Network Security

Professor Gaurav S. Kasbekar
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Week - 12

Lecture - 72

Security of the Internet of Things (IoT), Hardware Security: Part 2

Hello, recall that in the previous lecture, we discussed the applications of the Internet of Things and the structure of a typical IoT device. We will continue our discussion on the security of the Internet of Things and hardware security in this lecture. What are the challenges in the networking of IoT nodes? Existing internet protocols were designed for powerful devices such as desktop computers, laptops, and smartphones. They were not designed for IoT nodes, which are resource-constrained.

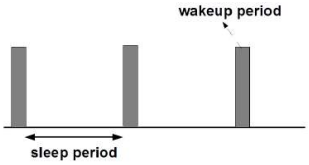
IoT nodes such as sensors and actuators are typically designed for low cost. In an application such as, say, precision agriculture, there are a large number of IoT nodes, so they have to be low cost; otherwise, the overall cost of the system will be very high. IoT nodes are also designed for low power consumption. So, IoT nodes have limited available power. They are often battery operated and they need to last for a long time, such as several years, without the battery having to be changed.

And they also have limited memory. Possibly just a few hundred kilobytes or a few megabytes. And they have limited processing resources as well. Another challenge is that IoT nodes are disabled for long time intervals called sleep periods to save energy. In this figure, the horizontal axis is the time axis, and these shaded bars, they show the periods in which IoT nodes are active, and the intervals between the shaded bars are the intervals in which the IoT nodes are in a low power state called sleep state.

They switch to this sleep state to conserve power. So, in the sleep state, their radio transceivers are switched off and the processor goes into a low power state. So, this is done to conserve power. This introduces some challenges. One challenge is that the local clocks of different nodes have to be synchronized. So, they know when the wake up periods and the sleep periods start.

So, they all have to wake up at the same time and sleep at the same time. Another challenge is that suppose node A sends a packet to node B when node B is in sleep state, in that case, node B will not receive the packet because its radio transceiver is switched off. So, a node can only receive traffic when it is in the active state. Hence, protocols must take this into consideration. Other challenges in the networking of IoT nodes are that IoT nodes have different data traffic characteristics and quality of service requirements from those of traditional devices connected to the internet.

- So IoT nodes:
  - ❑ Have limited available power (often battery operated)
  - ❑ Have limited memory
  - ❑ Have limited processing resources
  - ❑ Are disabled for long time intervals (sleep periods) to save energy



So, data traffic characteristics are the kind of data traffic that these IoT nodes emit. For example, whether they emit steady traffic or traffic at sporadic traffic package sent at random intervals. So, what kind of traffic do they emit? IoT nodes have different data traffic characteristics from those of traditional devices. And also the quality of service requirements, that is, the throughput, delay, jitter, and so on, how stringent these requirements are—that determines the quality of service requirements.

So, the QoS requirements of IoT nodes are different from those of traditional devices connected to the internet. Some examples will make this clearer. In the case of IoT nodes, network access often needs to be provided to an extremely large number of IoT devices. For example, several sensors in each smart home, a lot of smart meters, and so on. In a typical cell, in the case of human-to-human communication, there may be just, say, a few tens of humans, but in the case of IoT devices, in the same area, there'll be hundreds or thousands of IoT nodes.

So, that's one defining characteristic of IoT devices. IoT nodes may transmit small bursts of data periodically or randomly. For example, these bursts of data may contain the soil moisture or temperature that is read by sensors in the case of precision agriculture. Notice that this kind of traffic is different from that of human-to-human communication, where a human makes a call and then is active on the call for several minutes or tens of minutes. And when the human is not in a call, then no traffic is emitted by the human.

So, in contrast, IoT nodes may transmit small bursts of data periodically or randomly. They may also have stringent latency requirements or they may need priority access to communicate alarms, for example, in healthcare and security applications. If a burglar is detected in the case of a security application or some event such as a heart attack being detected that arises in a healthcare application, so that message has to be sent with a very stringent latency requirement. IoT devices may also require highly reliable communication, for example, in remote payment systems. Since financial transactions are involved, the communication has to be highly reliable.

