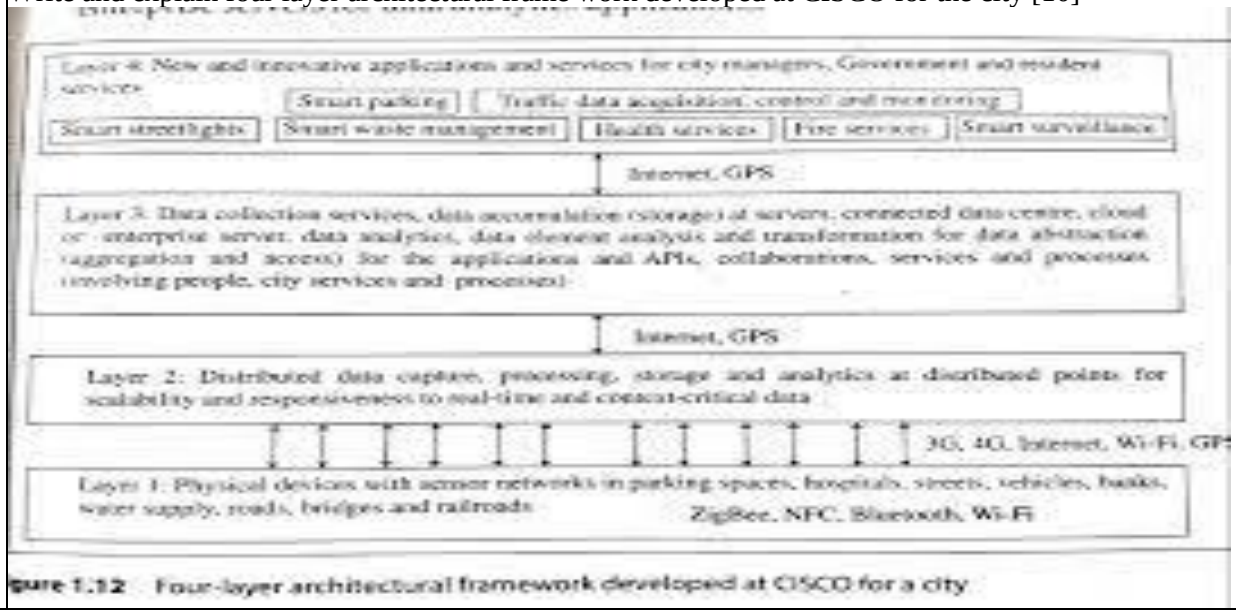
Insecure network traffic (free Wi-Fi (wireless) network)

- It means there's no special login or screening process to get on the network, which means you and anyone else can use it.
- It can lead to weak authentication of the client side, on the server side, Lack of encryption, Replay attacks, relying of "unknown" or "hard to understand protocols"

