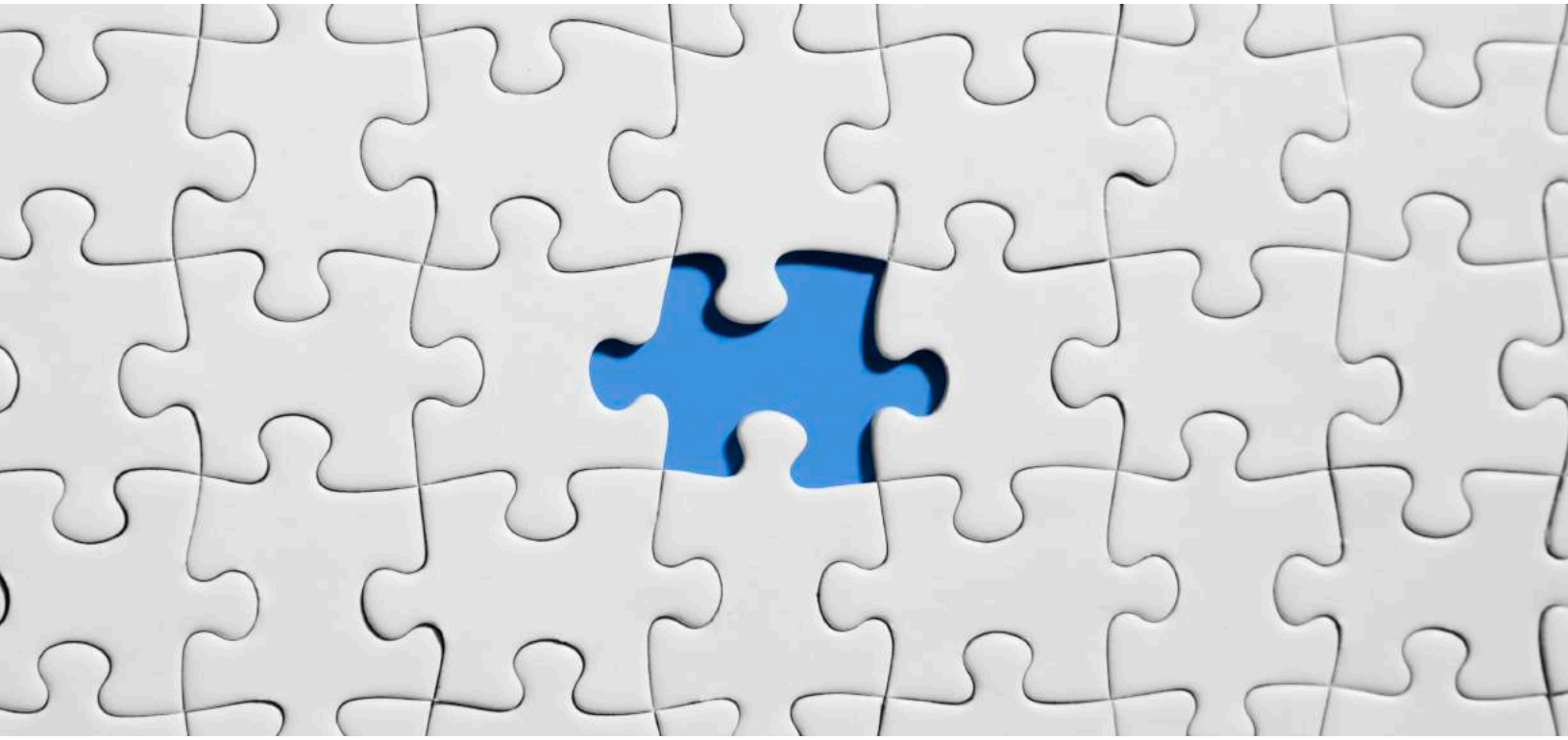
DELL EMC

IOT SECURITY: CHALLENGES, SOLUTIONS & FUTURE PROSPECTS



Mikhail Gloukhovtsev

Sr. Solutions Architect

Digital Solutions, Cloud & IoT

Orange Business Services

Mikhail.Gloukhovtsev@orange.com

DELL EMC PROVEN PROFESSIONAL

Knowledge Sharing Article

Table of Contents

Disclaimer: The views, processes or methodologies published in this article are those of the author. They do not necessarily reflect Dell EMC's views, processes or methodologies.

1. Introduction

1.1 What Is IoT?

The concept of the Internet of Things (IoT) was introduced by Kevin Ashton, a co-founder of the Auto-ID Center at MIT, in 1998.[1] The vision is that objects ("things") are connected to each other and thereby they create IoT in which each object has its distinct identity and can communicate with other objects. IoT objects can vary dramatically in size from a small wearable device to a cruise ship. IoT transforms ordinary products such as cars, buildings, and machines into smart, connected objects that can communicate with people, applications and each other.

There are various definitions of IoT. The International Telecommunication Union (ITU) defined the term Internet of Things as "Internet of Things will connect the world's objects in both a sensory and intelligent manner".[2] In 2014, the Joint Technical Committee of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) defined IoT as "an infrastructure of interconnected objects, people, systems and information resources together with intelligent services to allow them to process information of the physical and the virtual world and react".[3] At the IoT reception layer (Section 2.1.2), sensors placed within devices, objects, and machinery collect, measure, and record information about the physical environment, such as temperature, humidity, gas pressure, and motion. This information can be read, integrated and analyzed at higher IoT layers.

NIST uses two acronyms, IoT and NoT (Network of Things).[4] IoT is considered a subset of NoT, since IoT has its "things" connected to the Internet. In contrast, some types of NoT use only Local Area Networks (LAN), with none of their "things" connected to the Internet.

Figure 1: Key Business Drivers for IoT Development

The IoT growth is driven by business needs as part of enterprise digital transformation (Fig. 1). According to Machina Research,[5] the total number of IoT connections will grow from six billion in 2015 to 27 billion by 2025. It means a compound annual growth rate (CAGR) of 16%. In terms of market growth, the Berg Insight report[5] predicts an increase of the global third party IoT platform market from €610m in 2015 to €3.05bn in 2021.

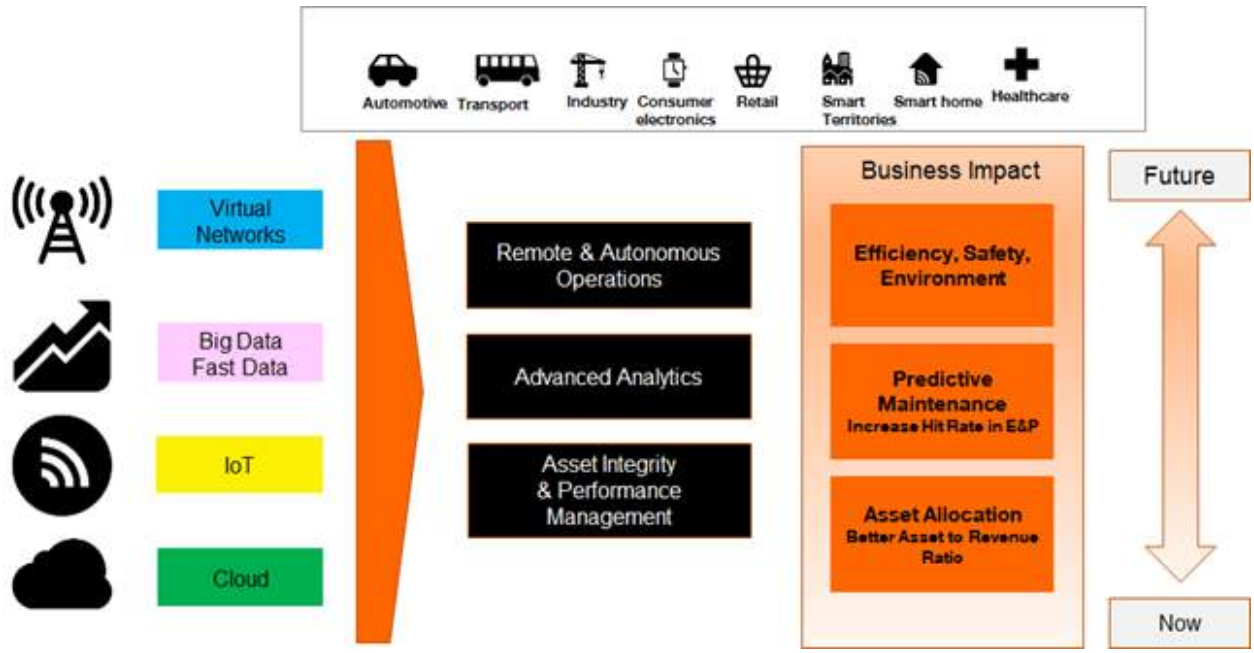


Figure 2: IoT Connecting Technologies and Disparate Industries

IoT solutions not only involve various technology domains such as mobile communications, cloud, data, security, telecommunications, and networking but they also lead to cross-industrial use of data (for example, data generated in smart home and industrial applications is used in the automotive domain) (Fig. 2). This opens a possibility for establishing business partnerships between horizontal industries, such as telecommunication operators, and vertical industries, such as car manufacturers, as new business models. IoT-enabled digital transformation of business is much more than just using connected objects – it makes it possible to develop innovative business models that were impossible before.

1.2 Security Role in the IoT Development

As discussed above, IoT is growing fast across various industry verticals along with increases in the number of interconnected devices and variety of IoT applications. However, IoT technologies are not mature yet and there are many challenges to overcome. Security is the most significant of them. There are millions of connected devices and billions of sensors and their numbers are growing. All of them need secure and reliable connectivity. Hence, well-designed security IoT architectures are required by companies and organizations adopting IoT technologies.

Indeed, the IoT threat landscape is large and growing: the attack surface is very large, as any IoT device could be a possible attack target. Some IoT devices are located in untrusted areas and attackers can gain physical access to them and even get control of the device. Many IoT devices do not meet security best practices requirements such as least-privileged or role-based access. For example, many smart-home IoT devices such as TVs, webcams, home thermostats, remote power outlets, sprinkler controllers, home alarms, door locks, and garage door openers communicate over the network without any form of encryption and do not offer the user the possibility to enable strong passwords. IoT devices are resource-constrained and are designed to consume little power while at the same time providing all required functionality at a reasonable cost. As a result, security is an after-thought, often placed at the bottom of the priority list in the development lifecycle.

IoT attack vectors can target devices, gateways, SIM/cell, transceivers, and wearables and can take advantage of weak passwords, lack of encryption, backdoors, etc.

