Threat Model Analysis - IoT

Group Members:
Satya Kamala Immidisetty - si2502
Alec Pippas - awp251
Xibo He - xh2774
Suprateek Chatterjee - sc10344
Nico Flores

Introduction

The purpose of this analysis is to evaluate the security threats associated with Internet of Things (IoT) environments, where interconnected devices often operate with limited computational resources, in varied network conditions, and with high exposure to physical and remote attack vectors. The primary objective of this threat model is to uncover the most relevant risks and propose cryptographic solutions that can mitigate these vulnerabilities, given IoT's unique constraints.

Scope

This analysis focuses on identifying IoT-specific threats, including data interception, device spoofing, and network-level attacks that compromise the confidentiality, integrity, and availability of sensitive information and device operations. Through this analysis, we aim to build a clearer understanding of the cryptographic measures required to secure IoT systems, ultimately improving device resilience and data security across IoT ecosystems.

Threat Identification

Key Components in the IOT

The Internet of Things (IoT) refers to a network of interconnected physical devices, or "things," that are embedded with sensors, software, and other technologies to exchange data over the internet. These devices, ranging from smart home appliances and wearable gadgets to industrial machines and healthcare devices, can collect and share data, enabling automation and real-time decision-making across diverse environments. IoT acts as a bridge between the digital and physical worlds, enhancing efficiency, insights, and user experiences in various sectors like healthcare, agriculture, manufacturing, and smart cities.

1. **IoT Device:** Physical devices embedded with sensors, actuators, and communication capabilities to collect data from their environment (e.g. temperature, motion) or perform actions (e.g. controlling a thermostat).

2. **Connectivity and Communication:** Protocols and networks that enable data transmission between devices and other systems. Common protocols include Wi-Fi, Bluetooth, MQTT, Zigbee, and 5G.
3. **IoT Gateways:** Devices that bridge IoT sensors and cloud systems, managing data flow, security functions, and sometimes preprocessing data locally before it is sent to the cloud.
4. **Cloud and Edge Computing:** Infrastructure that processes, analyzes, and stores data collected by IoT devices. Cloud computing offers centralized, advanced data handling, while edge computing allows processing close to the source, reducing latency and enabling faster decision making.
5. **Data Storage and Analytics:** Databases and analytic systems that store and analyze IoT data to gain insights, detect trends, or trigger actions. This may include machine learning models to enhance predictive capabilities.
6. **User Interface (UI):** Applications, web dashboards, or mobile apps that allow users to view and control IoT devices, monitor data, and receive alerts in real time.
7. **Security Components:** Mechanisms such as encryption, authentication, and secure firmware updates to ensure the confidentiality, integrity, and availability of IoT data and devices.

IoT Workflow

1. **Data Generation and Collection:** IoT devices with sensors gather data from their environment, (e.g. temperature, location, humidity, motion), or take actions based on received instructions, such as adjusting brightness or unlocking doors.
2. **Data Transmission:** Collected data is transmitted over a network to an IoT gateway or directly to a cloud server, typically through communication protocols like MQTT, HTTP, or CoAP.
3. **Data Processing and Analysis:** Data is either processed at the edge (near the devices) or sent to the cloud for in-depth analysis. For example, real-time data might be analyzed at the edge by a decision making algorithm and trigger immediate actions within automated systems (e.g. shutting down factory machinery if IoT sensors detect unsafe temperature or pressure levels). Meanwhile, historical data analysis within the cloud could reveal patterns and trends that are used for short-term insight (e.g. smart electrical grid management) or long-term strategic planning (e.g. retail inventory management) by either human or algorithmic decision-makers.
4. **Data Storage:** Data is securely stored in databases, often in the cloud, for ongoing analysis, historical records, or compliance purposes. Storage can be centralized (cloud) or distributed (edge).
5. **Action or Feedback Loop:** Based on analysis, insights are generated and may trigger actions, such as sending alerts to users, initiating maintenance, or adjusting device settings automatically.
6. **Human User Interaction:** The processed data or actions are presented to the human end-users via a UI, allowing them to monitor, control, and gain insights into their IoT devices and systems.

