Chapter 23

Internet of Things (IoT) Security

The Internet of Things

- The *Internet of Things (IoT)* is a term that refers to the expanding interconnection of smart devices
- A dominant theme is the embedding of short-range mobile transceivers into a wide array of gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves
- The Internet now supports the interconnection of billions of industrial and personal objects, usually through cloud systems
- The objects deliver sensor information, act on their environment, and in some cases modify themselves, to create overall management of a larger system

IoT

- The IoT is primarily driven by deeply embedded devices
- These devices are low-bandwidth, low-repetition data capture and low-bandwidth data-usage appliances that communicate with each other and provide data via user interfaces
- Embedded appliances, such as high-resolution video security cameras, video VoIP phones, and a handful of others, require high-bandwidth streaming capabilities
- Countless products simply require packets of data to be intermittently delivered

Evolution

- The Internet has gone through roughly four generations of deployment culminating in the IoT:

  1. **Information technology (IT):** PCs, servers, routers, firewalls, and so on, bought as IT devices by enterprise IT people, primarily using wired connectivity
  2. **Operational technology (OT):** Machines/appliances with embedded IT built by non-IT companies, such as medical machinery, SCADA (supervisory control and data acquisition), process control, and kiosks, bought as appliances by enterprise OT people and primarily using wired connectivity
  3. **Personal technology:** Smartphones, tablets, and eBook readers bought as IT devices by consumers (employees) exclusively using wireless connectivity and often multiple forms of wireless connectivity
  4. **Sensor/actuator technology:** Single-purpose devices bought by consumers, IT, and OT people exclusively using wireless connectivity, generally of a single form, as part of larger systems

- It is the fourth generation that is usually thought of as the IoT, and which is marked by the use of billions of embedded devices