The wide variety of IoT-specific operating systems, firmware versions (Section 2.1.2, Table 1), and custom configurations makes development of general IoT security solutions difficult. Monitoring and patching the various IoT OSes is a tremendous challenge. Furthermore, IoT

security solutions should be extremely scalable to apply to an exponentially increasing number of various IoT devices. A growing variety of IoT applications creates new security challenges. In addition to traditional security domains such as cryptography, secure communication, and privacy assurances, IoT security also focuses on trust/identity management (Section 5.1), data confidentiality (see Section 5.2), privacy protection (Section 5.3), etc.

This article considers IoT security challenges, security requirements for IoT architecture, current security solutions and new evolving technologies. I hope my article will help the readers in selecting secure IoT technologies for their businesses.

2. IoT Security and IoT Architectures

2.1 IoT Architectures

As IoT uses a very broad range of various technologies (Table 1),[6] it is not possible to design a single reference architecture that can be used as a blueprint for all conceivable implementations. Hence, several reference architectures will co-exist in IoT. Before we consider various reference IoT architectures and security implementations in them in detail, let's take a look at IoT components.

2.1.1 IoT Components

While there are many different IoT architecture patterns (see Section 2.1.2), they all share one common set of components – a three-tier topology consisting of physical device, edge, and platform.

Physical Devices. All IoT physical devices have a common attribute – their individual identity as a physical device. The ability to uniquely identify "things"' is critical in IoT as it enables not only unique identification of billions of devices but also control of remote devices through the Internet. The physical devices may have some level of computing power that is either embedded in the device or directly attached in the form of their actuators or controllers. Connectivity varies from devices connected directly to other physical devices or to edge and to connectivity to one or more IoT systems. Today device security is largely implemented on a case-by-case basis in connection with customer demands and capabilities.

Edge. Sensors, controllers, actuators, tags and tag readers, communication components, gateways and the physical devices are components that form the edge. At the edge tier data from all the end-nodes is collected, aggregated, and transmitted over the proximity network to a border gateway. The edge sizes run the gamut from a small single physical device with a direct connect

to a platform to a large manufacturing plant comprising all manufacturing equipment with a communications functional component and edge computing platform, or anything in between.

Platform. The data from the edge tier is sent over the access network to the platform that is responsible for data transformation and processing. The platform tier also manages control data flowing in the other direction, for example, from the enterprise to the edge tiers. The majority of the functions related to the information and operations domains is at the platform tier.

Depending on the scale and timeliness requirements of IoT applications, the data will be either streamed to a centralized cloud (see Section 8) or will require distributing storage and compute to the edges —- closer to the devices. The latter is typical for applications generating data volumes that are too large to be affordably transferred to a centralized cloud. Architecture with data processing closer to where data is generated or used is called Fog Computing.[7] Fog Computing can be seen as an extension of the cloud to deploy cloud services closer to the "things" that produce IoT data.[7] From the IoT security perspective, Fog Computing architecture provides better security as it keeps sensitive data inside the network and the data spends less time in transit. Moving security capabilities into the edge is the easiest way to protect endpoints and devices behind the gateway in a uniform manner. Dell and other major IoT solution providers are working on the vendor-neutral, open source project EdgeX Foundry.[8] The goal is to provide the open interop platform for the IoT edge with the simplification and standardization of the IoT edge framework.

Another classification of IoT components is presented by NIST, which considers IoT a technology domain involving sensing, computing, communication and actuation.[4] The NIST IoT guideline defines five core primitives: sensor, aggregator, communication channel, external utility and decision trigger. The term "primitive" is related to smaller blocks from which larger blocks or systems can be built. These primitives are considered the building blocks for a Network of Things (NoT) that includes IoT as a subdomain. The model also defines six elements — environment, cost, geographic location, owner, Device_ID, and snapshot that are key factors in IoT trustworthiness. The concept of primitives and elements makes it easier to develop IoT security solutions. For example, while issues such as geo-location and sensor ownership can be addressed by implementing authentication, authentication may not be relevant if an adversary gains control of the sensors.

2.1.2 IoT Architectures and IoT Security

As we mentioned in Section 2.1, the variety of IoT applications has resulted in various IoT architecture models. We start with a three-layer architecture:[9]

1. Perception layer
2. Network layer
3. Application layer

The perception layer – also called the recognition layer[9] – is the lowest layer of the conventional architecture of IoT. This layer is responsible for collecting data from "things" or the environment (such as Wireless Sensor Networks [WSN], heterogeneous devices, sensors, etc.) and processing them.

Some other models include one more layer: a support layer that lies between the application layer and network layer. For example, the ITU-T (International Telecommunications Union - Telecommunication Standardization Sector) suggests a layered IoT architecture that is composed of four layers (Fig. 3).[10] The IOT application layer containing the application user interface is the top layer. The services and application support layer is the second layer from the top. The third layer is the network layer which contains the networking and transport capabilities. Finally, the lowest layer is the device layer, which contains gateways, sensors, RFID tags, etc. The security capabilities categorized into generic and specific (Fig. 3), are distributed along all four layers.

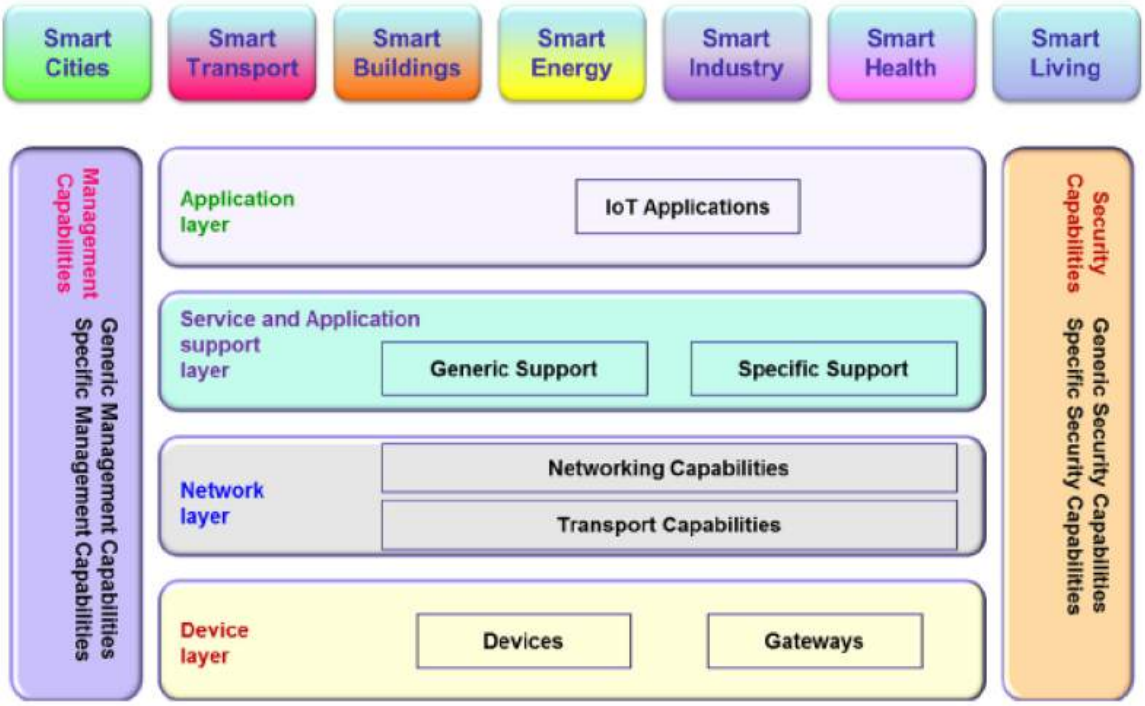


Figure 3: IoT layered Architecture (Ref.10)

The IoT European Research Cluster (IERC) adds more details to the ITU-T architecture of IoT by presenting the functions included in every layer (Fig. 4).[11] For example, the third layer – the network and communication layer – includes the network and communication capabilities such as gateway, routing and addressing, energy optimization, QoS (Quality of Service), flow control and reliability, and error detection and correction. The security management functions listed on the right side include authorization, key exchange and management, trust, identity management and authentication. We will review these IoT security functions in detail later (Sections 4-6).

Cisco has suggested a seven-level IoT reference model describing the functionality each level should have.[12] Figure 5 shows the Cisco IoT reference model and its levels. Data flows in the model are in both directions. While control information flows from the top of the model (Level 7) to the bottom (Level 1), the flow of information is the reverse in a monitoring pattern. The security measures presented in the Cisco IoT Reference must (1) secure each device or system; (2) provide security for all processes at each level; and (3) secure movement and communication between each level, whether north- or south-bound. Therefore, the security functions are spread through all the levels, as shown in Figure 5.

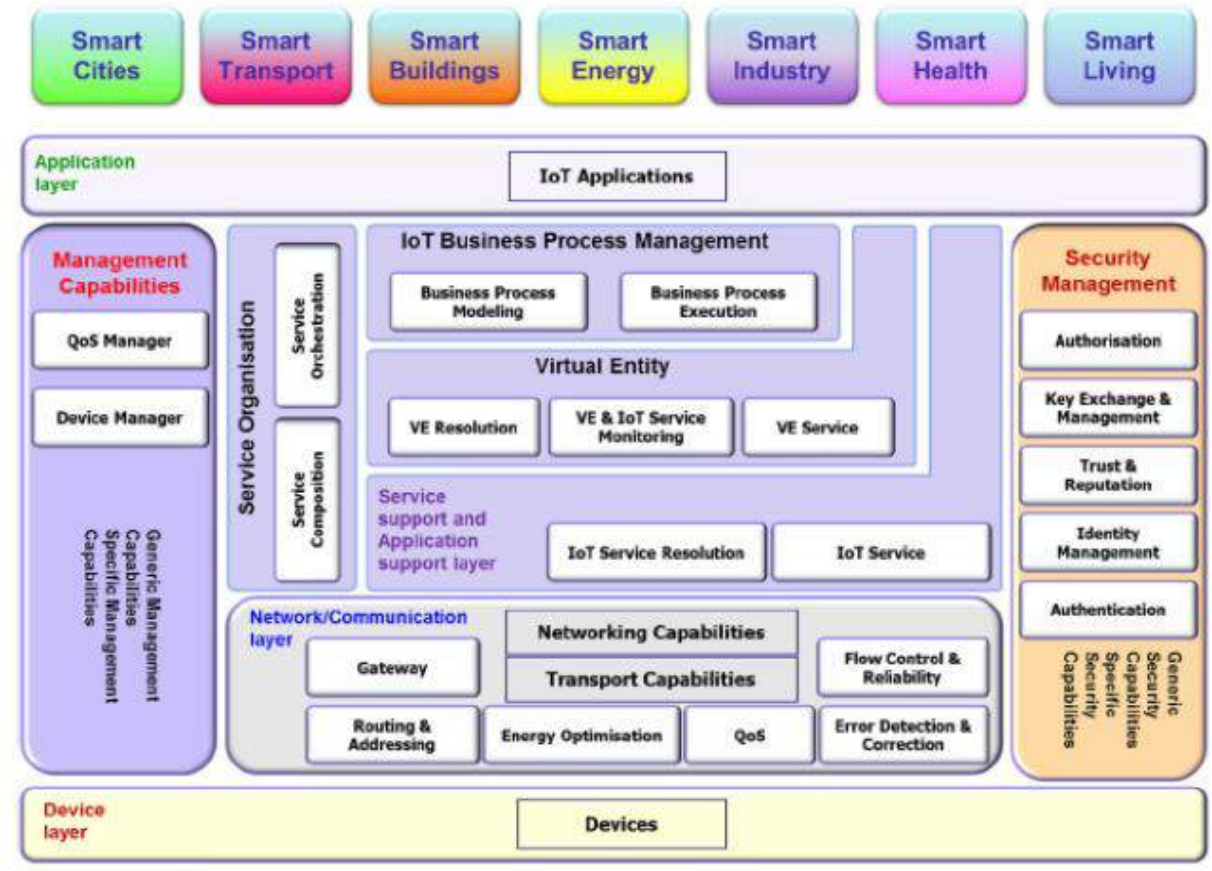


Figure 4: Detailed IoT Layered Architecture (Ref.11)