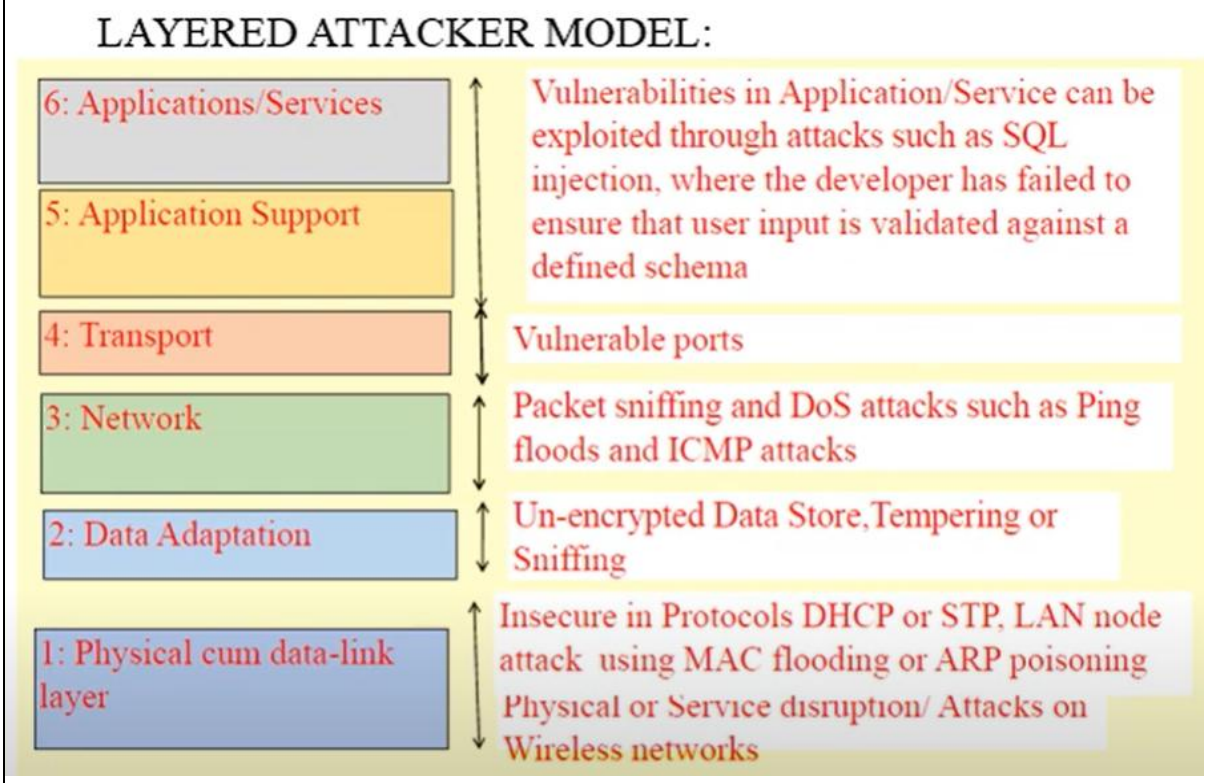
Insecure device firmware

- Firmware is used in managing interactions between the hardware and the higher-level software used to configure, manage and operate the device e.g. wireless routers,
- copiers, printers and cameras.
- Insecure device firmware can lead to the data can be stolen or modified i.e. sending data without encryption and devices taken control of for the purpose of attacking other devices

2 Write and explain four layer architectural frame work developed at CISCO for the city [10]



ure 1.12 Four-layer architectural framework developed at CISCO for a city

3	What is IoT security Tomography? Explain Layered attack model and possible attack at each layer [10] **Tomography: It means producing 3D picture of internal structure of an object by observation and recording.** **Network tomography** ➢ *It refers to the study of vulnerabilities and security aspects for network monitoring in complex system such as RFID, WSN or IoT networks.* ➢ *It also helps in observing each network section.* **Security Tomography:** It enables finding the attack vulnerable sections/subsections on observation for behaviors using finite number of objects or threats in a complex set of subsystems 
4	Explain security requirements and threat analysis in WSN [10] Security functional group contains five sets of functions which are required for ensuring security and privacy. Five functional components of security are defined in IOT are: 1) Identity management 2) Authentication 3) Authorization 4) Key exchange and management 5) Trust and reputation **Identity Management:** ➢ An object's identity should always be unique compared to the other objects from its family. ➢ Unique identity can be called core identity, as an object can also have several temporary identities. ➢ Devices must establish their identity before they can access gateways and upstream services and apps. ➢ An object should knows the identity of its owner if it exists. **Secure Authentication/Authorization** ➢ IoT devices should be authenticated with strong usernames IDs and password before being allowed to communicate with other IoT devices on the network. ➢ Authentication token/session should always be unique to each user along with user id, app id and device id. ➢ System's credentials, application, device and server should be authenticated as spoofed devices could transmit malicious data to other devices and can implement a denial- of-service attack on the IoT network **Key exchange and management**

- Public key cryptography also known as asymmetric encryption keeps the data safe and secure during transmissions.
- In a public key encryption system, any person can encrypt a message using the receiver's public key. That encrypted message can only be decrypted with the receiver's private key.
- **Certification Authority (CA):** This is effectively done by a issuing a digital certificate to confirm the authenticity of the device, firmware / software updates, and facilitate encrypted communications.

Trust and reputation

Trusted IoT Device:

The trusted IoT devices should be able to communicate with the intended hosting services only.

And the firmware / software should be frequently updated.