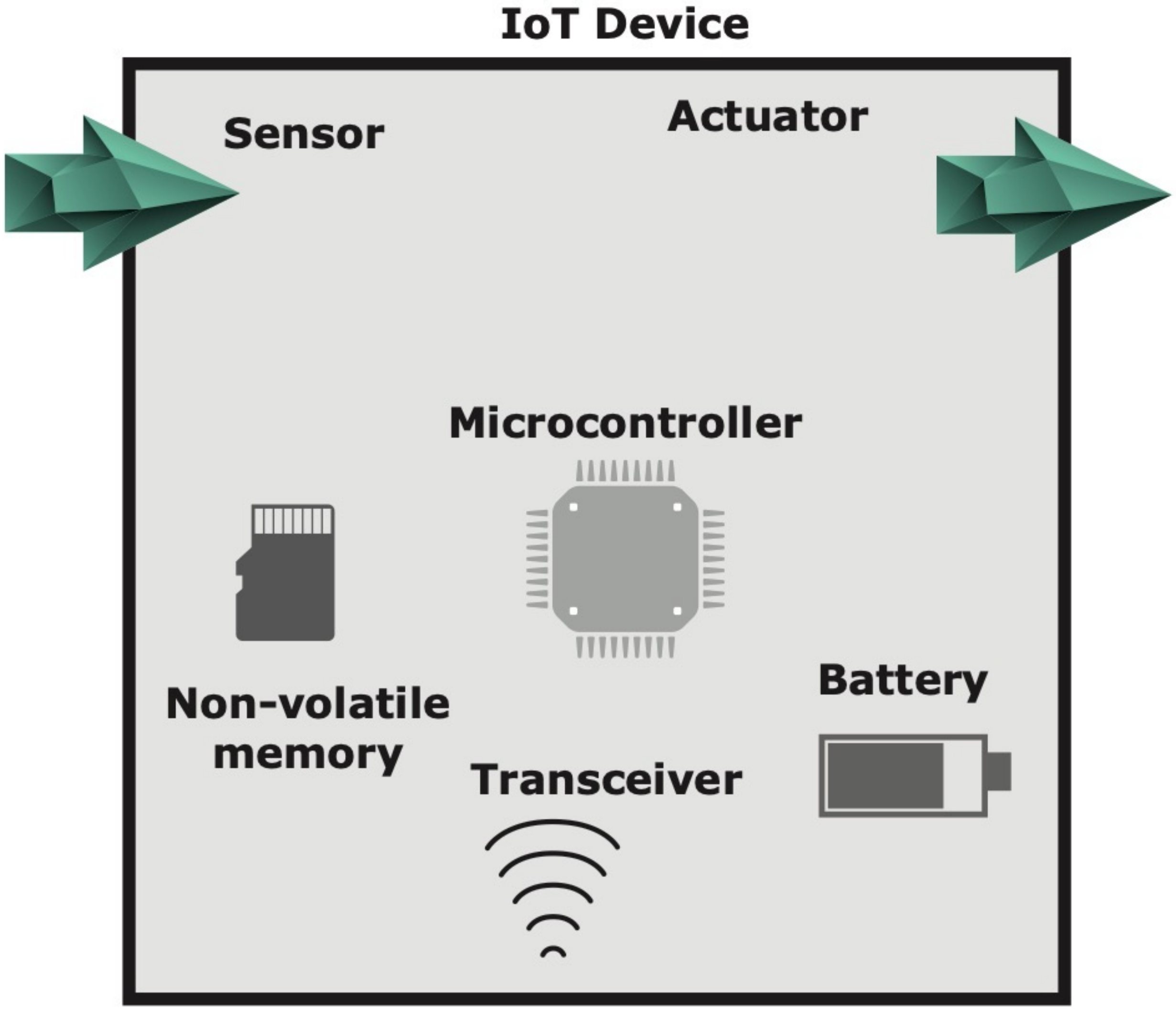


Figure 23.1 IoT Components

IoT and RFID

- Radio-Frequency Identification (RFID)
- RFID technology, which uses radio waves to identify items, is increasingly becoming an enabling technology for IoT
- The main elements of an RFID system are tags and readers
  - RFID tags are small programmable devices used for object, animal, and human tracking
    - They come in a variety of shapes, sizes, functionalities, and costs
  - RFID readers acquire and sometimes rewrite information stored on RFID tags that come within operating range
    - Readers are usually connected to a computer system that records and formats the acquired information for further uses

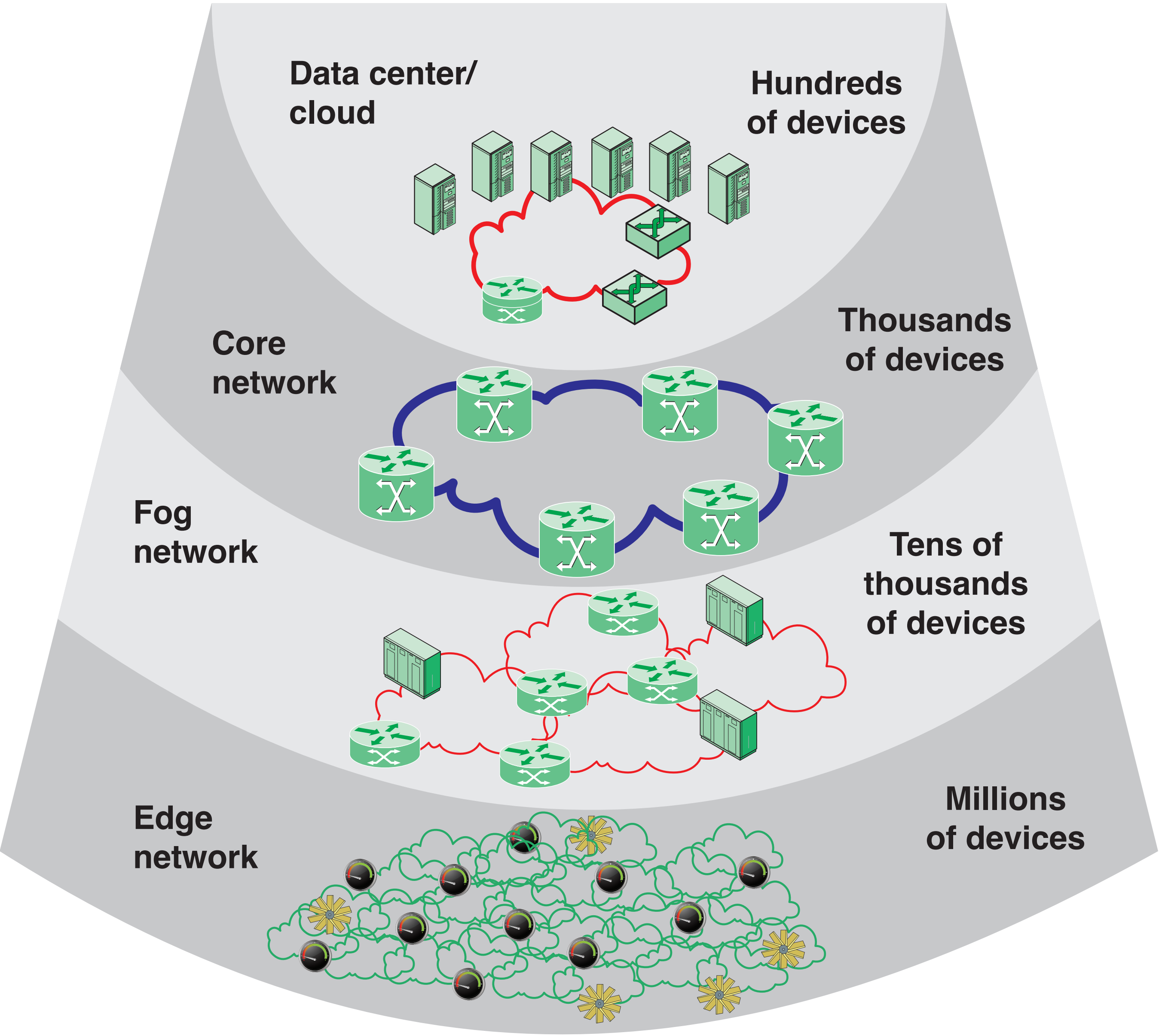


Figure 23.2 The IoT/Cloud Context

Edge

- At the **edge** of a typical enterprise network is a network of IoT-enabled devices, consisting of sensors and perhaps actuators
  - These devices may communicate with one another
- A gateway interconnects the IoT-enabled devices with the higher-level communication networks
  - It performs the necessary translation between the protocols used in the communication networks and those used by devices
  - It may also perform a basic data aggregation function

Fog

In many IoT deployments, massive amounts of data may be generated by a distributed network of sensors

