🔐 Why and How IoT is *Not* Secure at All

😬 The Harsh Truth: Most IoT Devices Are Born Vulnerable

IoT is like building a city full of doors… but forgetting to add strong locks on most of them.

📉 Why IoT Isn't Secure:

1. **Cheap Devices = No Built-in Security**
    - Manufacturers focus on cost, not protection.
    - Many devices ship without proper encryption or update features.
2. **No Regular Updates**
    - Unlike your phone or laptop, many IoT devices never get security patches.
    - Once hacked, always hacked.
3. **Default Passwords**
    - Many devices come with default usernames like `admin/admin`.
    - Users don't change them. Easy entry for hackers.
4. **Always Connected**
    - Devices stay connected to the internet 24/7.
    - That's like leaving your house door open all day.
5. **Lack of Standards**
    - No common security rules followed across companies.
    - One device might be secure, another a total disaster.

🧠 Real-World Scary Examples:

- **Baby Monitors Hacked**: Parents heard strangers talking to their babies.
- **Smart TVs Spied On**: Some TV models were secretly recording voices.
- **Mirai Botnet Attack**: Thousands of hacked IoT devices were used to crash major websites in 2016.

🔐 IoT Security & Privacy Measures

Here's how we **fight back** and secure the IoT world:

🔒 1. Encryption

- Converts data into code so that only the correct device/person can read it.
- Just like speaking in a secret language.

🧠 **Example**: Sensor data from your smartwatch to your phone is encrypted, so even if someone intercepts it — they can't read it.

🛡️ 2. Secure Communication

- Use secure protocols like:
  - **HTTPS** (not HTTP)
  - **MQTTS** (secure MQTT)
  - **DTLS** (secure version of UDP)

- Also uses **certificates**, **tokens**, and **authentication keys** to verify who's allowed in.

🔗 3. Blockchain for IoT Security

Now this is interesting. Let's dive deep.

🔗 Blockchain in IoT – A Digital Trust Machine

Blockchain isn't just for cryptocurrency. It's becoming a **security shield** for IoT.

🤔 What is Blockchain?

- A **distributed, unchangeable ledger**.
- Every transaction is saved as a "block".
- Once added, it can't be changed or faked.

🛡️ How Blockchain Helps IoT:

1. **Device Identity & Trust**
    - Every IoT device gets a secure, verifiable ID on the blockchain.
    - Prevents fake devices from entering the network.
2. **Data Integrity**
    - Once data is recorded, no one can modify it.
    - Great for medical, industrial, and financial IoT data.

3. **Decentralization**
    ○ No single point of failure.
    ○ Even if one server is hacked, the system stays safe.
4. **Smart Contracts**
    ○ Self-executing rules like:

      “If sensor detects temperature > 50°C, shut down motor and alert supervisor.”

    ○ Executes securely without human help.

💡 Real Example:

A **smart energy grid** uses blockchain to:

- Track power usage
- Securely bill customers
- Prevent tampering of energy meters

All done transparently and tamper-proof.

☁️ Data Storage & Processing in IoT: Cloud vs Edge

IoT devices don't just sense data… they must **store it, process it, and make decisions**. That happens in two main places:

☁️ Cloud Computing

Think: All data goes to a big server somewhere far away.

✅ **Pros:**

- Huge storage capacity
- High processing power
- Easy to run AI models
- Central management of devices

❌ **Cons:**

- Delay (Latency): Takes time to send and receive data
- Needs constant internet
- Privacy risks (your data goes to external servers)

🌐 Edge Computing

Think: Processing is done *locally*, close to the sensor/device.

✅ **Pros:**

- Super fast (real-time response)

- Works without internet
- More privacy (data doesn’t leave the device)

❌ **Cons:**

- Limited storage
- Less processing power
- Harder to update software

🧠 When to Use What?

Situation	Use Cloud or Edge?	Why?
Voice assistant in your phone	Edge	Needs fast response
Analyzing traffic in a smart city	Cloud	Huge data, advanced AI needed
Smartwatch heart monitoring	Edge + Cloud	Immediate alerts + long-term storage
Industrial machine control	Edge	Safety-critical, low latency

📗 IoT Device Power Management

(Optimizing Battery Life & Energy-Efficient Computing)

🪫 Why Power Management is So Important in IoT?