They may also require high throughput, for example, in a video surveillance application. There may be some cameras which record videos in a certain area for security purposes, so that video has to be streamed to a server, which requires high throughput. We now discuss how IoT nodes connect to the rest of the network, that is, we discuss IoT node access methods. The access network connects IoT nodes to the infrastructure, such as the internet. The access network can be either wired or wireless.

If it is wired, then it may use different technologies such as cable, digital subscriber line, or optical, which we discussed earlier. Or it may be wireless, and it may use different wireless technologies, which we will discuss now. In the case of wired access, the advantages are that wired access can provide high reliability because there is a dedicated cable connecting the IoT node to the network. So, the communication is highly reliable. Also, the cable can provide high data rates and low latency or delay.

But the disadvantages are that wired access is typically expensive. We need to connect a separate cable to every IoT device, which is expensive. It is also difficult to scale for the same reason. If there are hundreds of IoT devices in a small area, then it is difficult to connect a separate cable to each device. We also cannot support mobile IoT nodes with wired connections.

Wireless access comes in different types. One is short-range wireless access, which operates in an area of, say, a few tens of meters or hundreds of meters, such as Wi-Fi and IEEE 802.15.4. This IEEE 802.15.4 is the standard on which ZigBee is based. Wireless access can also be wide-area or cellular, such as LTE-Advanced 4G networks and 5G networks. Wireless access can also take the form of low-power wide-area networks.

Examples of these are LoRa and Sigfox. These are popular wide-area networks. They operate over large areas, such as several tens of kilometers, and consume very little power. Hence, the name low-power wide-area networks. Let’s discuss these in more detail.

Short-range wireless technologies such as Wi-Fi and IEEE 802.15.4. The advantages are that they are inexpensive, scalable, and low-power. In particular, IEEE 802.15.4 is a very low-power technology. The disadvantages are that they provide low data rates. Wi-Fi provides very high data rates, but 802.15.4 provides low data rates.

They also, Wi-Fi as well as 802.15.4, suffer from interference because they use unlicensed bands, which we discussed earlier. So, they can face interference from other networks which operate on the same frequency bands. There is also a lack of universal coverage. Only if there is a Wi-Fi hotspot in an area, for example, we can get Wi-Fi connectivity. It's not available everywhere, and even if it is available, then one may not have the login credentials to use the network.

Hence, there is a lack of universal coverage. We now discuss wide-area cellular networks, such as LTE Advanced and 5G. The advantages are that they provide ubiquitous coverage and mobility, and there is no question of interference from other networks because these use licensed bands. So, the network operator licenses spectrum from the regulator, and that is for the exclusive use of that operator. So, there is no interference from other networks.

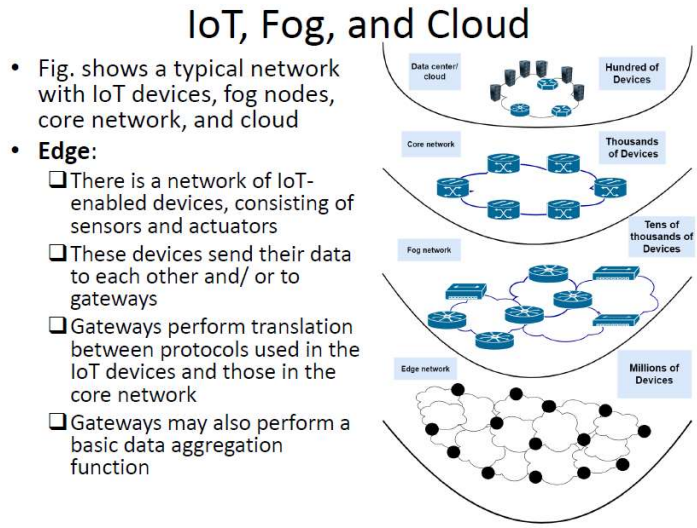
Disadvantages are that due to the high demand for human-to-human communication services, such as voice and data, only a limited amount of radio spectrum is available with cellular operators to support IoT node communication. So, the available spectrum has to be shared between human-to-human communication services and IoT services. We now discuss the architecture of a typical IoT network. There are different components, such as IoT, fog, and cloud. This shows a typical network with IoT devices at the bottom over here.