Rather than store all of that data permanently (or at least for a long period) in central storage accessible to IoT applications, it is often desirable to do as much data processing close to the sensors as possible

The purpose of what is sometimes referred to as the *edge* computing level is to convert network data flows into information that is suitable for storage and higher level processing

Processing elements at these level may deal with high volumes of data and perform data transformation operations, resulting in the storage of much lower volumes of data

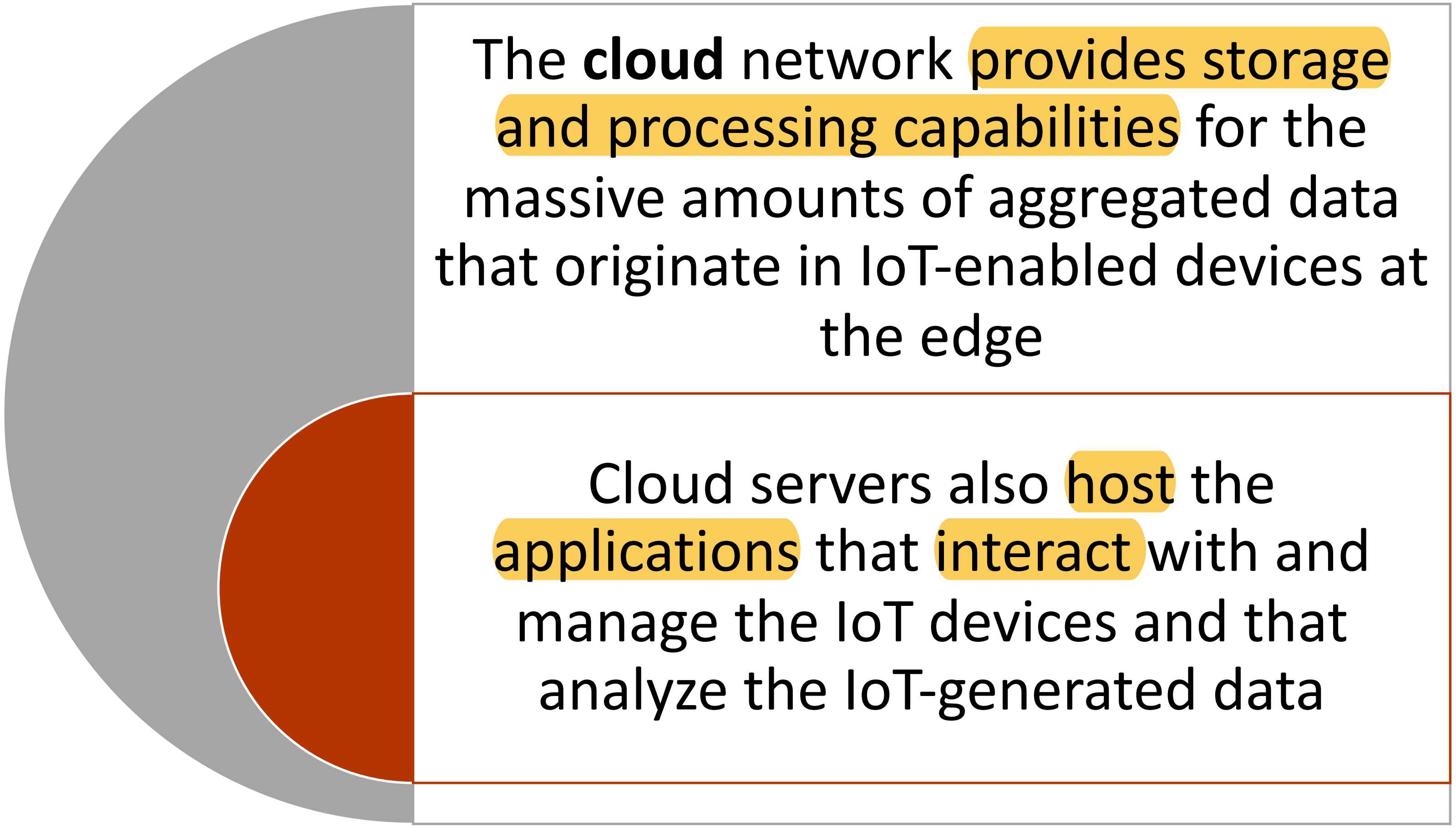
Fog

- The following are examples of **fog** computing operations:
- **Evaluation:** Evaluating data for criteria as to whether it should be processed at a higher level
- **Formatting:** Reformatting data for consistent higher-level processing
- **Expanding/decoding:** Handling cryptic data with additional context (such as the origin)
- **Distillation/reduction:** Reducing and/or summarizing data to minimize the impact of data and traffic on the network and higher-level processing systems
- **Assessment:** Determining whether data represents a threshold or alert; this could include redirecting data to additional destinations

Core

- The **core** network, also referred to as a **backbone network**, connects geographically dispersed fog networks as well as provides access to other networks that are not part of the enterprise network
- Typically, the core network will use very high performance routers, high-capacity transmission lines, and multiple interconnected routers for increased redundancy and capacity
- The core network may also connect to high-performance, high-capacity servers, such as large database servers and private cloud facilities
- Some of the core routers may be purely internal, providing redundancy and additional capacity without serving as edge routers

