IOT Lecture 5 Notes

Lecture Outline

- Introduction to RFID Protocol
- Security Challenges in RFID
- Introduction to ZigBee Protocol
- Security in ZigBee
- Introduction to Bluetooth Protocol
- Security Challenges in Bluetooth

RFID

- **Identification** is essential for IoT implementation.
- The RFID community coined "Internet of Things" to describe discovering info about tagged objects through Internet addresses/databases.
- Forms the base of Web 4.0 (Web of Things).
- **RFID** = Radio Frequency Identification
- Uses radio waves to identify and track objects.
- A **tag/transponder** (microchip + antenna) is attached to an object.
- The **reader** emits radio waves to power the tag and read its ID.

RFID Architecture

1. **Tags** (Transmitters/Responders): Microchip + antenna, attached to objects.
2. **Readers** (Transmitters/Receivers): Use radio waves to communicate with tags.
3. **Controller/Host**: PC/workstation running the database and middleware.

RFID Spectrum

- Tags transmit data wirelessly when triggered by a reader.
- **Passive tags** don't need a power source (a key advantage).
- Advantages over barcodes:
  - No line of sight required
  - High-speed and multiple reads
  - Read/write capability
  - Unit-specific ID

Frequency	Range	Example Application
125kHz (LF)	Few cm	Auto-immobilizer
13.56MHz (HF)	~1m	Building access
900MHz (UHF)	~7m	Supply chain
2.4GHz (Microwave)	~10m	Traffic toll collection

How RFID Works

- **Near Field (LF, HF)**: Uses **inductive coupling** (magnetic flux induces current in tag).
- **Far Field (UHF, Microwave)**: Uses **backscatter** (modulating antenna impedance).
- Energy loss is **$1/R^3$** in near field and **$1/R$** in far field.
- **Boundary**: $R = \text{wavelength} / 2\pi$

RFID Standards

Standard	Frequency	Range	Application
EPCglobal Class 1 Gen 2	860–960 MHz	~30 ft	Supply chain, inventory control
ISO 15693	13.56 MHz	~3 ft	Access control, libraries, asset tracking
ISO 14443	13.56 MHz	~4 in	NFC: Mobile payments, transit
ISO 18000-6C	860–960 MHz	~30 ft	Asset tracking, logistics, vehicle ID

EPC (Electronic Product Code) Structure

- Header: Tag version
- EPC Manager: Manufacturer ID
- Object Class: Product ID
- Serial Number: Unique unit ID
- 96-bit EPC: 268M companies × 16M products × 687B units

Types of RFID Tags

Passive Tags

- No battery or communication ability
- Only respond to reader commands
- Include an integrated circuit and antenna
- **Semi-passive** tags include an on-board power source

Active Tags

- Have own power source (battery or light-powered)
- Broadcast their own signals (like a phone)
- Longer range than passive tags

RFID Security Challenges

- **Unauthorized access**: Tags can be read by attackers
- **Data tampering**: Info on tag can be changed
- **Denial of Service**: Signal jamming or antenna blocking
- **Eavesdropping**: Signal capture by attackers
- **Malicious attacks**: Hacking to steal or disrupt
- **Other threats**: Cloning, Tracing, Data forging

RFID Security: IPSec

- **IPSec**: Secures IPv6 connections; optional in IPv4
- Security can be applied between:
  - Two nodes
  - Two security gateways
  - Gateway and node

IPSec Methods

- **Authentication Header (AH)**: Verifies origin + integrity, prevents replay
- **Encapsulating Security Payload (ESP)**:
  - Adds confidentiality
  - Can use tunnel mode to protect full packet (including headers)

RFID Security: Cryptography (Cloning)

Symmetric-Key Authentication Protocol

1. Tag sends its ID
2. Reader sends a random nonce
3. Tag responds with encrypted nonce
4. Reader verifies response using shared key

RFID Privacy Solutions

Kill & Sleep

- Tags can be permanently deactivated using a PIN-protected "kill" command
- “Sleep” to pause functionality

Renaming

- Encrypting identifiers isn’t enough; they must **change over time** to avoid tracking.

Relabeling

- Consumer can relabel tag ID
- Old tags can be reused for public services like recycling

Minimalist Cryptography

- Tag holds rotating pseudonyms
- Authorized readers can match all; unauthorized ones can’t track consistently
- Prevent pseudonym harvesting via rapid-fire reads

Standards Challenges

- **International collaboration** (IEEE, IETF, ISO/IEC, etc.) is essential for IoT standardization
- Examples:
    - **ISO/IEC 29167-11:2023**: PRESENT-80 crypto suite
    - **ISO/IEC 29167-10:2017**: AES-128 crypto suite
- Identity management is critical in IoT (smart cards, RFID, IPv6 will help)