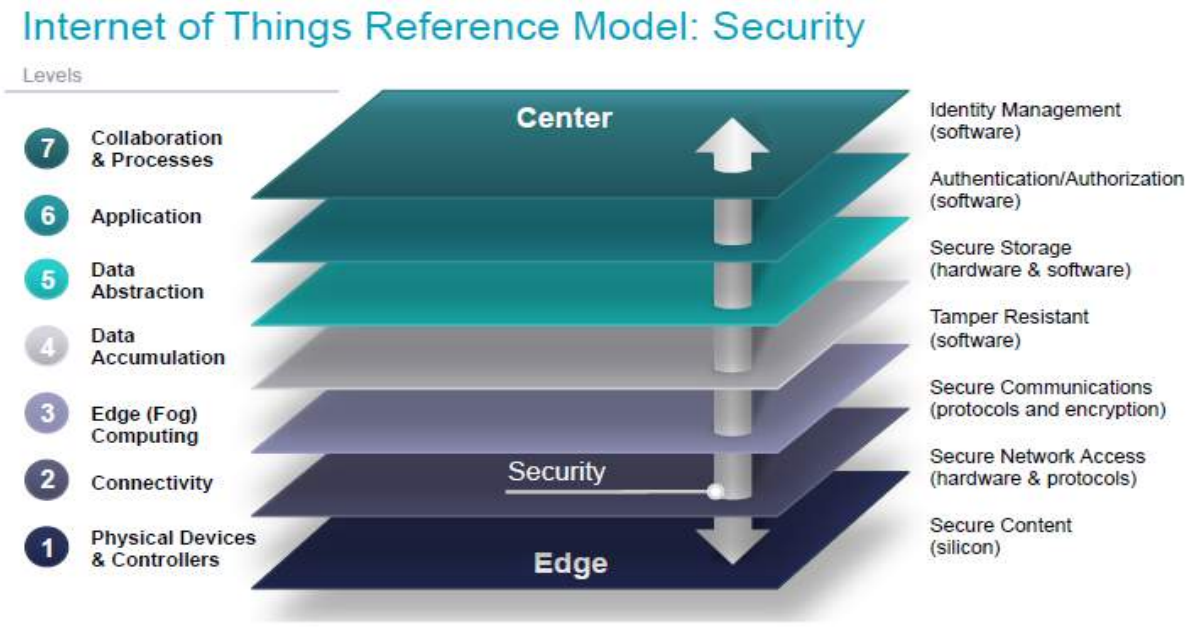


Figure 5: Pervasive Security throughout the IoT Reference Model (Ref.12)

To complete our review of the IoT architecture models, we list some IoT technologies used at various architecture layers (Table 1).[13]

Communication Technologies	
Short range	NFC, RFID, ANT, Bluetooth, Zigbee, Z-Wave, IEEE802 15.4, Wi-Fi
Medium range	WiMAX, Weightless, DASH7, EnOcean, PLC, QR Code, Ethernet
Long range	GPRS, GSM, GPS, 3G/4G, LTE, Satellite, LoRaWAN
Prototype Hardware	
Raspberry Pi, Hackberry, Arduino Yun, Arduino Uno, PCDuino, The Rascal, Cubie Board, BeagleBone Black, OpenPicus Flyport Wi-Fi, Pinoccio	
Operating System	
Tiny OS, Contiki, Mantis, Nano-RK, LiteOS, FreeRTOS, Riot OS, SNAP OS, Abacus OS, Sapphire OS	
Protocol	
REST, IPv6, 6LoWPAN, UDP, LoRa, LoRaWAN, DTLS, XMPP-IoT, SSI, NanoIP, MQTT	

Table 1: IoT Technologies

In this article, we discuss the security aspects of some of these technologies (Section 6).

3. IoT Security Challenges

3.1 IoT Security Risks and Challenges

Three categories of IoT risks include:

1. Risks that are typical in any Internet system
2. Risks that are specific to IoT devices
3. Safety to ensure no harm is caused by misusing actuators, for instance.

Traditional security practices such as locking down open ports on devices belong to the first category (for example, a fridge connected to the Internet in order to send alerts about the product inventory and temperature may use an unsecured SMTP server and can be compromised by a botnet). The second category includes issues specifically related to IoT hardware, e.g. the device may have its secure information compromised. For example, some IoT devices are too small to support proper asymmetric encryption. Furthermore, any device that can connect to the Internet has an embedded operating system deployed in its firmware and many of these embedded operating systems are not designed with security as their primary consideration.

In order to make IoT services available at low cost with a large number of devices communicating securely to each other, there are many security challenges to overcome. We will briefly review some main challenges.

Scalability: Managing a large number of IoT nodes requires scalable security solutions.

Connectivity: In IoT communications (Section 6), connecting various devices of different capabilities in a secure manner is another challenge.

End-to-End Security: End-to-end security measures between IoT devices and Internet hosts are equally important.

Authentication and Trust: Proper identification and authentication capabilities and their orchestration within a complex IoT environment are not yet mature. This prevents establishment of trust relationships between IoT components, which is a prerequisite for IoT applications requiring ad-hoc connectivity between IoT components, such as Smart City scenarios. Trust management for IoT is needed to ensure that data analytics engines are fed with valid data (Section 5.1). Without authentication it is not possible to ensure that the data flow produced by an entity contains what it is supposed to contain.

Identity Management: Identity management is an issue as poor security practices are often implemented. For example, the use of clear text/Base64 encoded IDs/passwords with devices and machine-to-machine (M2M) is a common mistake. This should be replaced with managed tokens such as JSON Web Tokens (JWT) used by OAuth/OAuth2 authentication and authorization framework (the Open Authorization).

Attack-Resistant Security Solutions: Diversity in IoT devices results in a need for attack-resistant and lightweight security solutions. As IoT devices have limited compute resources, they are vulnerable to resource enervation attacks.

3.2 IoT Security Threats and Attacks

To emphasize security risks in IoT, its acronym has been presented as Interconnection of Threats (IoT).[14] Indeed, IoT devices are particularly vulnerable to physical attacks, software attacks, side-channel attacks, and so on as presented in Table 2.

Threats	Attack Procedure	Security Requirement	Examples
Physical attacks	Tamper with the hardware and other components.	Tamper resistance	Layout reconstruction, micro-probing
Environment attacks	The device encryption key can be discovered by the attacker by recovering the encryption information.	Secure encryption scheme	Timing attack, side-channel attack, fault analysis attack
Cryptanalysis attacks	Find ciphertext to break the encryption.	Secure encryption scheme	Known-plaintext attack, chosen plaintext attack
Software attacks	Exploit vulnerabilities in the system during its own communication interface and inject malicious codes.	Proper antivirus update	Trojan horse, worms, or viruses

Table 2: Security Threats to IoT Devices (Ref.13)

Current IoT platforms are built using technology solutions from a wide variety of vendors. Some of these platforms are an eclectic mix of components repurposed from existing solutions for use in specifically designed platforms with the hope that the components will work together in a secure way. Security measures within the IoT components, if any, have not been designed to take into account the dependencies resulting from the IoT connectivity capabilities. For example, industrial devices often do not have proper authentication mechanisms because they have been designed to be used in physically protected and isolated environments. Another example is the challenge of providing software updates or security patches in a timely manner to end nodes without impairing functional safety.

Comprehensive risk and threat analysis methods as well as management tools for IoT platforms are required. Developing mitigation plans for IoT attacks requires understanding attack types and the sequence of actions taking place when the attacks are happening. Let us start with considering IoT attack categorization. Analysis of security attacks helps to understand an actual view of the IoT networks and enables us to determine mitigation plans.

3.2.1 Attack Categorization For IoT Process Phases

In general, an IoT process can be considered as a five phase sequence, from data collection to data delivery to the end users.[14] Table 3 demonstrates the variety of attacks categorized for the five phases of IoT: data perception, storage, intelligent processing, data transmission, and end-to-end delivery.

Phase	Attack/Threat	Description
Data Perception: Various types of data collectors can be used. The device may be a static body (body sensors or RFID tags) or a dynamic vehicle (sensors and chips).	Data Leakage or Breach, Data Sovereignty, Data Loss, Data Authentication.	Data leakage can be internal or external, intentional or unintentional, involving hardware or software.
Storage. If the device has its own local memory, data can be stored. In the case of stateless devices, the data can be stored in the cloud.	Attack on Availability, Access Control, Integrity, Denial of Service, Impersonation.	Availability is one of the primary security concerns. Distributed denial of service (DDoS) is an overload condition that is caused by a huge number of distributed attackers.
Intelligent Processing	Attack on authentication	An IoT solution provides data analysis and intelligent services in real time.
Data Transmission	Channel security, session hijack. Routing protocols, flooding.	Threats in transmission, such as interrupting, blocking, data manipulation, forgery, etc.
End-to-End Delivery	Man or machine. Maker or hacker.	Delivery of processed data on time without errors or alteration.

Table 3: Attack Taxonomy According to the IoT Process Phases

3.2.2 Attack Categorization According to IoT Architecture

As discussed in Section 2.1, there are various IoT architecture models. In general, the IoT architecture is assumed to have four layers, presented in Fig. 3. We will briefly review the main security threats at the perception, network, and service layers. The most important security concerns in IoT presented as four-layer architecture (Fig. 3) are summarized in Table 4.