Cloud



The **cloud** network provides storage and processing capabilities for the massive amounts of aggregated data that originate in IoT-enabled devices at the edge

Cloud servers also host the applications that interact with and manage the IoT devices and that analyze the IoT-generated data

Table 23.1 Comparison of Cloud and Fog Features

	Cloud	Fog
Location of processing/storage resources	Center	Edge
Latency	High	Low
Access	Fixed or wireless	Mainly wireless
Support for mobility	Not applicable	Yes
Control	Centralized/hierarchical (full control)	Distributed/hierarchical (partial control)
Service access	Through core	At the edge/on handheld device
Availability	99.99%	Highly volatile/highly redundant
Number of users/devices	Tens/hundreds of millions	Tens of billions
Main content generator	Human	Devices/sensors
Content generation	Central location	Anywhere
Content consumption	End device	Anywhere
Software virtual infrastructure	Central enterprise servers	User devices

(Table is on page 712 in the textbook)

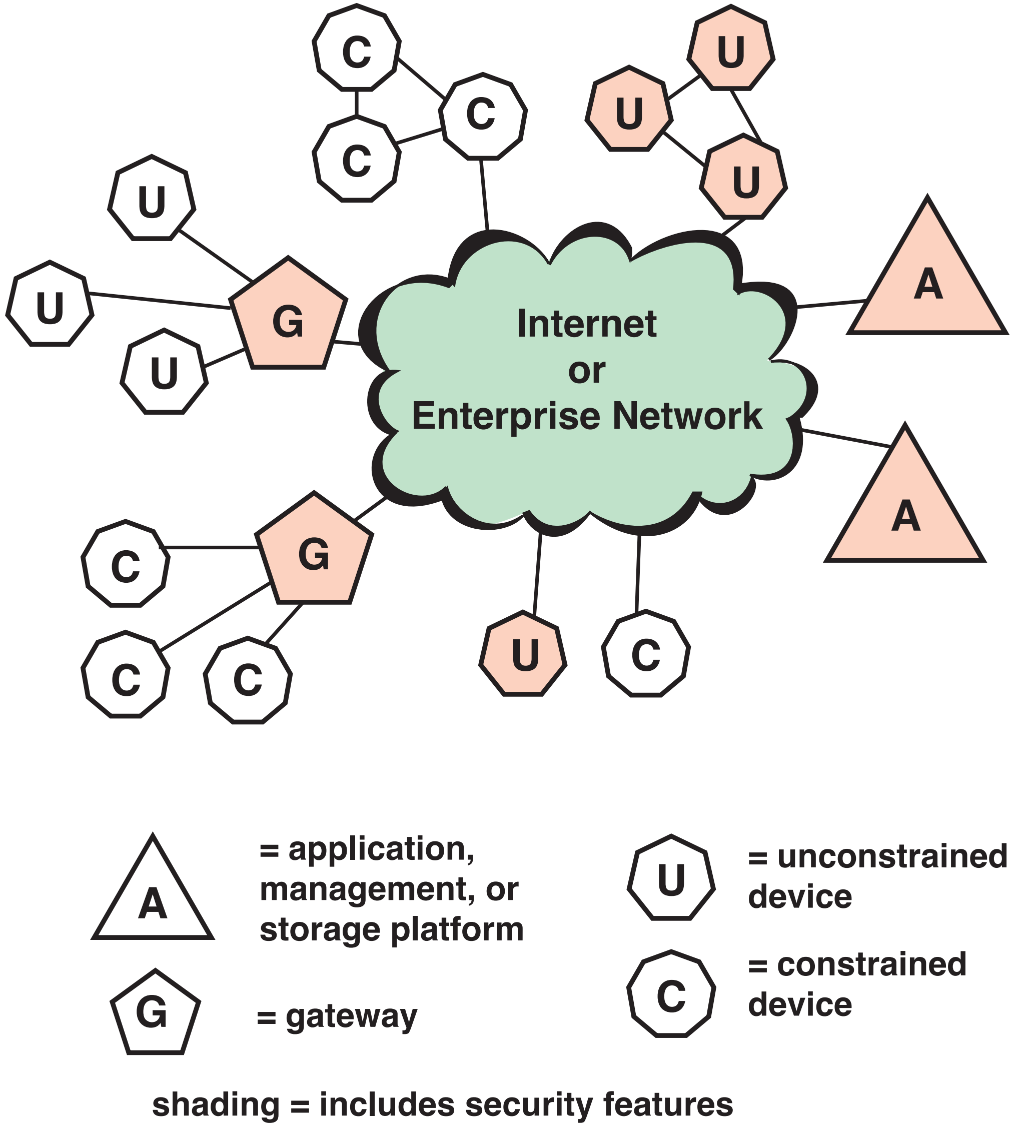


Figure 23.3 IoT Security: Elements of Interest

Development Issues

- Very large attack surfaces
- Limited device resources
- Complex ecosystem
- Fragmentation of standards and regulations
- Widespread deployment
- Security integration

- Safety aspects
- Low cost
- Lack of expertise
- Security updates
- Insecure programming
- Unclear liabilities