Technical Challenges

- Products with **metal/liquids** block reads
- **Multiple readers** nearby may cause:
  - False reads
  - Reader signal interference
- Requires **middleware** to coordinate reads and manage shelf data

---

RFID Security Best Practices

- **Authentication**: Use encrypted protocols and passwords
- **Encryption**: Secure tag-reader communications
- **Access Control**: Restrict physical access using cards or biometrics

---

RFID Hacking Tools

1. **Proxmark3 ID Dev Kit**:
   - Research tool for sniffing, analyzing, emulating RFID
2. **Flipper Zero**:
   - Multi-tool for pentesting (RFID, RF, IR, GPIO, Bluetooth, Wi-Fi)
3. **ESP RFID Tool**:
   - Data logger for Wiegand interface (used in access control systems)
   - Logs credentials from card readers, PIN pads, biometric systems

---

What is ZigBee?

- Mesh networking standard based on IEEE 802.15.4
- Designed for **low power**, **low data rate** (20–250 Kb/s) applications
- Very long battery life
- High reliability via mesh connectivity
- AES-128 encryption available → **Very secure**
- Self-configuring, supports ad hoc networks
- Easy to install and configure

ZigBee / IEEE 802.15.4 Market Features

- Transmits at 10–100 milliwatts (much less than Bluetooth)
- Inexpensive (as low as $3)
- Supports large networks (up to 65,000 nodes)
- Low message throughput
- Minimal QoS (Quality of Service) guarantees
- Protocol flexibility suits many applications

IEEE 802.15.4 Basics

- Lightweight packet data protocol
- Uses CSMA/CA with optional **time slotting**
- Message acknowledgment and optional beacon structure
- Ideal for:
    - Long battery life
    - Low-latency needs (e.g., controllers, sensors)
    - Remote monitoring, portable electronics
- Configured for maximum battery efficiency—can match battery shelf life

---

Device Types in IEEE 802.15.4

1. Full Function Device (FFD)

- Operates in any topology
- Can be:
  - Device
  - Coordinator
  - PAN Coordinator
- Can talk to any device

2. Reduced Function Device (RFD)

- Star topology only
- Cannot be a coordinator
- Only talks to the network coordinator
- Simple implementation

Topologies

Star Topology

- One **Network Coordinator** (FFD)
- Multiple **RFDs** communicate with the coordinator (master/slave)

Peer-to-Peer Topology

- Point-to-point communication
- Only **FFDs** participate

Tree Topology

- Hierarchical structure
- FFDs route messages
- RFDs at the leaf nodes

Combined Topology (Clustered Stars)

- Clustered nodes between areas (e.g., hotel rooms)
- Each room has its own star network
- Combines multiple topologies

Device Addressing

- Each PAN has a unique **PAN ID**
- Each device has a unique **64-bit extended address**
- PAN Coordinator assigns a **16-bit short address** when a device joins
- Addressing varies by topology:
    - **Star**: Network (64-bit) + Device (16-bit)
    - **Peer-to-peer**: Source/Destination (64-bit)
    - **Cluster tree**: Cluster + Device identifier (less clearly defined)

Channel Access Mechanisms

- **Non-beacon-enabled**: Uses **unslotted CSMA/CA**
- **Beacon-enabled**: Uses **slotted CSMA/CA**
- Devices align their **backoff period** with the **superframe slot boundaries** from the PAN coordinator
- MAC sublayer ensures PHY transmits only on backoff boundaries

---

CSMA/CA Algorithm Variables

- **NB** (Number of Backoffs): Retries for backoff
- **BE** (Backoff Exponent): Random delay before channel check
- **CW** (Contention Window): Slots to wait with clear channel before sending
    - Initialized to 2
    - Resets to 2 if the channel is busy
    - Must detect **two clear CCAs** before transmission

Data Transfer Models

Device to Coordinator

- **Beacon-enabled**:
  - Device finds beacon, syncs to superframe, sends data using **slotted CSMA/CA**
- **Non-beacon-enabled**:
  - Device sends data directly using **unslotted CSMA/CA**

Coordinator to Device

- **Beacon-enabled**:
  - Beacon signals **pending data**
  - Device listens periodically and sends a MAC request via slotted CSMA/CA
- **Non-beacon-enabled**:
  - Device sends MAC request using unslotted CSMA/CA
  - If data is pending → coordinator sends data frame
  - If not → sends a frame with **zero-length payload**

Superframe in ZigBee

- **Superframe**: The repeating structure defining how time is divided for device communication.
- **Transmitted by**: The Network Coordinator.
- **Parts**:
  - **Inactive**: All devices sleep to conserve power.
  - **Active**: Divided into 16 slots (called **MACRO slots**):
    - **CAP (Contention Access Period)**: Any node can access using CSMA/CA.
    - **CFP (Contention Free Period)**: Reserved for devices needing **guaranteed bandwidth** (Guaranteed Time Slots - GTS).