Security Concerns	Application & Interface Layer	Service Support Layer	Network Layer	Device Layer
Insecure web interface	✓	✓	✓	
Insufficient authentication/authorization	✓	✓	✓	✓
Insecure network services		✓	✓	
Lack of transport encryption		✓	✓	
Privacy concerns		✓	✓	✓
Insecure cloud interface	✓			
Insecure mobile interface	✓		✓	✓
Insecure security configuration	✓	✓	✓	
Insecure software/firmware	✓		✓	
Poor physical security			✓	✓

Table 4: Top Ten Vulnerabilities in IoT (Ref.15)

3.2.2.1 Security Threats at the Sensing/Perception Layer

To fully implement IoT security, it must be designed and built into the devices themselves. This means that IoT devices must be able to prove their identity, maintain authenticity, sign and encrypt their data to maintain integrity, and limit locally stored data to protect privacy. The security model for devices must be strict enough to prevent unauthorized use but flexible enough to support secure ad hoc interactions with people and other devices on a temporary basis. For example, while unauthorized changing of the toll rate on a connected parking meter should be prevented, the meter should have a secure interface to reserve and pay for the parking spot for a limited duration.

Physical Damage. Some attackers may lack technical knowledge and their attacks are limited by destroying devices. As device enclosures are often not tamperproof, the devices can be opened and their hardware can be accessed via probes and pin headers. Physical security

requires designing tamper resistance into devices so that it is difficult to extract sensitive information such as personal data, cryptographic keys, or credentials. Many devices cannot protect their code and data from external access. As a result, an attacker can clone entire devices or manipulate their software and data: for example, to manipulate a glucometer so that it will provide incorrect readings. Another example is damage to hundreds of smart traffic light devices by thieves who stole the devices' SIM cards.[16] The stolen cards were then used to make mobile phone calls in South Africa. The damage to the traffic light system resulted in many car crashes and a high cost to fix the entire system.

Node Capture. An active attacker can extract the information that the devices contain instead of destroying them.

Sinkhole Attack. If sensors are left unattended in the network for long periods, they become susceptible to sinkhole attack. In this attack, the compromised node extracts the information from all the surrounding nodes.

Selective Forwarding Attack. Malicious nodes may choose packets and drop them out, thereby selectively filtering certain packets and allowing the rest. Dropped packets may carry necessary sensitive data for further processing.

Witch Attack. This attack occurs when a malicious IoT node takes advantage of failure of a legitimate node. When the legitimate node fails, the factual link takes a diversion through the malicious node for all its future communication, leading to data loss.

HELLO Flood Attacks. A malicious node initiates a HELLO flood attack by sending HELLO messages to all the neighbors that are reachable at its frequency level. Hence, it becomes a neighbor to all the nodes in the network. As the next step, this malicious node will broadcast a HELLO message to all its neighbors, affecting their availability. Flooding attacks cause non-availability of resources to legitimate users by distributing a huge number of nonsense requests to a certain service.

3.2.2.2 Security Threats at the Network and Service Support Layers

The service support layer (Fig. 3) represents the IoT management system and is responsible for onboarding devices and users, applying policies and rules, and orchestrating automation across devices. Role-based access control to manage user and device identity and the actions they are authorized to take is critical at this layer. To achieve nonrepudiation, it is also important to maintain

an audit trail of changes made by each user and device so that it is impossible to refute actions taken in the system. This monitoring data could also be used to identify potentially compromised devices when abnormal behavior is detected. We will briefly consider some typical attacks at the network and service support layer.

Man-in-the-Middle (MITM) Attack. Man-in-the-middle attack is an example of the eavesdropping possible in the IoT. As device authentication involves exchange of device identities, identity theft is possible due to man-in-the-middle attack.

Replay Attack. During the exchange of identity-related information or other credentials in IoT, this information can be spoofed, altered or replayed. Replay attack is essentially a form of active man-in-the-middle attack.

Denial of Service Attack. As the IoT devices in IoT are resource constrained, they are vulnerable to resource usage attack. Attackers can send messages or requests to a specific device to consume its resources.

4. IoT Security Requirements

Security must be addressed throughout the IoT lifecycle from the initial design to the services running. For example, implementation of security features should start during device manufacturing. Code signing and code obfuscation are some steps that manufacturers can follow to ensure their device is not hacked or unwanted code is not inserted by a malicious user.

The main security requirements in IoT scenarios include data confidentiality, privacy, and trust, as shown in Figure 6.

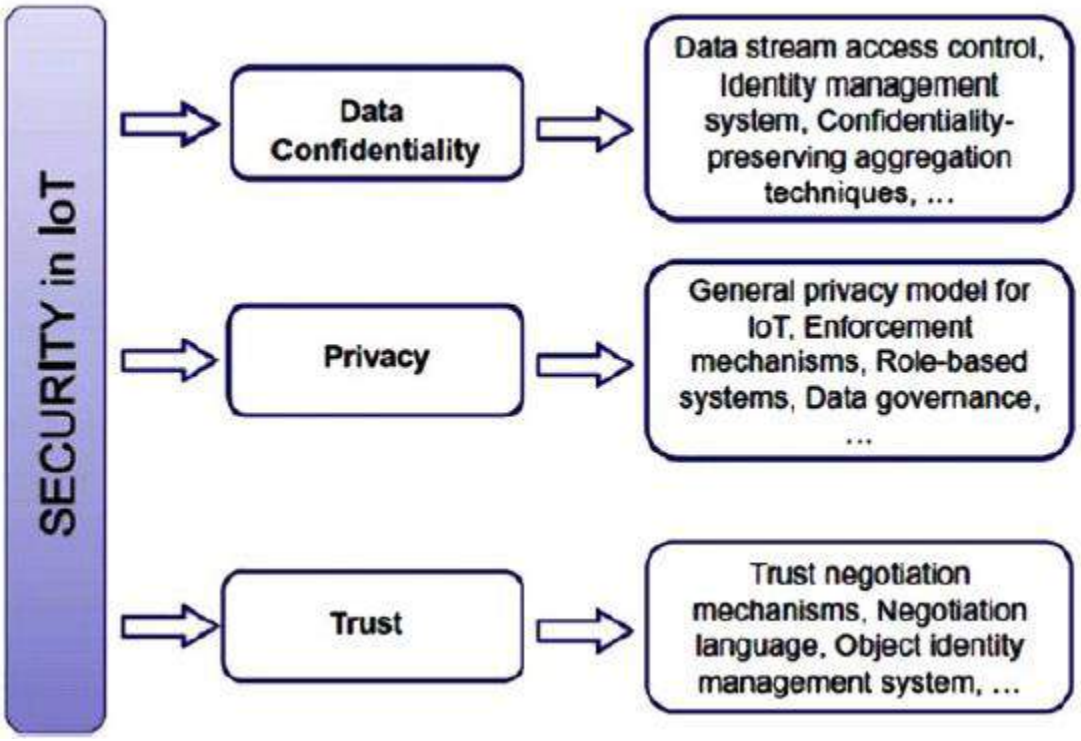


Figure 6: Security Challenges and Requirements in Internet of Things (Ref.17)

The IoT key security requirements can be presented as shown in Fig. 7. The main security requirements are categorized into six domains.

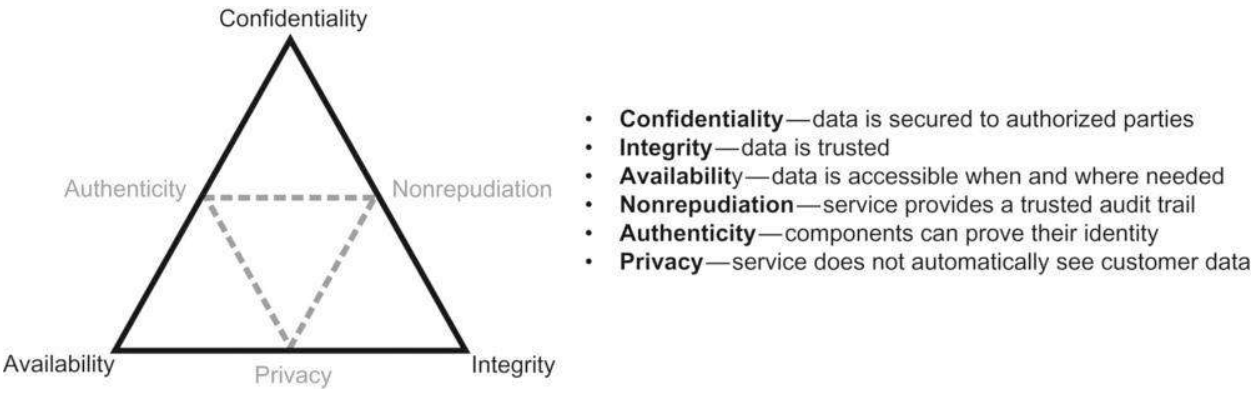


Figure 7: Security Requirements in Internet of Things (Ref.15)

We will discuss IoT security based on the three main domains shown in Figure 6: trust, data confidentiality, and privacy.

5. Trust, Data Confidentiality, and Privacy in IoT

5.1 Trust in IoT

Trust and security are based on tokens or credentials, provided by a trust management infrastructure, which are embedded in and potentially shared between devices. These tokens can be symmetric keys or digital certificates. They are useful in deflecting external attacks initiated by entities that are not in possession of credentials, but fail to deflect internal attacks, where credentials or nodes that own credentials have been compromised. Public key infrastructure (PKI) is used to generate and control certificates. In some security critical environments, the trusted platform modules (TPM; see Section 5.1.2) are used, which provide a hardware-based root of trust and a high level of confidence that the identity attributes delivered belong to the particular device. As IoT is a dynamic system, measures to attest the trustworthiness of IoT components throughout their lifetime are required.[18]

5.1.1 Trust and Security from a Device Perspective

IoT devices are vulnerable in many aspects (Section 3.2), so the trust management for them is a difficult endeavor. A general problem related to trustworthy firmware is that many embedded processors (even if they operate under a modern multitasking operating system) do not provide process encapsulation via memory virtualization. As a result, malicious code in a firmware image can access and manipulate credentials used by other system processes to initiate an internal attack. Hence, determining the trustworthiness of firmware components individually is not sufficient and the firmware image as a whole must be validated.

Devices with a static ("factory-flashed") firmware image can maintain a higher degree of trustworthiness over time compared to devices that are updated dynamically in the field via firmware download because the upload mechanism itself can have a potential back door for attacks. A secure device firmware updating or patching mechanism is an integral component to maintain security. A network-wide update mechanism should include robust integrity and authenticity checks, minimize service outages, and allow for a version rollback if needed.

Any trust management system for IoT deployments must have the ability to dynamically withdraw trust of individual devices. Likewise, individual devices must be able to dynamically validate the trustworthiness of other nodes they communicate with.[19]

5.1.2 Trust and Secure Key Storage

The robustness of trust tokens can be increased by using keystores. A keystore – either a file (software stores, see below) or a hardware device (hardware stores) – provides storage for keys. In the case of passive keystores that securely save and retrieve credentials, cryptographic operations are executed outside these stores by the device's CPU. In contrast, active keystores allow internal execution of cryptographic operations via an application program interface (API), so the credentials are never exposed. Various types of keystores are described below.