IoT Security Objectives

- **Restricting logical access to the IoT network**
  - This may include: using unidirectional gateways, using firewalls to prevent network traffic from passing directly between the corporate and IoT networks, and having separate authentication mechanisms and credentials for users of the corporate and IoT networks
  - An IoT system should also use a network topology that has multiple layers, with the most critical communications occurring in the most secure and reliable layer
- **Restricting physical access to IoT network and components**
  - A combination of physical access controls should be used, such as locks, card readers, and/or guards

IoT Security Objectives

- **Protecting individual IoT components from exploitation**
  - This includes deploying security patches in as expeditious a manner as possible, after testing them under field conditions; disabling all unused ports and services and assuring that they remain disabled; restricting IoT user privileges to only those that are required for each person’s role; tracking and monitoring audit trails; and using security controls such as antivirus software and file integrity checking software where technically feasible
- **Preventing unauthorized modification of data.**
  - This includes data that are in transit (at least across the network boundaries) and at rest
- **Detecting security events and incidents**
  - The object is to detect security events early enough to break the attack chain before attackers attain their objectives. This includes the capability to detect failed IoT components, unavailable services, and exhausted resources that are important to provide proper and safe functioning of an IoT system

IoT Security Objectives

- **Maintaining functionality during adverse conditions**
  - This involves designing IoT systems so that each critical component has a redundant counterpart
  - If a component fails, it should fail in a manner that does not generate unnecessary traffic on IoT or other networks, or does not cause another problem elsewhere
  - IoT system 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
- **Restoring the system after an incident**
  - Incidents are inevitable and an incident response plan is essential
  - A major characteristic of a good security program is how quickly the IoT system can be recovered after an incident has occurred

Tamper Resistance and Detection

Tampering

- An unauthorized modification that alters the intended functioning of a system or device in a way that degrades the security it provides

Tamper resistant

- A characteristic of a system component that provides passive protection against an attack

Tamper detection

- Techniques to ensure that the overall system is made aware of unwanted physical access

Tamper Resistance

- The common approach to tamper resistance is to use specialized physical construction materials to make tampering with a fog node difficult
    - Examples include hardened steel enclosures, locks, and security screws
- A second category of tamper resistance is the deterrence of tampering by ensuring that tampering leaves visible evidence behind
    - Examples include special seals and tapes that make it obvious when there has been physical tampering

Tamper Detection

- Mechanisms for tamper detection include:
- **Switches**
  - A variety of switches, such as mercury switches, magnetic switches, and pressure contacts can detect the opening of a device, the breach of a physical security boundary, or the movement of a device
- **Sensors**
  - Temperature and radiation sensors can detect environmental changes. Voltage and power sensors can detect electrical attacks
- **Circuitry**
  - It is possible to wrap components with flexible circuitry, resistance wire, or fiber optics so as to detect a puncture or break

Security Requirements

Communication security:

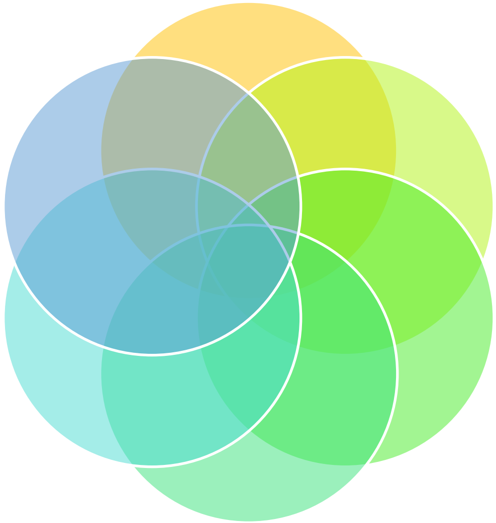
- Secure, trusted, and privacy-protected communication capability is required so that unauthorized access to the content of data can be prohibited, integrity of data can be guaranteed, and privacy-related content of data can be protected during data transmission or transfer in IoT

Data management security:

- Secure, trusted, and privacy-protected data management capability is required so that unauthorized access to the content of data can be prohibited, integrity of data can be guaranteed, and privacy-related content of data can be protected when storing or processing data in IoT

Security audit:

- Security audit is required to be supported in IoT. Any data access or attempt to access IoT applications are required to be fully transparent, traceable, and reproducible according to appropriate regulation and laws



Mutual authentication and authorization:

- Before a device can access the IoT, mutual authentication and authorization between the device and IoT is required to be performed according to predefined security policies

Service provision security:

- Secure, trusted, and privacy-protected service provision capability is required, so that unauthorized access to service and fraudulent service provision can be prohibited and privacy information related to IoT users can be protected

Integration of security policies and techniques:

- The ability to integrate different security policies and techniques is required to ensure a consistent security control over the variety of devices and user networks in IoT