IoT devices are like tiny **scouts** placed all over the world — in farms, on bridges, inside machines, or even in your shoes (hello, smart insoles 👟).
Many of these devices are:

- **Battery-powered**
- **Placed in remote/hard-to-reach places**
- Expected to work for **months or years** without human touch

⚠️ So if they run out of battery too fast… it's like having a phone without a charger in the jungle 🌳 — **useless**.

🎯 Goal: Maximize Battery Life

We want devices to:

- Run **longer** (months/years)
- Work **smarter** (only when needed)
- Use **less energy** (optimize computing, communication, sensors)

🔋 Power-Hungry Parts of an IoT Device

Component	Why It Uses Power
Sensors	Continuously collecting data (e.g., temperature, motion)
Microcontroller (MCU)	Processes data, runs code
Communication module	Sends data via Wi-Fi, LoRa, BLE — this eats up **a lot of power**
Display (if any)	Displays use extra power (like in smartwatches)

🧠 Smart Ways to Save Power in IoT Devices

1. Sleep Modes 💤

- Microcontrollers (like ESP32, STM32) have **sleep/deep sleep** modes.
- The device goes to "sleep" when not working and **wakes up only when needed**.
- Example: A temperature sensor might sleep 59 seconds and wake up 1 second every minute.

🧪 Like a lazy genius who only wakes up to do something important, then naps again.

---

2. Use Low-Power Communication Protocols

- Instead of Wi-Fi (high power), use:
  - **LoRa**: For long range, super low power
  - **BLE (Bluetooth Low Energy)**: For wearables

- **NB-IoT**: Narrowband, cellular-based, but energy-efficient

💡 Communication is often **the most power-hungry task** — so optimizing it is 🔑.

---

3. Edge Computing (Think before you speak)

- Instead of sending raw data to the cloud every time, process it **locally**.
- Only send **important results**.
- Saves **network power + cloud costs**.

🧪 Like filtering your thoughts before talking — more energy-efficient and less annoying 😄

---

4. Efficient Code & Scheduling

- Optimize the code on the microcontroller.
- Don't run unnecessary loops or tasks.
- Use **interrupts** instead of constantly checking things (polling).

Example: Instead of checking "is button pressed?" every second, use an **interrupt** that triggers only *when the button is pressed*.

---

5. Hardware Selection Matters

- Choose microcontrollers designed for low power:
  - **ESP32-S2**, **STM32L0 series**, **TI MSP430**, etc.
- Use **energy-efficient sensors** (with sleep modes)

---

6. Duty Cycling (Work in Bursts)

- Don't keep the system ON all the time.
- Collect, process, and transmit data in **short bursts** — then go back to sleep.

🔁 Example:

Time	Task
00:00	Wake up
00:01	Read sensor
00:02	Send data
00:03	Sleep
01:00	Wake up again!

🔋 Energy-Saving Modes in Popular Microcontrollers

MCU	Sleep Modes	Approx Power (Deep Sleep)
ESP32	Light, Deep, Hibernation	~10 μA
STM32L0	Sleep, Stop, Standby	~1 μA
Arduino Uno	Idle, Power-down	~0.2 mA

🧪 Even 0.1 mA = battery saved = device lives longer 🎉

🛠️ Real-Life IoT Example: Smart Agriculture Node (Soil Sensor)

- **Sensor**: Measures soil moisture every 1 hour

- **MCU**: ESP32
- **Comms**: LoRa (low power, long range)
- **Optimization**:
    - Sleep for 59 min 55 sec
    - Wake, read, send, sleep again
    - Use solar panel for backup charging

🧪 Result: Can work **years** with a small battery 🔋

📚 Summary – Power Management Tips

✅ Strategy	💡 Why It Helps
Sleep/Deep Sleep	Saves battery when idle
Use BLE, LoRa, NB-IoT	Less energy vs Wi-Fi
Local data processing	Avoids unnecessary comms
Use interrupts	Only react when needed
Smart scheduling	Duty cycling = longer life
Low-power hardware	Efficient by design

🎓 Final Thought:

> “In IoT, it’s not how fast your device works — it’s how long it survives the wild.”
> — Prof. Mortius, Dept. of Practical Futurism 😄