Hardware Stores. Modern cryptosystems use hardware security modules (HSMs). These specialized tamper-resistant devices are used for managing cryptographic keys. General-purpose HSMs provide a secure and generally configurable administration. Their main disadvantage is their lack of flexibility if uncommon token formats or algorithms are used. Cryptographic smart cards (embedded or otherwise) and cryptographic USB dongles are low-cost HSMs. They are particularly adequate for resource-constrained nodes or low-cost trust management infrastructures.

Trusted Platform Modules. Trusted platform modules (TPMs) developed by the Trusted Computing Group are dedicated microcontrollers or can be integrated within devices such as memories. Their function is to protect hardware (by authenticating devices), booting processes, and so on. Secret data such as encryption keys are stored securely on the TPM by hashing. By providing the means to verify that a platform will behave as it should, the TPM helps establish a hardware root of trust.

Software Stores. The natural place for software keystores is in devices with low security requirements or low-cost embedded systems that have no provisions to physically connect to hardware modules. There is a large number of active and passive software stores that can be used in IoT systems. For example, the homonymous public-key cryptography standard (PKCS) initially defined by RSA Security (now part of Dell EMC) is used in PKCS#12 software stores. In principle, PKCS#12 defines two types of integrity/privacy modes; asymmetric cryptography and password-based. Java stores are part of a much larger programming framework, the Java cryptography architecture/Java cryptography extension (JCA/JCE). This framework defines a provider-based, pluggable architecture that includes, among many other things, keystore implementations.

5.1.3 Identity Management

Current identity and access management (IAM) solutions in IoT are limited in their ability to adjust to storing identities and entities on a large scale.[19] This limitation has resulted in a lack of application integration layers for IoT-based applications. At this time, no overall framework exists for how to discover and manage IoT entities and their identities across different solutions.

The typical approach is to provide limited access based on an expected role rather than least-privileged access in traditional IAM systems. As a result, authentication from the same device may provide different access capabilities based on how the user has authenticated to the device. IoT will require traditional IAM systems to include M2M entities. In general, IAM platforms will need to be modified in order to cover identity in IoT-based systems.

5.2 Data Confidentiality in IoT

The data a node sends or receives can be trusted if its integrity, optionally in combination with data confidentiality via symmetric encryption (using the Advanced Encryption Standard [AES] algorithm as a de facto industry standard), is assured. For example, in a body area network, a wireless glucometer sends glucose readings to an integrated insulin pump. This information must be protected from accidental or deliberate tampering, and patient privacy considerations require the data to be encrypted. However, there are challenges of cryptography on devices with constrained resources, for example, 8-bit microcontrollers with limited RAM. Encryption is often implemented directly in hardware, while data integrity is provided via message authentication codes or cryptographic hashes that are attached to the data payload.

For establishing peer authenticity, a peer should be able to validate another peer's identity before a communication link is established.[19] Coming back to the above example of an insulin pump, the pump must be able to validate that it actually connects to a trusted glucometer (and subsequently receives data from it) and not to a malicious device. Proof of authorization provides assurance that a peer has the authority to (a) communicate with another peer and (b) conduct a certain action. In our example, (1) a glucometer accepts only data requests from an insulin pump (and not from the blood pressure monitor); in addition, both glucometer and pump must be from the same manufacturer; and (2) a reset command sent to the glucometer sensor by the insulin pump (after a sensor reconfiguration) should only be executed if the insulin pump has the required authorization level.

5.3 Privacy in IoT

Preserving privacy in IoT is still a significant challenge.[20,21] Privacy involves security of personal information as well as the ability to control what happens to this information. Privacy issues with IoT systems are complicated by the fact that a system is more than the sum of its parts. Privacy considerations for low-level devices may well differ from the concerns generated at an application or data analytics level. At the same time, privacy breaches at any level in the system affect the entire system.

A lot of private information can be collected from the smart devices. Control of this information is weak in current IoT techniques. In many cases data is collected passively and because of it some privacy breaches can go unnoticed for a long time. The question of IoT data ownership – who owns which data and who controls where data goes – creates major issues from regulatory, ethical, and financial standpoints. End users believe they own all the data. The original equipment manufacturers believe they own, or at least have access rights to, the data generated by their endpoints. The service providers in many cases believe they own the data, as do the application providers. Issues of data ownership become increasingly complex as more heterogeneous IoT systems with more players from divergent organizations are deployed. Decommissioned old devices can still keep a lot of privacy-sensitive information and data sanitization should be done for them.

Seemingly benign combinations of IoT data streams from various sources can jeopardize privacy. For example, a user's network-enabled toothbrush might capture and transmit harmless data about a person's tooth-brushing habits. However if the user's refrigerator reports the inventory of the foods he/she eats and his/her fitness-tracking device communicates his/her activity data, the aggregation of these data streams provides a much more detailed and private description of the person's overall health. In some cases the user might not be even aware that an IoT device is collecting data about the individual and potentially sharing it with third parties. This type of data collection is becoming more common in consumer devices such as smart TVs and intelligent personal assistants. These devices have voice recognition or vision features that allow them to continuously listen to conversations or watch for activity in a room and selectively transmit that data to a cloud service for processing, which sometimes includes a third party. People might be in the presence of such devices without knowing their conversation or activities are being monitored and their data captured.

Privacy-enhancing techniques are often lacking in data analytics and are substituted by non-technical means such as SLAs and other customer agreements to ensure data processing in compliance with legal regulations that can have significant regional differences.

6. Security in IoT Networks

6.1 Overview of IoT Communication Technologies

IoT connectivity requirements are very diverse (Fig. 8) and, as a result, various types of communication technologies are used (Fig. 8, Table 5).

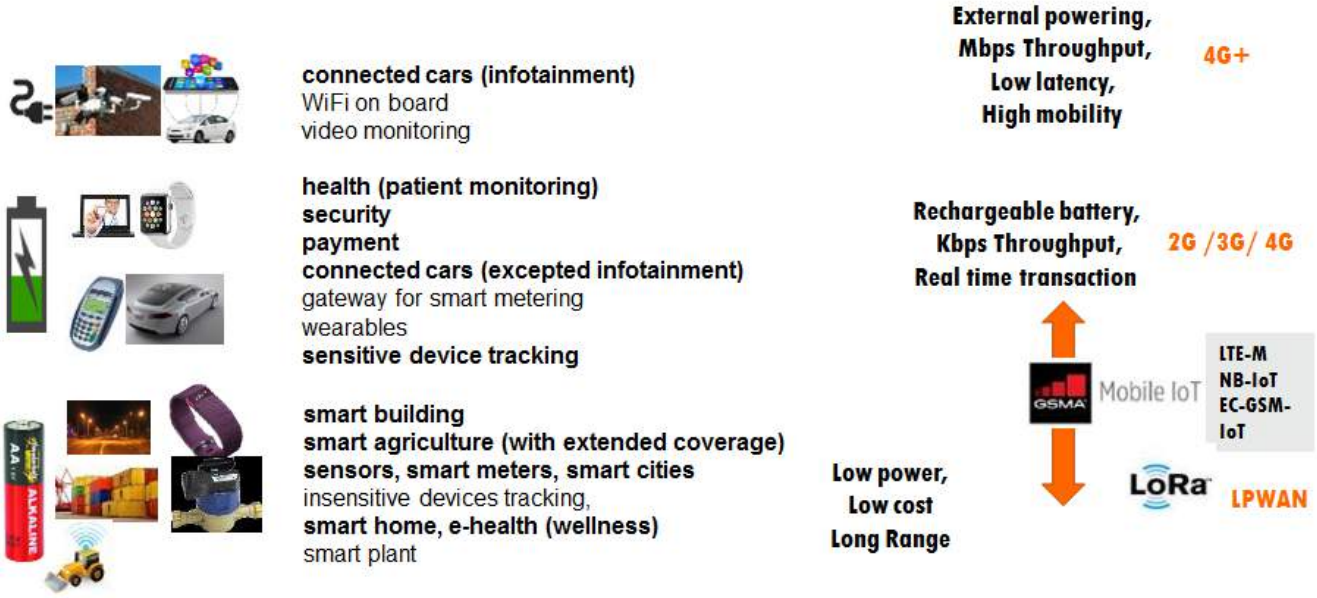


Figure 8: IoT Connectivity Requirements

Various communication technologies have been deployed by enterprises to implement IoT solutions (Table 5):

	LR-WPAN	LoRaWAN	BLE	RFID
Standard	IEEE 802.15.4	LoRaWAN R.1.1	IEEE 802.15.1	ISO/IEC 18000
Frequency band	868/915/2450 MHz	868/900 MHz	2.402 – 2.481 GHz	125 or 134 KHz for Low-Frequency RFID;13.56 MHz for High-Frequency RFID systems, 860 ~ 960 MHz for Ultra High Frequency RFID
Transmission range	10-20 m	Several km (2-5 km in urban areas and 15 km in suburban areas))	10-100 m	Up to 100 m (active tag)
Data rate	40-250 Kbps	0.3-50 Kbps	The theoretical over-the air data rate is 1 Mbps (the LE 1M PHY Layer transfer rate). The practical application throughput depends on many factors and is reported as 10-20 Kbps.[22,23]	6.7 - 848 Kbps (HF Passive)
Energy consumption	Low	Very Low	Very Low	Low
Cost	Low	High	Low	Low
Article section	6.2.1	6.3.1	6.2.3	6.2.5

Table 5: IoT Communication Technologies

As seen from Table 5, two main categories of networks used in the IoT are short-range and long-range low power networks. We will consider security aspects of each of these types.

6.2. Security in Short-Range Low Power IoT Networks

6.2.1. 6LoWPAN Security

Low-data-rate, low-power wireless personal area networks (LR-WPANs) are based on IEEE 802.15.4 Standard for Low-Rate Wireless Networks. The standard is implemented by using several technologies such as 6LoWPAN (an IETF standard), Zigbee (Section 6.2.4), Z-Wave and EnOcean (building and home automation standard protocols), and SNAP (Simple Network Access Protocol). The idea of 6LoWPAN is a combination of IPv6 and IEEE 802.15.4. LoWPAN standard allows IPv6 to be used over 802.15.4 wireless networks. The Thread protocol for home automation devices also runs over 6LoWPAN.

A 6LoWPAN network consists of one or more LoWPAN networks connected to the Internet through the edge router that controls flows incoming and outgoing from the LoWPAN. Within LoWPAN, devices do not use the IPv6 address or user datagram protocol (UDP) full header for transmissions as it remains at the edge router to communicate with the outside. Routing issues in 6LoWPAN are addressed by the IETF-ROLL Working Group in its design of RPL (a de facto routing protocol for Low-power and Lossy Networks [LLNs]).