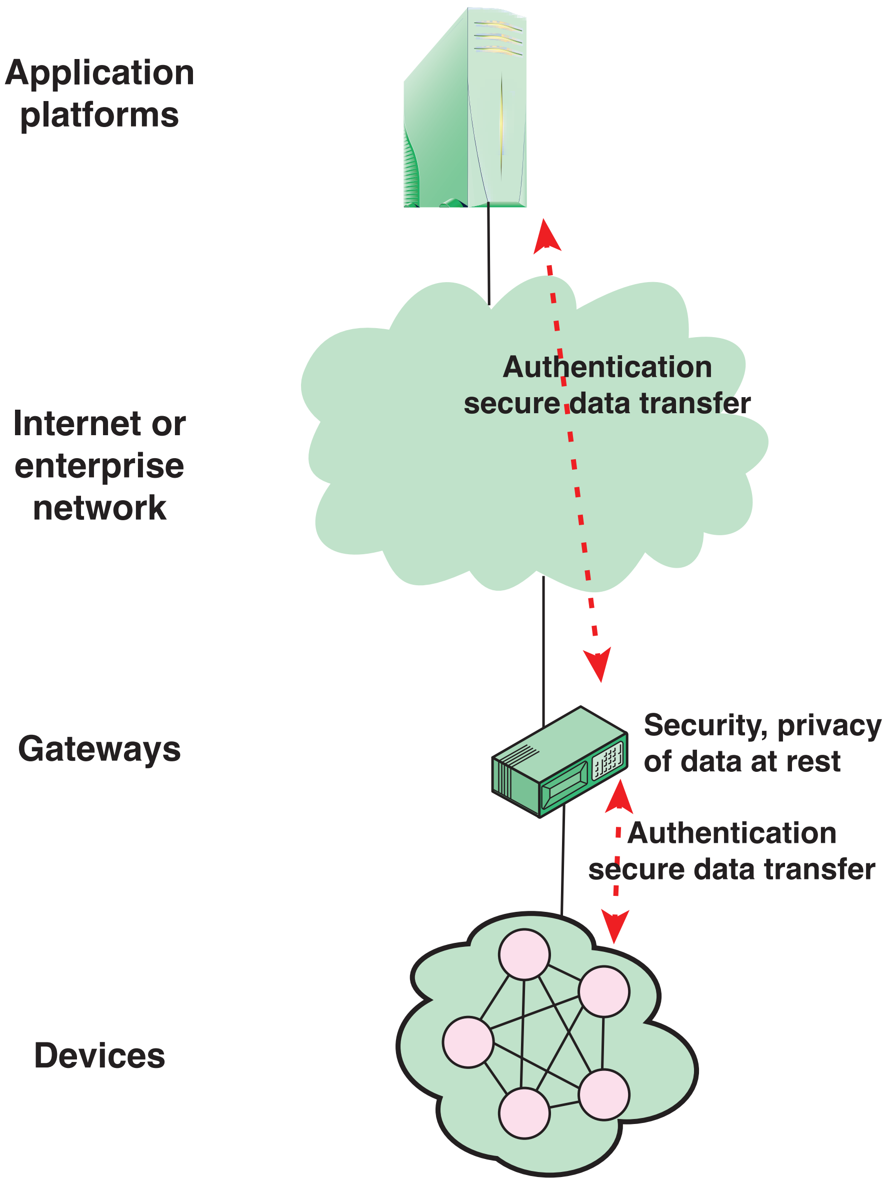

Application platforms
Internet or enterprise network
Authentication secure data transfer
Gateways
Security, privacy of data at rest
Authentication secure data transfer
Devices