This IoT workflow enables seamless connectivity, real-time monitoring, and enhanced automation, providing valuable information and control across multiple industries and applications.

Common Threats and Vulnerabilities

1. **Spoofing**
    - **Threat:** Attackers may impersonate legitimate devices or users to gain unauthorized access to other IoT devices or the IoT network at large.
    - **Vulnerabilities:**
        i. Weak or no device authentication protocols.
        ii. Default or hardcoded passwords.
        iii. Insecure registration processes for new devices.
    - **Example Attack Vectors:**
        i. An attacker impersonates a legitimate sensor to inject false data.
        ii. Malicious actors spoof a gateway device to intercept communication.
2. **Tampering**
    - **Threat:** Unauthorized modification of data or device firmware.
    - **Vulnerabilities:**
        i. Physical access to devices lacking tamper-resistant hardware.
        ii. Inadequate firmware update mechanisms.
        iii. Lack of cryptographic integrity checks for data.
    - **Example Attack Vectors:**
        i. An attacker alters the firmware of a device to execute malicious functions.
        ii. Manipulating sensor readings (e.g., temperature or pressure data)..
3. **Repudiation**
    - **Threat:** Users or devices deny performing an action, such as sending a command or accessing data.
    - **Vulnerabilities:**
        i. Insufficient logging or auditing mechanisms.
        ii. Weak or absent user and device identity tracking.
    - **Example Attack Vectors:**
        i. An attacking user denies sending a command to an IoT device by tampering or removing access and activity logs
        ii. A compromised device denies transmitting specific sensor readings by changing its ID to that of another IoT device within the system
4. **Information Disclosure**
    - **Threat:** Unauthorized access to sensitive, confidential data.
    - **Vulnerabilities:**
        i. Unencrypted data transmission or storage.
        ii. Inadequate access controls for cloud-stored data.
        iii. Misconfigured APIs exposing sensitive information.
    - **Example Attack Vectors:**
        i. Eavesdropping on unencrypted IoT communication channels.
        ii. Leaking personal data from a smart home hub.
5. **Denial of Service**
    - **Threat:** Disrupting the availability of services by overwhelming the network or devices with excessive requests.

- **Vulnerabilities:**
    i. Insufficient capacity to handle traffic spikes or malicious floods.
    ii. Lack of rate limiting or throttling mechanisms.
    iii. Weak device resilience to network attacks.
- **Example Attack Vectors:**
    i. Botnet attacks like Mirai leveraging IoT devices for massive DoS attacks.
    ii. Overloading a device with frequent, malicious requests.

6. **Elevation of Privilege**
    - **Threat:** Gaining higher access rights than authorized.
    - **Vulnerabilities:**
        i. Insecure firmware allowing privilege escalation exploits.
        ii. Poorly designed access control mechanisms.
        iii. Lack of separation of duties for device and network administration.
    - **Example Attack Vectors:**
        i. A hacker escalates from a regular user account to an administrator.
        ii.
        iii. Exploiting a vulnerable API to control multiple devices.

Risk Assessment and Management

1. **Risk Assessment Matrix**

Threat	**Impact**	**Probability**	**Impact Severity**	**Risk Management Strategies**
Device/Identity Spoofing	Comprised access to devices could potentially lead to false data that disrupts operations or leads to poor decision making	High	High	ECDSA for device authentication and HMAC for data integrity
Tampering	Altered device firmware could lead to device failures and require halting operations. False or deleted data could cover-up safety issues.	Medium	High	Secure boot and firmware signing, HMAC for integrity checks on transmitted data
Repudiation	Difficulty in attributing critical actions could delay or prevent incident responses, or it could violate auditing regulations	Low	Medium	ECDSA for non-repudiable, robust logging mechanisms
Information Disclosure	Exposure of user or operational data could empower inside	High	High	AES encryption for data confidentiality,

	threat actors, jeopardize the privacy of customers/employees, or ruin competitive advantage			ECDH for secure key exchange
Denial of Service	Overloaded network bandwidth or device resources (e.g. memory), could disrupt operations or, for customer IoT devices, lead to decreased UX and in turn impact revenue	High	High	Rate limiting IoT device requests, traffic filtering, and device hardening (e.g. disabling unused ports)
Elevation of Privilege	Unauthorized access to admin functions, allows attackers to disable security features of device and access confidential data	Medium	High	Secure firmware with access control mechanisms, principle of least privileges enforcement