The security in the 6LoWPAN networks must limit data access only to authorized users, provide data integrity and be capable of detecting malicious intrusion. Since 6LoWPAN combines IEEE 802.15.4 and IPv6, an intrusion detection system is required to monitor the traffic of two sides.

The lack of authentication at the 6LoWPAN layer, the best effort semantics for fragment transmissions, and scarce memory resources of the networked devices make the packet fragmentation mechanism of 6LoWPAN vulnerable.[24] For example, an attacker can selectively prevent correct packet reassembly on a target node. Specifically, an attacker can mount attacks by only sending a single protocol-compliant 6LoWPAN fragment.[25]

6.2.2. Security in RPL

IPv6 Routing Protocol for LLNs (RPL) is designed for routing IPv6 traffic in low-power networks implemented over 6LoWPAN with high or unpredictable amounts of packet loss. The RPL security utilizes a “Security” field after the 4-byte ICMPv6 message header. Information in this field indicates the level of security and the cryptography algorithm used to encrypt the message. RPL offers support for data authenticity, semantic security, protection against replay attacks, and confidentiality and key management. RPL attacks include selective forwarding, sinkhole, Sybil, Hello flooding, wormhole, black hole and denial of service attacks.

6.2.3 Security in Bluetooth Low Energy (BLE)

BLE Protocol. BLE is a low-power version of the Bluetooth 2.4 GHz wireless communication protocol (Table 5). While the BLE data rate and radio range are lower than the same metrics in classic Bluetooth, BLE is designed for very low-power applications running off a coin battery (for example, the popular CR2032). The low-power and long battery life make it possible for BLE sensor devices to operate for many years without needing a new battery. To enhance security, the BLE version 4.2 introduces the new BLE Secure Connections pairing model. Let us briefly review the main BLE security challenges: passive eavesdropping, MITM attack (Section 3.2.2.2), and identity tracking.

Eavesdropping. The protection against passive eavesdropping can be based on encrypting communication with a key. While earlier versions of BLE (Bluetooth 4.1 or older) devices used easy-to-guess temporary keys to encrypt the link for the first time, BLE 4.2 uses the Federal Information Processing Standard (FIPS) compliant Elliptic Curve Diffie-Hellman (ECDH) algorithm for key generation (Diffie-Hellman Key—DHKey).

Man-in-the-Middle (MITM) Attacks. Protection against MITM attacks is to ensure that the device the communication started with is indeed the intended device rather than an unauthorized device presenting as the intended one. LE Secure Connections pairing provides MITM protection by using the numeric comparison method.

Privacy/Identity Tracking. As most of the BLE advertisement and data packets contain the source addresses of the devices that send the data, third-party devices could associate these addresses to the user identity and track the users. A frequent change of the private addresses so only the trusted parties could resolve them can serve as protection against this thread.

6.2.4 Zigbee Security

Zigbee Protocol. Zigbee is a wireless technology based on the IEEE 802.15.4 standard and used in various application areas, including home automation, smart energy, remote control and health care. It has a longer range than BLE and a lower over the air data rate than BLE (Table 5). The Zigbee Alliance has developed the Zigbee Health Care Profile for secure non-critical patient monitoring, chronic disease management, drug administration (e.g. insulin pumps), and personal wellness monitoring. ISO/IEEE 11073 Personal Health Data standards-conformant devices (for example, blood pressure monitors, respirometers, pulse oximeters, ECGs, weight scales, and thermometers) are supported by the Profile.

Zigbee Security Features. As with other IoT protocols, Zigbee has unavoidable trade-offs made to keep the devices low-cost, low-energy and highly compatible. To simplify the interoperability of devices, Zigbee establishes the same security level for all devices on a given network and all layers of a device. In addition, it assumes that "the layer that originates a frame is responsible for initially securing it".[24] Zigbee supports 128-bit AES encryption.

Zigbee security includes an assumption that keys are securely stored, and devices are pre-loaded with symmetric keys so they have never to be transmitted unencrypted. However, when a non-preconfigured device joins a network, a single key may be sent unprotected to enable encrypted communication. This one-time transmission of the unprotected key creates a short timeframe of exploitability during which the key could be sniffed by an attacker.

As discussed in Section 3.2.2.1, the low-cost nature of some types of devices such as light switches or temperature sensors limits the device security features and it cannot be assumed that the hardware is built tamper-resistant. Hence, if an attacker obtains physical access to such a device, it may be possible to access the secret keying material and other privileged information as well as to access the security software and hardware.

A paper published in 2016 explains the attack targeted on Philips Hue Light Bulbs implemented with the Zigbee standard.[26] The light bulbs were infected with a worm/virus that gave the attackers the ability to turn them on and off. The worm was able to attack a light bulb from up to 400 meters away and then spread to nearby bulbs because Zigbee uses hard-coded skeleton keys. Zigbee Alliance in its response claimed that the vulnerability was not part of Zigbee standard, but rather an internal implementation error made by Philips. This allows us to generalize that while technology can be secure, its erroneous implementation could lead to security weaknesses.

6.2.5 RFID Security

Radio Frequency Identification (RFID) is the method of uniquely identifying "things" by transmitting their identity (usually a serial number) using radio waves. At a minimum, an RFID system consists of a tag, a reader, and an antenna. RFID tags storing identifiers and data are attached to devices for reading by an RFID reader. RFID tags can be active, passive, or assisted passive. Active RFID tags using their own power source can broadcast with a read range of up to 100 meters (Table 5). Passive tags are ideal for devices without batteries, as the ID is passively read by the reader. They have a read range from near contact and up to 25 meters and utilize the power of a reader's interrogation signals for any response. Assisted passive tags become active when an RFID reader

is present. RFID technology is used not only in traditional applications such as asset or inventory tracking, but also in security services such as electronic passports and RFID-embedded credit cards. Even many pets – including my cat – have RFID chips in them. Some of the numerous RFID security and privacy threats are presented in Table 6 (adapted from Ref. 27).

Threats	Key Component	Security Need
DoS attacks	RFID tags and reader communications	Encryption
Eavesdropping	User private data	Encryption
Skimming	User private data	Shielding, blocking tags
Relay attack	Authentication result	Synchronization
Side-channel attack	User private data	Authentication
Hardware destruction	Tags	Protective electronic component
Software destruction	Commands	Key, password

Table 6: Security Threats in RFID Technology

The main RFID security measures and defenses are presented in Table 7.

RFID Security Solution	Advantages	Limitations
Killing tags	The simplicity and effectiveness of the method.	1) The tag cannot be reused, its lifetime is limited and it cannot be utilized for after-sale purposes while consumers may wish to keep them alive after buying them. 2) In some applications, the RFID tag is required to be alive and it cannot be killed.
Sleeping tags	The user can switch the state of the tag between active and inactive.	It is possible that the password used for controlling the tags might be overheard by an eavesdropping attack.
Faraday cage	Extremely effective at providing consumer privacy against eavesdropping and tracking attacks.	The tag is protected from being read by unauthorized reader only when it is inside the cage.
Blocker tags	The major advantage is keeping the functionality of tags. In contrast	A major drawback of this method is its limited safety. The attacker cannot have

	to the killing tag solution with the tag lifetime is limited by the purchasing time, this method makes the tags more useful by expanding their lifetime.	access to tags within a defined range but tags are not protected from attacks beyond this range. Furthermore, blocker tags are not applicable everywhere.
Minimalist cryptography	The scheme can offer some resistance to corporate espionage, like clandestine scanning of product stocks in retail environments.	An adversary could query a tag multiple times to capture all names so as to defeat the scheme.
Proxy privacy devices	Relying on the reader to provide consumer privacy protection. Alternatively, privacy-enforcing devices like the RFID Guardian can be added to RFID systems.	Relying on the reader for privacy is risky when the reader is public.

Table 7: RFID Security Solutions

The security on RFID tags or during their communications with readers is very limited. RFID security attacks fall into two main categories: privacy violations and security violations. In the former, the attacker tries to collect information from the objects by eavesdropping on the communications between the object and the reader. In the latter, the attacker counterfeits the behavior of a tag or a reader for the purpose of making undesirable communications. Such security attacks may target the physical tag, the communication channel between the tag and the reader, or the application using the RFID technology. To protect the privacy of RFID tags against possible attacks and threats, solutions such as use of Faraday cage, tag killing, tag blocking, re-encryption and many others have been introduced (Table 7). A RFID blocking Faraday cage is an enclosure design made of conducting materials to exclude electromagnetic fields. Various types of Faraday cages — Faraday cage wallet cage, Faraday laptop sleeve, and even Faraday backpack — are sold. Since any exterior radio signals cannot penetrate inside the cage, no reader can have access to the tag to read it as long as the RFID tag is inside such a cage.

6.2.6 Security in NFC

Near-Field Communication (NFC) is a subtype of RFID technology — High-Frequency (HF) RFID — and is based on 13.56 MHz, HF passive RFID/contactless card technology. As NFC devices must be in close proximity to each other (no further than a few centimeters in most cases), it makes NFC a popular choice for secure peer-to-peer communication between consumer devices

such as smartphones. In contrast to typical RFID devices, an NFC device is able to act both as a reader and as a tag.

Threats	Key Component	Security Need
Phishing attacks	Application processor	Interfaces authentication
User tracking	User privacy	Random UIDs
Relay attacks	Tag/reader	Synchronization
Data corruption and manipulation	User data	Use of secure channels
Eavesdropping	User data	Use of secure channels
Interception attacks	User data	Devices should be in an active-passive pairing
Malicious host	Application processor	Interfaces authentication

Table 8: Security Risks and Their Mitigation in NFC.

NFC security threats and protection solutions are shown in Table 8 (adapted from Ref.13).

6.3 Security in Long-Range Low Power IoT Networks

6.3.1 Security in LPWAN: LoRa and LoRaWAN

LoRa and LoRaWAN Protocols. Low-Power Wide-Area Network (LPWAN) is a wireless technology for long range communications at a low bit rate between "things" (connected objects), such as sensors operated on a battery. The LPWAN data rate has a range of 0.3 – 50 Kbps per channel. In general, LPWAN networks have more node and link constraints than 6LoWPAN networks (Section 6.2.1). LPWAN networks do not have IPv6 addressing capabilities yet and the IETF Working Group is working on enabling IPv6 connectivity for LPWAN.[28] LPWAN may be used to create a private wireless sensor network, but it may also be a service or infrastructure offered by a third party, allowing the owners of sensors to deploy them in the field without investing in gateway technology.