Beacon: Sent by the coordinator at regular intervals. Contains network info, superframe structure, and pending message notifications.
GTS Duration: 15ms * 2^n ($0 \le n \le 14$), assigned to a device for either transmit (t-GTS) or receive (r-GTS).

Superframe Structure Parameters

- **BO (Beacon Order)**: Defines the **length of the beacon interval**.
- **SO (Superframe Order)**: Defines the **length of the active period**.
- **In CFP**:
  - GTS may span multiple slots, all for one device.
- **In CAP**:
  - No fixed slot structure.
  - Divided into 20-symbol-long **contention slots** for CSMA/CA backoff.

Security Models in ZigBee

Centralized Security Model (Secure but Complex)

- Managed by a **Trust Center (usually the coordinator)**.
- Trust Center duties:
  - Authenticate and configure devices.
  - Generate and rotate **network keys**.
  - Assign unique **link keys** for secure communication with each device.
  - Maintain overall network security.

Distributed Security Model (Simpler, Less Secure)

- Only routers and end devices; **no central Trust Center**.
- Routers enroll other routers/devices.
- All devices use:
  - **Same network key** for encryption.
  - **Pre-configured link key** to protect key exchange.

Data Encryption in ZigBee

- Based on **IEEE 802.15.4 security**.
- Uses **AES-128** encryption (16 bytes).
- Appends **AES-based Message Authentication Code (MAC)** to messages:
  - Ensures integrity of MAC header + payload.
  - MAC can be 32, 64, or 128 bits (always generated using AES-128).
- **Auxiliary Security Header** is used when the security flag is enabled.

ZigBee Security Keys (128-bit symmetric)

1. Network Key

- Used in **broadcast** communication.
- Generated by Trust Center and distributed via:
  - **Key transport** or
  - **Pre-installation**
- Types:
  - **Standard** (sent in plaintext)
  - **High-security** (encrypted)

2. Link Key

- Used for **unicast** (device-to-device) communication.
- Obtained via:
  - **Pre-installation**
  - **Key establishment** (using a master key)
  - **Key transport**
- **Trust Center Link Key** is pre-configured out-of-band (e.g., QR code).
- Between devices: Trust Center generates and sends it encrypted with the network key.

3. Master Key

- Long-term security between two nodes.
- Used only during **SKKE** (Symmetric Key Key Establishment) to protect link key exchange.
- Shared via:
  - **Key transport**
  - **Pre-installation**
  - **User-entered** methods (e.g., PIN/password)

Advanced Key Management

Pre-installation

- Manufacturer embeds keys; user selects via hardware interface (e.g., jumpers).

Key Establishment

- Local generation of keys using master key.
- Derives other service-specific keys using one-way functions.

Key Transport

- Device requests keys from Trust Center.
- Used for any of the three key types.
- **Key-load key** protects master key transport.
- Supports **CBKE (Certificate-Based Key Establishment)** in centralized model.

Install Code & Trust Center Link Key

- Each ZigBee device may have a **unique install code** (128-bit + 16-bit CRC).
- Used to generate the **Trust Center Link Key** using the **MMO hash function**.
- Trust Center verifies the install code before allowing the device to join.

ZigBee Vulnerabilities

1. Implementation Vulnerabilities

- **Insecure key storage**: Keys can be reverse-engineered from firmware.
- **Unencrypted over-the-air key transport**: Keys intercepted during join process.
- **Energy depletion attacks**:
  - **Invalid security headers** force device to process junk frames.
  - **Polling rate abuse** increases power consumption.

2. Protocol Vulnerabilities

- **Link Layer Jamming**: Flooding the MAC layer with frames to cause DoS.
- **Default Link Key usage**:
  - Many devices use the same default key (e.g., ZigBeeAlliance09).
  - Attacker can join network using this known key.
- **Unencrypted Link Key delivery**:
  - Trust Center sends keys in plaintext to new devices → easily sniffed.
- **Link Key Reuse**:
  - Rejoining with reused keys lets attacker spoof device identity.

Acknowledgment (ACK) Attacks

- **ACK Spoofing**: Legitimate receiver's frame is blocked; attacker sends fake ACK with correct sequence.
- **ACK Dropping**: Attacker jams ACKs, forcing retransmission → bandwidth waste + battery drain.

Bluetooth Overview

- **Definition:** Bluetooth is a short-range wireless communication standard.
- **Range:** Typically up to 10 meters (can extend up to 100 meters in modern versions).
- **Origin of Name:** Named after King Harald “Bluetooth” Blatand, who unified Denmark and Norway.
- **History:**
  - 1994: Developed by Ericsson for linking mobile phones to accessories.
  - 1998: Bluetooth SIG (Special Interest Group) formed by 5 companies.
  - 1999: First Bluetooth specification released.