And fog nodes are over here. And there are data centers or clouds which are here. The core network connects fog nodes to the cloud or data center. So, this architecture works as follows. These IoT nodes are different sensors or actuators.

They may collect some information, such as soil moisture, temperature, and so on. Or they may take some actions, such as opening a valve, closing a valve, and so on. These are at the edge of the network, and these sensors send their collected information to some gateways, which are located here. So, these gateways do some local processing. For example, they may aggregate the data from a large number of sensors.

So that happens at these fog nodes, which are connected to the gateways. These fog nodes do some local processing, and they may take some actions as well. The advantage of doing some processing and storing some data in the fog nodes is that there is low latency

compared to sending the data all the way to the cloud, which is usually far away from the IoT devices. So, sensors collect some data, these fog nodes do some local processing and take actions, and then these fog nodes communicate the data to the cloud or data center over the core network. The core network consists of a large number of routers and communication links connecting them.



Then, any detailed or intensive data processing happens in the cloud or data center, and it also provides a lot of storage for the data collected by IoT devices. So, let's discuss different components in more detail. We start with the edge network. There is a network of IoT-enabled devices consisting of sensors and actuators. These devices send their data to each other and to gateways.

Gateways collect data from a large number of sensors. Gateways perform translation between protocols used in IoT devices and those in the core network. For example, IoT devices may use protocols such as BLE, which stands for Bluetooth Low Energy. So, a gateway may perform translation between BLE and some protocol used in the core network. For example, TCP/IP-based protocols.

So, gateways perform such translation between different protocols used in IoT devices and those used in the core network. Gateways may also perform basic data aggregation functions. For example, suppose sensors are deployed in a farm and we only require the average temperature in the farm for a certain application. In that case, gateways may perform some data aggregation. They may find out the average of the data collected by all the sensors, and only this average value is sent further on towards the cloud. So, such

gateways perform basic data aggregation functions, such as averaging, addition, multiplication, and so on, of the collected readings.

Then, we discuss fog nodes. So, what's the reason for having fog nodes and doing some processing and storing some data close to the IoT devices? In many IoT deployments, massive amounts of data such as several terabytes a day may be generated by a distributed network of sensors. For example, some applications such as smart meters may generate a lot of data. This volume of data generated by IoT devices can be very large and it can easily overrun the capabilities of the cloud.

So, if this collected data is directly sent to the cloud without any local processing, then the cloud may get overwhelmed even though it has a lot of processing and storage abilities. So, the solution to this challenge is to distribute data management throughout the IoT system as close to the edge of the IP network as possible. We need to introduce some processing and storage capabilities close to the IoT devices, that is, in the fog network. So, the best known embodiment of edge services in IoT is fog computing. Any device with computing storage and network connectivity can be a fog node.

Some examples are industrial controller switches, routers, embedded servers, IoT gateways, and so on and so forth. So, all these devices reside in this fog network and they do some processing and store some data. What are the advantages of analyzing IoT data close to where it is collected? One important advantage is that the latency is minimized. Suppose some actions have to be taken on real time based on the data collected by these sensors.

If the data is sent all the way to the cloud and then the actions are taken, then that may incur a latency of several tens of milliseconds or even hundreds of milliseconds. So, the actions may be taken too late for the task to be successfully performed. In contrast, if the collected data is sent to a fog node and actions are taken at the fog node, which is close to the IoT devices, then the actions can be performed in quick time. So, latency is reduced by doing the processing and storage close to the IoT devices in contrast to sending it all the way to the cloud. Another advantage is gigabytes of network traffic is offloaded from the core network.

So, data that is processed in the fog nodes, that data does not have to be sent to the cloud. Only some limited amount of data that is extracted from the aggregate data sent by the sensors is sent to the cloud. So, gigabytes of network traffic is offloaded from the core network. Data which would have been sent over the core network; if the fog nodes were

not there, that data is not sent over the core network. So, it is offloaded from the core network, bandwidth is saved in that way.