Figure 23.4 IoT Gateway Security Functions

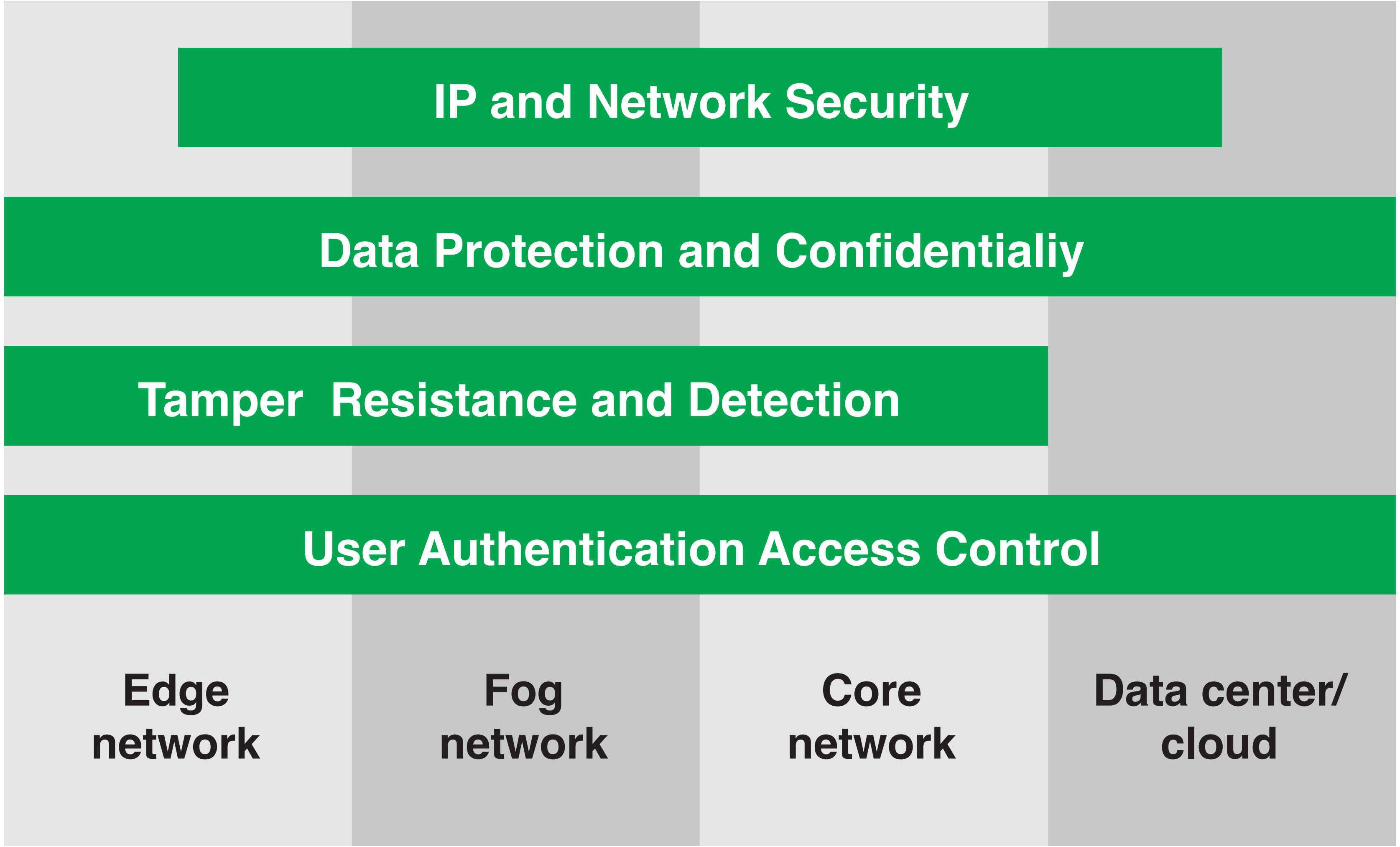

IP and Network Security
Data Protection and Confidentialiy
Tamper Resistance and Detection
User Authentication Access Control
Edge network
Fog network
Core network
Data center/ cloud

Figure 23.5 IoT Security Environment

MiniSec

- An open-source security module that is part of the *TinyOS* operating system
    - *TinyOS* is designed for small embedded systems with tight requirements on memory, processing time, real-time response, and power consumption

- **MiniSec** is designed to be a link-level module that offers a high level of security, while simultaneously keeping energy consumption low and using very little memory

- **MiniSec** provides confidentiality, authentication, and replay protection

MiniSec

- **MiniSec** has two operating modes:
  - One tailored for single-source communication
  - One tailored for multi-source broadcast communication
- **MiniSec** is designed to meet the following requirements:
  - Data authentication
  - Confidentiality
  - Replay protection
  - Freshness
  - Low energy overhead
  - Resilient to lost messages

Skipjack

- Encryption algorithm used by MiniSec
- Was developed in the 1990s by the U.S. National Security Agency (NSA)
- Is one of the simplest and fastest block cipher algorithms, which is critical to embedded systems
- Makes use of an 80-bit key
- With its efficient computation and low memory footprint, Skipjack is an attractive choice for IoT devices

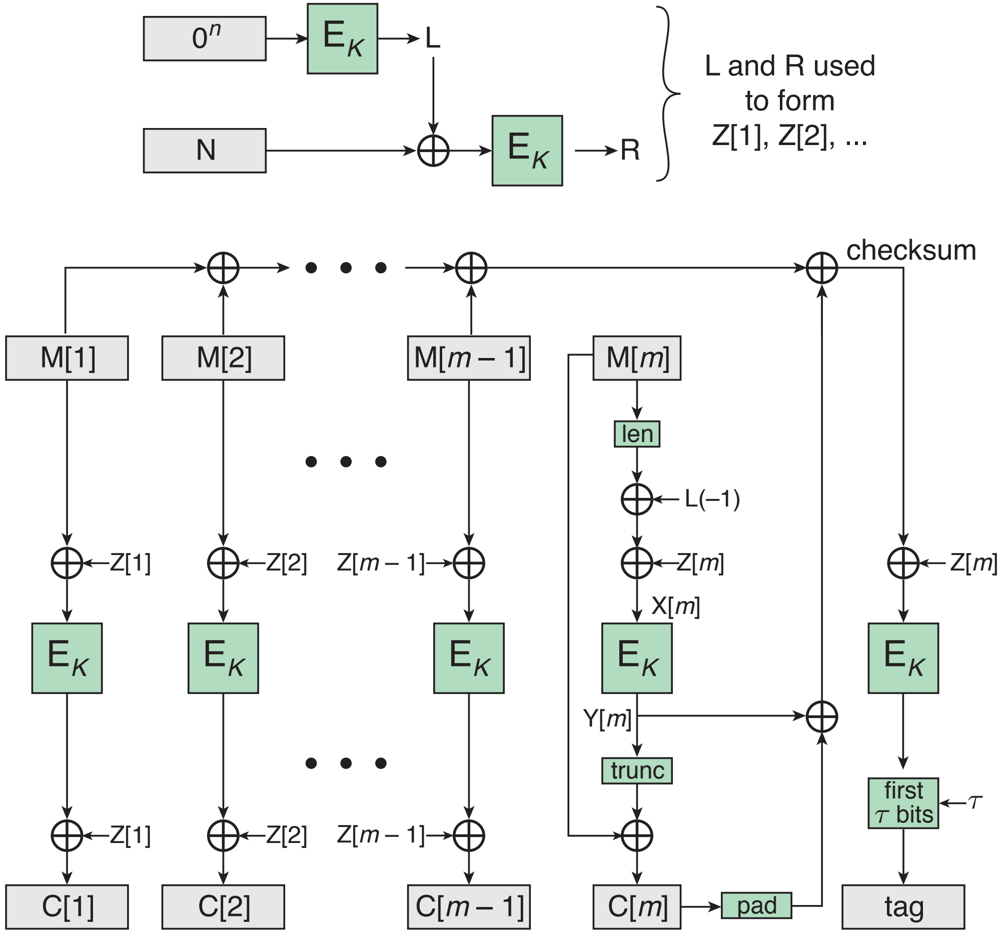
Uses an 80-bit key on 64-bit data blocks, using an unbalanced Feistel network with 32 rounds, designed to be used in secured phones. In March 2016, NIST no longer certifies Skipjack for US government applications.