Bluetooth Versions

- **1.x (1999):** Basic wireless connectivity, low range, low speed.
- **2.x (2004):** Introduced Enhanced Data Rate (EDR), speed up to 3 Mbps.
- **3.x (2009):** High-Speed Bluetooth (HSB), speed up to 24 Mbps.
- **4.x (2010):** Introduced Bluetooth Low Energy (BLE) for low power devices.
- **5.x (2016):** Extended range, speed, BLE Mesh, LE Audio introduced in 5.2.
- **6.0:** Latest version (referenced but not officially released as of mid-2025).

Frequency Band

- **Uses:** 2.4 GHz ISM band (shared with Wi-Fi and others).
- **Technique:** Frequency Hopping Spread Spectrum (FHSS).
- **Channels:** 40 channels (1 MHz bandwidth each), hops at 1600 times/sec.

Bluetooth Architecture

- **Layered Model:**
  - Application Layer: Services & Profiles
  - GATT / ATT: Data abstraction and structuring
  - GAP: Device discovery & connection
  - L2CAP: Logical link management
  - SM: Security Manager
  - LL & PHY: Link layer and physical radio

Bluetooth Profiles

- **A2DP:** Audio streaming (e.g., to headphones).
- **HFP:** Hands-free calling.
- **HID:** Keyboards, mice, controllers.
- **OPP:** File transfers.
- **SPP:** Serial communication.
- **GATT:** BLE device communication.

Key Bluetooth Operations

1. **Pairing:** Establish secure link.
2. **Discovery:** Devices find each other.
3. **Connection:** Establish data channel.
4. **Transmission:** Exchange of data.
5. **Disconnection:** Ends session.

Transmission & Networking

- **Time Division:**
  - Slot-based (625 µs)
  - Master: Transmits in even slots
  - Slave: Transmits in odd slots
- **Networking:**
  - **Piconet:** 1 master + up to 7 slaves.
  - **Scatternet:** Multiple piconets interconnected.
  - Devices in a piconet hop together based on master's ID and clock.

Bluetooth vs Other Wireless Technologies

Feature	Bluetooth	RFID	Wi-Fi	NFC
Range	Up to 100m	Few cm to few m	Up to 100m	Up to 10cm
Data Rate	Up to 24 Mbps	Up to 424 Kbps	Several Gbps	Up to 424 Kbps
Power	Battery needed	Reader-powered	High	Reader-powered
Security	Moderate	Moderate	High	Lower than Bluetooth
Cost	More expensive	Cheaper	More expensive	Cheaper
Use Cases	Audio, IoT	Tracking, inventory	Streaming, internet	Mobile payments

---

Bluetooth Security

- **Pairing:** Establishes secure communication using key exchange.
- **Encryption:** AES (typically 128-bit) used to protect data.
- **Authentication & Authorization:** Controls access to services/devices.
- **SSP (Secure Simple Pairing):** Prevents MITM attacks via OOB pairing.

Security Modes & Levels

- **Modes:**
    - Mode 1: No data signing
    - Mode 2: With data signing
    - Mixed: Supports both
- **Levels:**
    - Level 1: No security (unpaired)
    - Level 2: AES-CMAC with no pairing
    - Level 3: Requires pairing
    - Level 4: Uses ECDHE (P-256)

Pairing Phases

1. **Phase 1 (Capabilities Exchange):**
    - Uses ATT values to decide the pairing method.
2. **Phase 2 (Key Generation):**
    - Generates STK or LTK depending on mode.
3. **Phase 3 (Key Distribution):**
    - Distributes LTK, IRK, and CSRK for secure communication.

Pairing Methods

- **Numeric Comparison:** Both devices display same code.
- **Just Works:** No display, no MITM protection.
- **Passkey Entry:** One device displays, other inputs.
- **OOB (Out-of-Band):** External method (e.g., NFC, camera scan).

Bluetooth Vulnerabilities

Vulnerability	Description
Bluejacking	Sends unsolicited messages (annoyance/phishing)
Bluesnarfing	Steals data like messages, photos
Bluebugging	Backdoor access, full control
BlueFrag (2020)	Android 8-9 vulnerability, remote code execution
Bluewave (2020)	macOS bugs, device takeover without interaction
BleedingTooth	Linux kernel zero-click exploit
Bluesmacking	DoS attack via oversized packets
Car Whispering	Eavesdropping on car audio/Bluetooth
Privacy Leaks	Location tracking via persistent Bluetooth signals

---

Bluetooth Hacking Tools

- **Bluelog:** Discover & log devices nearby.
- **Bluemaho:** GUI security testing suite.
- **Blueranger:** Locate devices via ping.
- **Btscanner:** GUI scanner.
- **Redfang:** Find hidden devices.
- **Spooftooph:** Bluetooth spoofing tool.
- **Spooftooth:** Available in Kali Linux (2020+).