Cryptographic Solutions

To mitigate the identified threats and vulnerabilities in IoT systems, we propose the following cryptographic solutions tailored to the specific needs and constraints of IoT environments:

1. **Symmetric Encryption with AES (Advanced Encryption Standard)**
    a. Application: Data Confidentiality
    b. Purpose: Prevents Information Disclosure by encrypting data transmitted between IoT devices and servers.
    c. Implementation: Use AES with a secure key to encrypt sensitive data before transmission. Keys are securely managed and distributed only to authorized entities.
    d. Benefits: AES is efficient and widely supported, making it suitable for resource-constrained IoT devices.

2. **Hash-Based Message Authentication Code (HMAC) with SHA-256**
    a. Application: Data Integrity Verification
    b. Purpose: Prevents Tampering by ensuring that data has not been altered during transmission.
    c. Implementation: Generate an HMAC for each message using a shared secret key. The receiver verifies the HMAC to confirm data integrity.
    d. Benefits: Provides a lightweight and effective method for message authentication compatible with IoT devices.

3. **Elliptic Curve Digital Signature Algorithm (ECDSA)**
    a. Application: Device Authentication and Non-Repudiation
    b. Purpose: Prevents Spoofing and Repudiation by verifying the authenticity of devices and messages.

c. Implementation: Each device holds a private key to sign messages, while public keys are distributed for signature verification.
d. Benefits: Offers strong security with shorter key lengths, suitable for devices with limited processing power.

4. **Elliptic Curve Diffie-Hellman (ECDH) Key Exchange**
    a. Application: Secure Key Exchange
    b. Purpose: Establishes encrypted communication channels to prevent Information Disclosure.
    c. Implementation: Devices use ECDH to securely derive shared secret keys over unsecured networks.
    d. Benefits: Efficiently establishes secure communication with minimal computational overhead.

5. **Secure Boot and Firmware Signing**
    a. Application: Device Integrity and Secure Updates
    b. Purpose: Prevents Tampering and Elevation of Privilege by ensuring only authenticated firmware runs on devices.
    c. Implementation: Firmware images are digitally signed using cryptographic signatures. Devices verify signatures before installation.
    d. Benefits: Protects against malicious firmware, ensuring devices operate as intended.

Project Proposal

Focus Area for **Further Exploration:** Implementing Post-Quantum Digital Signatures for IoT Device Authentication Using Open Quantum Safe (OQS)

Objective

1. Enhance IoT device authentication and message integrity by implementing post-quantum digital signatures using the Open Quantum Safe (OQS) library.

Proposed Approach

1. Algorithm Selection:
    a. Choose a suitable post-quantum digital signature algorithm from OQS (e.g., CRYSTALS-Dilithium) that balances security and performance for IoT devices.
2. Prototype Development:
    a. Utilize OQS's Python bindings to develop a prototype for device authentication.
    b. Simulate IoT devices signing messages and verifying signatures using the selected algorithm.
3. Performance Evaluation:
    a. Assess computational overhead and resource usage.
    b. Determine feasibility for deployment on resource-constrained IoT hardware.

4. Security Validation:
    a. Test resistance against spoofing and tampering.
    b. Ensure quantum-resistant security properties are met.

Expected Outcome
Demonstrate secure IoT device authentication with enhanced protection against current and future quantum threats, providing insights into practical implementation of post-quantum cryptography in IoT environments.

Conclusion

This threat model provides a foundational framework to identify and address security risks within the Internet of Things (IoT) domain using cryptographic solutions. By focusing on implementing post-quantum digital signatures with Open Quantum Safe (OQS), we aim to achieve secure, future-proof authentication and data integrity for IoT devices. This approach ensures resilience against current and emerging threats, supporting the safe and scalable deployment of IoT systems across diverse applications.