LoRa and Sigfox are the most popular LPWAN technologies. LoRa is a proprietary, chirp spread spectrum (CSS) radio modulation technology. LoRa technology is owned by Semtech that has formed the LoRa Alliance. It is used by LoRaWAN (Table 5) that supports low-cost, mobile, and secure bi-directional IoT communication for M2M, smart city, and industrial applications. LoRa devices communicate with LoRa gateways sending data to a network server and onto an application server accessible by owners of LoRa devices. LoRaWAN defines the communication protocol and system architecture for the network and the LoRa physical layer enables the long-range communication link. LoRaWAN also provides management of the communication frequencies, data rate, and power for all devices. Version 1.1 of the LoRaWAN specification maintained by the LoRa Alliance was released in October 2017.[29]

As Sigfox does not offer support for private deployments, users are required to connect to a licensed provider. In contrast, the market model of LoRa is flexible as it enables any customer to buy a LoRa base station for a few hundred dollars and set up his or her own LoRa ecosystem. Companies such as Orange,[30] Comcast, KPN, and Actility are deploying public LoRa networks to meet market demands.

LoRa and LoRaWAN Security. LoRaWAN security based on the security developed for IEEE 802.15.4 radio communication is extended by also using two session keys: a network session key (NwkSKey) and an application session key (AppSKey). These two types of symmetric session keys are unique to each LoRa device. The NwkSkey is used for network layer message integrity from the LoRa device to LoRa network server. The AppSkey is used for application layer end-to-end AES-128 encryption from the LoRa device to the application server.

The LoRaWAN network join procedure requires mutual authentication between an end-device and the LoRaWAN network. LoRa devices can join the network in two ways: either using Over-the-Air Activation (OTAA) or Activation by Personalization (ABP) (Table 9). After a node has joined a LoRaWAN network, all future messages will be encrypted and signed by using a combination of NwkSKey and AppSKey session keys that are used for provisioning LoRa devices. As these keys are known only by the Network Server and the specific node, it is not possible for another node or a man in the middle attack to recover the clear-text data.

The LoRa security model employs symmetric encryption and authentication.[31] It means that the same session keys must be stored in the LoRa device and on the network/application server. This results in several vulnerabilities. While the data in transit between the LoRaWAN network and end devices are protected, the nodes can be vulnerable to physical attacks, in particular if devices are installed in remote or unsupervised areas. If stored keys are extracted, a device can be impersonated on the network. If an attacker with NwkSKey and AppSKey is able to produce correctly signed and encrypted messages, the data coming from individual nodes can be potentially untrustworthy. For example, if the LoRa system sends utility usage information, the usage data can be falsified by the attacker.[32]

The long air time of LoRa messages can be exploited in attacks that include selective jamming. In wormhole attacks, two devices connected using faster technologies can record, jam, and replay recorded messages over time to prevent triggering alarms and make the operations look normal.

As a result, an attacker can temporarily disable specific LoRa devices or even eliminate select messages.[33]

6.3.2 Security in LPWAN: NB-IoT and LTE-M

The LPWAN Narrow-Band-Internet of Things (NB-IoT) and LTE-M (also referred to as LTE Cat-M1 or Long Term Evolution [4G], category M1) standards have been designed for providing low-power and low-cost IoT communication options using existing cellular networks. NB-IoT, the newest of these standards, and LTE-M are complementary technologies. NB-IoT has a lower bit rate than LTE-M (Table 9) but it is suited well for indoor devices (for use cases such as utility meter reading). LTE-M connects IoT devices and applications directly to a 4G LTE network without a gateway.[34]

	LoRa/LoRaWAN	**NB-IoT**	**LTE-M**
The average module cost (typical metering use case, 10 years)	$6 (module) $3.5 (SOC = System On Chip)	$8-11 module Average $13 for dual mode NB-IoT & LTE-M modules	$10-15 module Average $13 for dual mode LTE-M & NB-IoT modules.
Energy consumption	Very low power consumption. >10 years with metering use case	Low power consumption (10 years+ with metering use case - 200 bytes /day)	LP consumption (10 years+ with metering use case - 200 bytes /day)
Throughput	5 Kbps UL/DL (125 KHz Bandwidth), up to 50 Kbps with channel aggregation.	72 UL / 32 DL Kbps	Max 375 UL/300 Kbps DL Half Duplex
Scalable deployment	National: Specific gateways. Local: Nano Gateway, dongle	National: Roll out by software upgrades or hardware depending on suppliers, on 2G/4G networks	National: Roll out by software upgrades on 4G network
Efficient for fast moving objects	Tested in mobility up to 80 km/h.	Support of cell reselection only	Up to 300 km/h. Full support of Mobility

Identity protection	Partial (Dev/Addr)	Temporary Mobile Subscriber Identity	Temporary Mobile Subscriber Identity
Data integrity	Limited	Optional (with DoNAS)	Limited
Replay protection	Yes	Optional (with DoNAS)	Yes
Key provisioning	Pre-provisioned (ABP) or OTAA	Pre-provisioned or RSP	Pre-provisioned or RSP

Table 9: Comparison of LoRaWAN, NB-IoT and LTE-M General and Security Features

A comparison of LoRaWAN, NB-IoT and LTE-M security features is presented in Table 9. NB-IoT offers the same security and privacy features of LTE mobile networks including support for user identity, data confidentiality, entity authentication, data integrity, and mobile device identification (Table 9). The L2 security includes (1) authentication between UE ((User Equipment) and core network; (2) encryption and integrity protection of both AS and NAS (Non Access Stratum) signaling; (3) encryption of user plane data between the UE and radio network; (4) key management mechanisms to effectively support mobility and UE connectivity mode changes; (5) authentication and core network signaling security as in normal LTE; (6) security supporting optimized transmission of user data.[35]

7. Managed IoT Security Services: IoT Security-as-a-Service

Managed IoT security services are offered as part of the IoT managed service (for example, see Ref. 36) or as a separate service. Managed IoT security solutions should provide security to each layer of the IoT ecosystem. As the article scope and size limitations do not make a detailed IoT manager security services provider (MSSP) discussion possible, we mention only a few providers as examples of the MSSP offerings.

Verizon provides IoT Security Credentialing service that adds an “over-the-top” layer of security, above the customer/client’s existing security.[37] According to Verizon, IoT Security Credentialing offers trusted authentication (the ability to give select employees and/or devices access to apps or IoT devices) and data privacy to help keep data secure through encryption. It uses cryptography techniques to secure communications at the network edge.[37]

Trustwave offers a managed IoT security service to monitor and secure IoT infrastructure and services.[38] The service allows developers and providers of IoT products and services to perform security scanning of embedded devices, interface applications, back-end services, and APIs.

According to **Paladion**, it provides the managed security service with cyber defense capabilities beyond traditional MSSP services as it combines machine learning, artificial intelligence and response automation.[39]

CyFlare, in partnership Solution Synergy 24×7 MSSP services, has developed a managed IoT security solution based on ZingBox IoT Guardian for healthcare organizations.[40]

8. IoT Security in Public Cloud

8.1 Security Features of IoT Cloud Solutions

Integration of the IoT concept with cloud computing results in so-called Cloud of Things (CoT).[41] CoT is able to process and analyze the growing volume of IoT data. The size limits of this article does not allow me to discuss the security of the IoT services offered by public cloud providers — Azure, AWS, Google Cloud Platform, and others — at length. I will outline just some main security features.

8.2 IoT Security in Azure

The Azure IoT Hub within the Azure IoT Suite offers a fully-managed service that enables secure bi-directional communication between IoT devices and Azure services. Per-device security credentials and access control are used. Azure IoT Suite Security can be categorized into three main areas: (1) device provisioning and authentication; (2) secure connectivity; (3) secure processing and storage in the cloud.[42]

Azure IoT supports Device Identity Composition Engine (DICE) and various types of HSMs (for HSM, see Section 5.1.2). DICE is an upcoming standard at Trusted Computing Group for device identification and attestation, which enables manufacturers to use silicon gates for creating device identification based in hardware. The Azure IoT Hub identity registry provides secure storage of device identities and security keys for an IoT solution. The communication path between devices and Azure IoT Hub, or between gateways and Azure IoT Hub, is secured using industry-standard Transport Layer Security (TLS) with Azure IoT Hub authenticated by using X.509 protocol.

The Security Program for Azure IoT is offered by Microsoft.[43] The goal of this service is to assist customers and solution architects in assessing the security of their IoT infrastructure and help find the right security approach for their IoT deployments.

8.3 IoT Security in AWS

AWS Cloud security mechanisms protect data in transit between AWS IoT and other devices or AWS services. A credential is required for each connected device to access the AWS IoT message broker or the Thing Shadows service. All the communications are encrypted by the AWS IoT message broker and Thing Shadows service using TLS. TLS is also used to ensure the confidentiality of the application protocols (MQTT, HTTP) supported by AWS IoT.[44]

Recently AWS has introduced AWS IoT Device Defender, a fully managed IoT security service.[45] AWS IoT Device Defender audits the security policies associated with customers' devices against a set of defined IoT security best practices and identifies security gaps. It is also capable of detecting anomalies in device behavior that may indicate a compromised device. Security alerts generated by AWS IoT Device Defender when a security policy audit fails or when behavior anomalies are detected are published to the AWS IoT Console, Amazon CloudWatch, and Amazon SNS. AWS IoT Device Defender also provides customers with the tools, including contextual information, to help them investigate and mitigate the security problem.

8.4 IoT Security in Google Cloud Platform

Google Cloud Platform (GCP) offers Google Cloud IoT – a set of integrated services for implementing IoT solutions on GCP. Google Cloud IoT Core (beta) announced in 2017 provides a device manager for registering devices with the service and two protocol bridges (MQTT and HTTP) for connecting devices to GCP.[46] Google Cloud IAM roles and permissions are applied to devices to control the access. Industry-standard security protocols provide device data security. Public/private key authentication can be done per device by using JSON Web Tokens (Section 3.1).

9. Security in the Future IoT Systems

9.1 Main Trends in the Next Generation IoT Security

We have considered the current status of the main IoT security domains in the sections above. In this section, we will review the trends in IoT security development. We will briefly consider some emerging technologies that can make the next generation IoT more secure, reviewing the general trends first. Then we will focus on developments in the key IoT security domains — Trust, Data Confidentiality, and Privacy. They are presented in Figure 6 and their current capabilities and limitations are discussed in Section 5. In the present section, we will discuss which new security features and technologies are required to address these limitations in the future.

Holistic security capabilities covering the whole lifecycle of an IoT system and its components are needed for future IoT systems. Development of new threat analytics and risk management as well as self-healing capabilities to detect and defeat potential attacks are required. Collecting, integrating and processing heterogeneous data from different sensors, devices and systems will need new federated identity and access management solutions. Future IoT systems should be able to quickly and appropriately respond to threats and attacks, incorporate and learn from new threat information, and develop and enact thread mitigation plans.[19] The capability to cooperatively diagnose problems and implement security plans for various subsystems in the system, which may be owned by different entities, is also required.