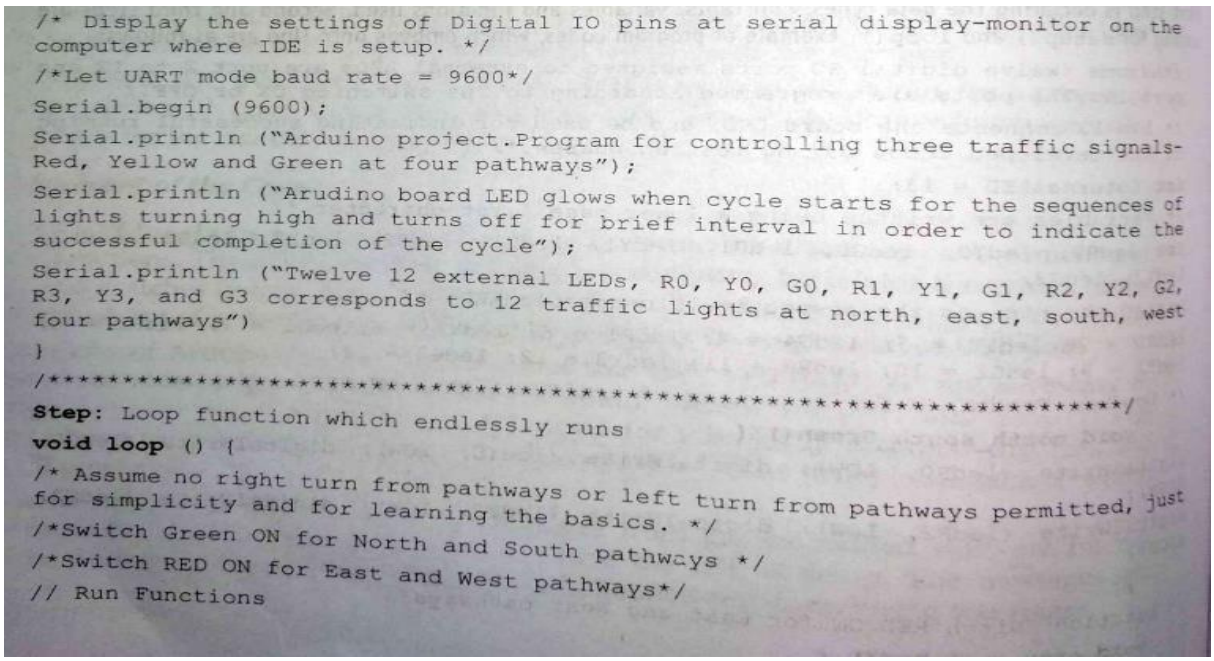
The Trusted IoT Master:

A trusted master must provide a secure communication with dependent sensor devices, and issue firmware/software updates to those devices and ensures that the code is authentic, unmodified and non-malicious

5 Explain about Arduino platform and write a program for displaying traffic light. [10]

```
testing phases.*/                    successful running
int internalLED = 13;
/* Variables are written using a lower case first character */
int ledR0, ledY0, ledG0, ledR1, ledY1, ledG1, ledR2, ledY2, ledG2, ledR3,
ledY3, ledG3;
Assign the pins to the respectively connected LEDs */
ledR0 = 2; ledY0 = 3; ledG4 = 4; ledR1 = 5; ledY1 = 6;ledG1 = 7; ledR2 = 8;
ledY2 = 9; ledG2 = 10; ledR3 = 11; ledY3 = 12; ledG3 = 14;
/* Declare Functions for sequences of traffic lights ON-OFF as follows: */
   void north_south_Green() {
digitalWrite (ledR0, LOW); digitalWrite (ledY0, LOW); digitalWrite (ledG0,
HIGH);
digitalWrite (ledR2, LOW); digitalWrite (ledY2, LOW); digitalWrite (ledG2,
HIGH);
   };
/* Function Switch RED ON for East and West pathways*/
  void east_west_Red() {
```

```
digitalWrite (ledR1, HIGH); digitalWrite (ledY1, LOW);digitalWrite (ledG1,
LOW);
digitalWrite (ledR3, HIGH); digitalWrite (ledY3, LOW); digitalWrite (ledG3,
LOW);
     };
/*************************************************************/
void setup ( ) {
/* GPIO pins 2 to 12 and 14 are thus assigned port numbers corresponding to 12
external LEDs, R0, Y0, G0, R1, Y1, G1, R2, Y2, G2, R3, Y3, and G3.*/
/* Assign mode of each pin as output */
pinmode (ledR0, OUTPUT); // Constants are written in Upper Cases
pinmode (ledY0, OUTPUT);
.
pinmode (ledY3, OUTPUT);
pinmode (ledG3, OUTPUT);
/* Let Pin 13 be used for indicating successful running of the developed codes
during testing phases. Initialise internal Port 13 Digital IO Pin LED for
test.*/
```



```
/* Display the settings of Digital IO pins at serial display-monitor on the
computer where IDE is setup. */
/*Let UART mode baud rate = 9600*/
Serial.begin (9600);
Serial.println ("Arduino project.Program for controlling three traffic signals-
Red, Yellow and Green at four pathways");
Serial.println ("Arudino board LED glows when cycle starts for the sequences of
lights turning high and turns off for brief interval in order to indicate the
successful completion of the cycle");
Serial.println ("Twelve 12 external LEDs, R0, Y0, G0, R1, Y1, G1, R2, Y2, G2,
R3, Y3, and G3 corresponds to 12 traffic lights at north, east, south, west
four pathways")
}
/*************************************************************************/
Step: Loop function which endlessly runs
void loop () {
/* Assume no right turn from pathways or left turn from pathways permitted, just
for simplicity and for learning the basics. */
/*Switch Green ON for North and South pathways */
/*Switch RED ON for East and West pathways*/
// Run Functions
```