Operating Modes

- MiniSec has two operating modes: unicast (MiniSec-U) and broadcast (MiniSec-B)
- Both schemes use OCB with a counter that is input along with the plaintext into the encryption algorithm
- The least significant bits of the counter are also sent as plaintext to enable synchronization
- For both modes, data are transmitted in packets
- Each packet includes the encrypted data block, the OCB authentication tag, and the MiniSec counter
- MiniSec-U employs synchronized counters, which require the receiver to keep a local counter for each sender
- Once a receiver observes a counter value, it rejects packets with an equal or smaller counter value; therefore, an attacker cannot replay any packet that the receiver has previously received

Operating Modes

- MiniSec-U cannot be directly used to secure broadcast communication
  - It would be too expensive to run the counter resynchronization protocol among many receivers
  - If a node were to simultaneously receive packets from a large group of sending nodes, it would need to maintain a counter for each sender, resulting in high memory overhead
- Instead, it uses two mechanisms, a timing-based approach and a bloom-filter approach, that defend against replay attacks
- The timing approach is augmented with a bloom-filter approach in order to prevent replay attacks inside the current epoch
- Every time that a node receives a message, it checks if it belongs to its bloom filter
- If the message is not replayed, it is stored in the bloom filter; else, the node drops it



n = block length in bits
N = nonce
len(M[m]) = length of M[m] represented as an n-bit integer
trunc(Y[m]) = deletes least significant bits so that result is same length as M[m]
pad = pad with least significant 0 bits to length n
τ = length of authentication tag

Offset codebook mode (OCB mode)

Figure 23.6 OCB Encryption and Authentication

```
**algorithm** OCB-Encrypt$_K$(N, M)
Partition M into M[1]…M[m]
$L \leftarrow L(0) \leftarrow E_K(0^n)$
$R \leftarrow E_K(N \oplus L)$
**for** $i \leftarrow 1$ **to** $m$ **do** $L(i) \leftarrow 2 \cdot L(i-1)$
$L(-1) = L \cdot 2^{-1}$
$Z[1] \leftarrow L \oplus R$
**for** $i \leftarrow 2$ **to** $m$ **do** $Z[i] \leftarrow Z[i-1] \oplus L(ntz(i))$
**for** $i \leftarrow 1$ **to** $m - 1$ **do**
    $C[i] \leftarrow E_K(M[i] \oplus Z[i]) \oplus Z[i]$
$X[m] \leftarrow len(M[m]) \oplus L(-1) \oplus Z[m]$
$Y[m] \leftarrow E_K(X[m])$
$C[m] \leftarrow M[m] \oplus$ (first len(M[m]) bits of Y[m])
Checksum $\leftarrow$
    $M[1] \oplus \ldots \oplus M[m-1] \oplus C[m]0^* \oplus Y[m]$
Tag $\leftarrow E_K(Checksum \oplus Z[m])$ [first $\tau$ bits]
```

```
**algorithm** OCB-Decrypt$_K$(N, M)
Partition M into M[1]…M[m]
$L \leftarrow L(0) \leftarrow E_K(0^n)$
$R \leftarrow E_K(N \oplus L)$
**for** $i \leftarrow 1$ **to** $m$ **do** $L(i) \leftarrow 2 \cdot L(i-1)$
$L(-1) = L \cdot 2^{-1}$
$Z[1] \leftarrow L \oplus R$
**for** $i \leftarrow 2$ **to** $m$ **do** $Z[i] \leftarrow Z[i-1] \oplus L(ntz(i))$
**for** $i \leftarrow 1$ **to** $m - 1$ **do**
    $M[i] \leftarrow D_K(C[i] \oplus Z[i]) \oplus Z[i]$
$X[m] \leftarrow len(M[m]) \oplus L(-1) \oplus Z[m]$
$Y[m] \leftarrow E_K(X[m])$
$M[m] \leftarrow$ (first len(C[m]) bits of Y[m]) $\oplus C[m]$
Checksum $\leftarrow$
    $M[1] \oplus \ldots \oplus M[m-1] \oplus C[m]0^* \oplus Y[m]$
Tag' $\leftarrow E_K(Checksum \oplus Z[m])$ [first $\tau$ bits]
```

Figure 23.7 OCB Algorithms

Summary

- Explain the scope of the Internet of Things
- List and discuss the five principal components of IoT-enabled things
- Understand the relationship between cloud computing and IoT

- Define the patching vulnerability
- Explain the IoT Security Framework
- Understand the MiniSec security feature for wireless sensor networks