Another advantage is that sensitive data is kept inside the local network. If there is some confidential data or private data of users that is collected by sensors, that data remains restricted to the local network. It is not sent to the cloud. So that helps in addressing privacy concerns. That's our discussion of the fog network.

We now discuss the core network. It is also known as the backbone network. And it connects geographically dispersed fog networks to the data center or cloud. It typically uses very high-performance routers and high-capacity transmission lines. Since it collects data from a large number of fog nodes and IoT networks, the bandwidth needs to be very high, and these routers also need to be high-performance.

Cloud: we have discussed in detail in the earlier part of the course. It provides storage and processing capabilities for the massive amounts of data that originate from IoT devices at the edge. That concludes our discussion of the architecture of an IoT network. Next, we discuss IoT security objectives. One objective is restricting access to the IoT network.

If intruders access the IoT network, then they may launch some attacks such as installing malware on the IoT devices and so on. So, how can we restrict access to IoT networks? One way is to use unidirectional gateways. For example, if sensors collect some data in the IoT network which needs to be sent to users, then we can use unidirectional gateways which allow data to be transmitted from the sensors to the users but not the other way around. So, users are not allowed to send any data to the sensors.

This prevents malicious users from sending harmful files to the sensors. We can also use firewalls, which we discussed earlier. We can also enforce authentication mechanisms and credentials for users of the IoT network. Only authenticated and legitimate users are allowed to access sensors in the IoT network. That's another way of restricting access to the IoT network.

Another security objective is restricting physical access to IoT networks and components. So, we can use a combination of physical access controls, such as locks, card readers where users need to swipe their cards to gain access, and/or guards who can restrict physical access to the IoT network and components. Another security objective is protecting individual IoT components from exploitation. One way to do this is to deploy security patches in an expeditious manner. These can defend against viruses or worms, for example.

Also we can disable all unused ports and services so that attacks cannot be launched on these ports and services which are disabled. Another means is we restrict IoT user privileges to only those that are required for each person's role. So, a user is not able to perform functions that are not required for that user. The user privileges are restricted to only those that are required for that person's role. Another technique is using antivirus software and file integrity checking where feasible.

Another IoT security objective is preventing unauthorized modification of data. This includes data in transit and at rest. For this, we can use message integrity techniques, which we discussed earlier. Another method is hashing files. Consider some data that is at rest.

We can find the hash of that data and store it. Later on, if the data has been modified, then by checking the hash of the data, we can detect that there has been a modification. So, this way, we can prevent unauthorized modification of data. Then, another IoT security objective is detecting security events and incidents. Security events must be detected early enough to break the attack chain before attackers achieve their objectives.

This includes the capability to detect failed IoT components, unavailable services, and exhausted resources. Another IoT security objective is maintaining functionality during adverse conditions. One way to do this is to design IoT systems so that each critical component has a redundant counterpart or a backup. So, if one component goes down or runs out of battery and so on, then it has a backup which can perform the function of that device. Another way is if a component fails, it should fail in a manner that does not generate unnecessary traffic on IoT or other networks.

For example, it should not send a large number of error messages, which block the other devices. It should also not cause another problem elsewhere. So, if a component fails, it should not cause any trouble to the other devices or create problems elsewhere. IoT systems should also allow for graceful degradation, such as moving from normal operation with full automation to emergency operation with operators more involved and less automation, to manual operation with no automation. So, there are these three stages.

During normal operation, there is full automation. If there is some event which happens, then the system transits to the second state where there is emergency operation with operators more involved and less automation. And if another event happens, then there is transition to the third state where there is manual operation with no automation whatsoever. Another IoT security objective is restoring the system after an incident. So, incidents are inevitable, and an incident response plan is essential.

The IoT system should be recovered quickly after an incident has occurred. Incidents, as we discussed earlier, they are inevitable, and in case some security incident happens, then the IoT system should be recovered quickly after the incident occurs. In summary, we discussed the challenges in networking of IoT nodes and we discussed different IoT security objectives. We'll continue our discussion on the security of IoT networks and hardware security in the next lecture. Thank you.