Future IoT systems should also be able to ensure controllable data ownership across enterprise boundaries. To preserve the privacy of customers and/or enterprises while processing a large amount of data, new data analytics algorithms and new cryptographic methods, such as homomorphic or searchable encryption (Sections 9.2.1 and 9.2.2), are needed. Sharing threat intelligence information by different systems enables cooperative security measures that are capable of realizing more cohesive knowledge of the current and future attacks.

Risk assessment and risk management methods for the entire lifecycle of complex IoT systems require new technologies to collect and process security-related data and to perform dynamic and online threat analytics based on that data. New approaches based on machine learning algorithms are needed to perform real-time threat analytics. The required novel threat analytics algorithms must produce warnings with high accuracy and minimal amounts of false positives. They must also be resilient against adversarial attacks that can deliberately compromise and subvert learning data in order to control the behavior of the machine learning algorithms. New cooperative risk management systems and security protocols are required to enable early warning in future IoT systems.

Development of test-based and monitoring-based continuous security audit methods which support dynamic assessment of real-time security levels of IoT systems will be required. These continuous audit methods need to be able to assess various heterogeneous IoT components by using a broad range of solutions, from minimal-invasive, lightweight approaches required for thin devices to comprehensive security evaluations of platforms and the edge components.

New capabilities for tracking data ownership and enforcing data access rules will be an integral part of future IoT platforms. As more data processing is moving to the edge (see Section 2.1.1),

more data anonymization capabilities should be available at the edge and different anonymization algorithms will work on data at different levels. The ability to facilitate and perform anomaly detection at the edge also becomes important.

9.2 Next Generation IoT Security: Data Confidentiality

9.2.1 Homomorphic Encryption

Homomorphic encryption schemes make it possible to perform mathematical operations on ciphertexts. As a result, using fully homomorphic encryption (FHE) data analytics on encrypted data or searching on encrypted data can be performed without revealing search patterns and without actually seeing the original information. An example of the use case for FHE is an analysis of private healthcare IoT data to study the opioid crisis so that the data owners can be assured of data privacy.[47]

9.2.2 Searchable Encryption

Searchable encryption schemes allow a storage provider to search for keywords or patterns in encrypted data. While keyword searches can be performed, the stored data cannot be decrypted and it is not possible to gain any knowledge of the underlying plaintext.

9.3 Next Generation IoT Security: Trust

9.3.1 Trust Establishment

In most IoT scenarios trust must be established ad-hoc with previously unregistered and unknown peers, and without user interaction. This requires new and lightweight trust establishment algorithms. Current trust establishment solutions mainly focus on establishing trust in public keys and their assignment to users (Section 5.1). Future IoT solutions will also need trust in transactions and agreements (Section 9.3.2), as well as trust in the integrity of devices and platforms (Section 9.3.3).

9.3.2 Blockchain and IoT: Trust in Transactions

Blockchain-based protocols that are gaining popularity can address the challenge of establishing trust. One of the key building blocks of future IoT trust infrastructures can be smart contracts based on blockchains, as they are a prerequisite for business-critical interaction between devices without direct human interaction. However, blockchains require computational resources and have high bandwidth overhead. This limits their use in IoT and new lightweight blockchain-based technologies are needed.

9.3.3 Trust in Platforms

Two approaches on automated establishment of trust in remote platforms exist: hardware and software remote attestation. Hardware remote attestation has high costs as it uses specific hardware modules such as HSMs (Section 5.1.2) which may be prohibitive for low-cost sensor hardware. Furthermore, additional resource consumption by such hardware is not acceptable for many battery-powered devices. Software remote attestation can provide an acceptable protection level for most applications but it cannot conceptually guarantee trust in the overall platform. Further development of code obfuscation, white-box cryptography, and control-flow integrity technologies can provide holistic software-only remote attestations in the future.

9.3.4 Identity Management

The existing identity and access management systems that we considered in Section 5.1.3 provide secure, integrated management of data from different devices and systems. In the future, autonomous data exchanges among different entities are expected to be controlled based on advanced security and trust management technologies, e.g. usage control (Section 9.4.1).

9.4 Next Generation IoT Security: Privacy

9.4.1 Privacy Through Data Usage Control

Data usage control is an extension of traditional access control concepts. Future data usage control technologies will extend traditional access control concepts to track and label data as it is processed by various systems. They will define fine-granular usage restrictions in order to enforce privacy properties over large data sets while still allowing for running learning algorithms and analytics over them**.**

The key advantage of data usage control is that it provides users with the ability to control the usage of their data even when it is managed by others. This will help to meet legal requirements in many jurisdictions (for example, General Data Protection Regulation [GDPR] in the European Union). Future IoT system implementations will need to be able to locally control data exposure and to interface with a variety of other systems while maintaining end-to-end privacy guarantees.

9.4.2 Privacy in Multifaceted and Dynamic Contexts

When services from a utility company, the device manufacturer or an application provider access the data, it results in additional attack surfaces for breaching confidentiality of the user data. From the data owner’s point of view, services with consensual access to user data are still all potential adversaries. As more data is being stored, transmitted and processed via shared infrastructure,

future IoT platforms will require new advanced services and technologies to enforce adequate access controls.

10. Conclusion

This article provides an overview of IoT security threats, solutions for addressing them, and new evolving technologies. It shows the paramount importance of security in developing viable IoT solutions. I hope my article will help you in selecting secure IoT technologies for your organization.

11. References

1. http://en.wikipedia.org/wiki/Kevin_Ashton
2. http://www.itu.int/pub/S-POL-IR.IT-2005/e
3. ISO/IEC JTC 1, Internet of Things (IoT), Geneva, 2014.
4. J. Voas. Networks of 'Things.' NIST Special Publication 800-183. 2016.
5. http://www.m2mzone.com/bergmac
6. IERC Cluster SRIA 2014 – Internet of Things.
7. http://www.openfogconsortium.org; Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. Cisco White Paper, 2015.
8. https://www.edgexfoundry.org
9. H. Suo, J. Wan, C. Zou, and J. Liu, Security in the internet of things: a review. In Computer Science and Electronics Engineering (ICCSEE), p. 648, 2012.
10. International Telecommunication Union – Telecommunication Sector, Series Y: Global Information Infrastructure, Internet Protocol Aspects and next Generation Networks - Frameworks and functional architecture models - Overview of the Internet of things, Y.2060", June 2012.
11. O. Vermesan and P. Friess, Eds. ERC Cluster SRIA 2014 – Internet of Things – From Research and Innovation to Market Deployment. River Publishers Series in Communication, 2014.
12. The Internet of Things Reference Model. Cisco, June 2014.
13. K. Laeeq and J. A. Shamsi. A Study of Security Issues, Vulnerabilities, and Challenges in the Internet of Things. In Securing Cyber-Physical Systems. Taylor and Francis. Oct 2015.
14. N. Jeyanthi. Internet of Things (IoT) as Interconnection of Threats (IoT). In: Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations. Fei Hu (Ed). CRC Press, 2016.

15. Shancang Li, Li Da Xu. Securing the Internet of Things. Syngress, 2017.
16. R. H. Ahmad, Al-Sakib K. Pathan. A Study on M2M (Machine to Machine) System and Communication: Its Security, Threats, and Intrusion Detection System. In: The Internet of Things: Breakthroughs in Research and Practice. Information Resources Management Association. IGI Global, 2017
17. D. Miorandi, S. Sicari, F.D. Pellegrini, and I. Chlamtac, I. Internet of Things: Vision, applications and research challenges. Ad Hoc Networks v.10, p.1497, 2012.
18. M. Schukat, P. C. Castilla, and H. Melvin. Trust and Trust Models for the IoT. In: Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations. Fei Hu (Ed). CRC Press, 2016.
19. IoT 2020: Smart and secure IoT platform. IEC White Paper. http://www.iec.ch/whitepaper/iotplatform
20. X. Lu, Z. Qu, Q. Li, and P. Hui, P. Privacy Information Security Classification for Internet of Things Based on Internet Data. International Journal of Distributed Sensor Networks, 11(8), 932941, 2015.
21. J. Kanniappan and B. Rajendiran. Privacy in the Internet of Things. In Lee (Ed.). The Internet of Things in the Modern Business Environment. IGI Global, 2017.
22. Practical BLE Throughput. Rigado LLC, 2016. www.rigado.com/modules.
23. http://atmosphere.anaren.com/wiki/Data_rates_using_BLE.
24. V. Anitta, F. Fincy, and P.S. Ayyappadas. Security Aspects in 6lowpan Networks. IOSR Journal of Electronics and Communication Engineering. v.10, p.8, 2015.
25. R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle. 6LoWPAN fragmentation attacks and mitigation mechanisms . Proceeding WiSec '13 Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks. pp. 55-66, 2013.
26. https://www.theregister.co.uk/2016/11/10/iot worm can hack philips hue lightbulbs spread across cities
27. A. Khattab et al. RFID Security, Analog Circuits and Signal Processing. Springer International Publishing AG, pp.27-40, 2017.
28. http://datatracker.ietf.org/wg/lpwan/documents
29. http://www.lora-alliance.org/technology
30. http://partner.orange.com/lora-in-a-nutshell
31. LoRaWAN Security. White Paper. LoRa Alliance. Feb. 2017.

32. R. Miller. LoRa Security. Building a Secure LoRa Solution. MWR Labs White paper. 2016. http://labs.mwrinfosecurity.com/publications/lo
33. E. Aras, N. Small, G. S. Ramachandran, S. Delbruel, W. Joosen and D. Hughes. Selective Jamming of LoRaWAN using Commodity Hardware. In the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. ACM, New York, 2017.
34. LPWA Technology Security Comparison. White Paper. Franklin Heath, 2017
35. R. P. Jover. Security and Impact of the Internet of Things (IoT) on Mobile Networks. In: Security and Privacy in Internet of Things (IoT): Models, Algorithms, and Implementations. Fei Hu (Ed). CRC Press, 2016.
36. http://marketing.integron.com/managed-iot-services.html
37. http://www.verizonenterprise.com/solutions/integrated-managed-services/managed-services/iot-security-credentialing
38. http://www.trustwave.com/Services
39. http://www.paladion.net
40. http://cyflare.com
41. B. Alohali, Security in Cloud of Things (CoT). In: Managing Big Data in Cloud Computing Environments. IGI Global, 2016.
42. http://docs.microsoft.com/en-us/azure/iot-suite
43. http://blogs.microsoft.com/microsoftsecure/2016/10/26/securing-the-internet-of-thingsintroducing- the-security-program-for-azure-iot
44. http://aws.amazon.com/blogs/iot/understanding-the-aws-iot-security-model
45. http://aws.amazon.com/iot-device-defender
46. http://cloud.google.com/iot-core
47. http://www.networkworld.com/article/3196121/security/how-to-make-fully-homomorphic-encryption-practical-and-usable.html