6 Explain how the data is read from sensors and devices [10]

1) Sensors senses the analog data and send the analog data to 10 bit ADC (Analog to digital converter).
2) ADC send the 10 bit parallel data to PISO converter.
3) Parallel IN serial out (PISO) converter converts parallel data to serial data.
4) This serial output connects to SPI (Serial Peripheral Interface) Pin of Arduino board.
5) For example let sensor senses the temperature of 100 degree.
6) ADC converts 100 degree to binary 1111111111,which is send to PISO then to SPI of arduino

7 Explain PAMAS Protocol with necessary figures.

PAMAS: Power Aware Multi-Access with Signaling

a) **Signaling channel/control channel:** For sending RTS/CTS messages and busy tone messages and it enables the nodes to determine that for how long they can power off themselves.

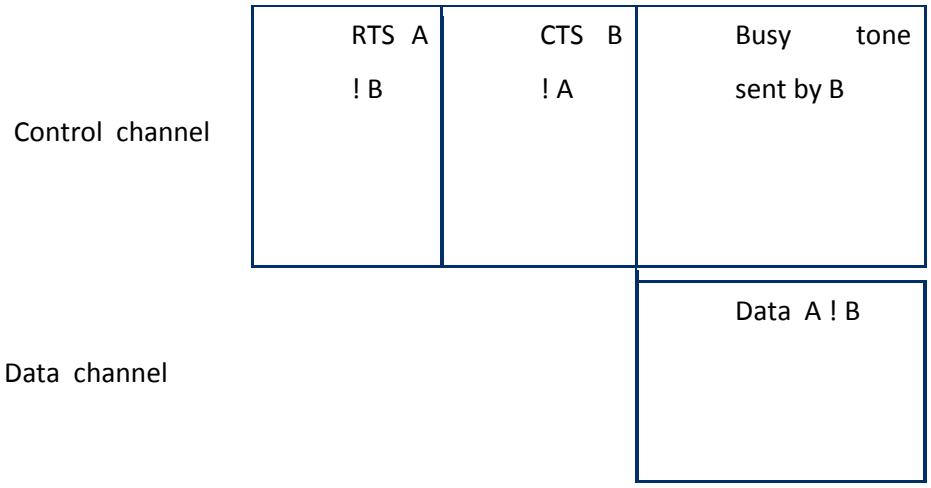
b) **Data channel:** Actual data transmission happens.

Procedure

a) Node A does not sense channel and transmits RTS on control channel. If node B receives RTS, it sends CTS on control channel if it can receive the data and does not know about ongoing transmissions

b) B then sends busy tone as it starts to receive data.

c) If node A fails to receive the CTS signal then it enters backoff scheme using binary exponential method.



PAMAS conserves the battery power by selectively powering off the nodes that are neither receiving nor transmitting and during overhearing.

Conditions that force nodes to turn off the power:

Case1) The node has no packet to transmit,

Case2) The node has packet to transmit

Case1) The node has no packet to transmit:

- Here the node should go to sleeping mode, turn off the power and wake up exactly when outgoing transmission ends so that it can receive the packet.
- ***Use a probing protocol on control channel***
- **Probing protocol defines the length of outgoing transmission (i.e. length of ongoing packet).**
- ***Use a probing protocol on control channel***
- **Probing protocol defines the length of outgoing transmission (i.e. length of ongoing packet).**
- Here the node sends a ***t_probe (l/2,l)*** packet where ***'l' is the length of packet.***

- Any transmitter node who finishes the transmission with in this time interval ***'(l/2,l)'*** answers with ***t_probe_response (t)*** packet indicating the time ***'t'*** where transmission ends.
- ***If the node manages to receives this response packet it knows where exactly this transmission ends and can wake up fast***

Case2) The node wakes up during ongoing transmission and has packet to transmit.

- Here the node has to take care of ongoing transmission as well as ongoing reception.
- **It uses a probing protocol to define the time for next wakeup.**
- **Probing protocol that defines the length of ongoing packets**