Faculty of Engineering
Department of Electrical and Electronics Engineering

EE 450 FINAL SURVEY

Spring 2025

Security and Privacy Issues in IoT:

A Layered Approach and the Role of Secure Firmware Updates

Submitted by

Hasan Alp Doyduk
S025015

İsmail Akbaş
S024094

ABSTRACT

As the Internet of Things (IoT) continues to expand across critical domains, the security and privacy of these systems have become increasingly urgent concerns. IoT devices are often deployed in distributed, unsupervised, and resource-limited environments, making them vulnerable to attacks that traditional network security solutions are not designed to withstand.

This project analyzes the security architecture of IoT systems through a layered approach, focusing on the perception, network, and application layers. Special attention is given to firmware-level risks, as firmware acts as the foundational software of most IoT devices and is a common target for exploitation. Threats such as authentication bypass, code injection, memory-based attacks, and insecure over-the-air (OTA) updates are examined as key vulnerabilities.

To address these challenges, the report explores modern countermeasures including firmware signature verification, secure boot mechanisms, lightweight encryption, federated learning, and fine-grained access control models such as RBAC, ABAC, and UCON. The use of blockchain for decentralized access verification and firmware update auditing is also considered.

The findings emphasize that securing the firmware update pipeline and applying multi-layered, scalable protection strategies are essential for ensuring IoT system integrity. A combination of hardware-based trust, software-level enforcement, and privacy-preserving techniques is necessary to protect these systems against evolving threats.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ABAC	Attribute-Based Access Control
AES	Advanced Encryption Standard
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
IoT	Internet of Things
MCU	Microcontroller Unit
MITM	Man-in-the-Middle
OTA	Over the Air
UCON	Usage Control
RBAC	Role-Based Access Control
RFID	Radio Frequency Identification
TLS	Transport Layer Security
TPM	Trusted Platform Module

1 INTRODUCTION

The IoT has become a fundamental component of modern digital infrastructure, enabling real-time monitoring, control, and data-driven automation across sectors such as healthcare, agriculture, manufacturing, and smart cities. As of 2024, the global number of IoT devices has exceeded 17 billion and continues to grow rapidly. Although this growth introduces unprecedented efficiency and intelligence into distributed systems, it also creates a wide and increasingly complex attack surface that challenges conventional notions of cybersecurity [1].

Unlike traditional computing systems, IoT environments are characterized by decentralized architectures, low-cost and low-power devices, and heterogeneous communication protocols. Devices are often deployed in unsupervised and physically exposed environments, operate under strict resource constraints, and lack the capacity to support robust cryptographic or intrusion prevention mechanisms. These limitations, combined with the absence of standardized security frameworks across vendors, make IoT ecosystems uniquely vulnerable to a broad range of threats [2].

IoT systems are typically structured in three architectural layers—perception, network, and application—each of which faces its own category of risks. Threats range from physical tampering and signal spoofing at the device level to network-based man-in-the-middle attacks and protocol abuse, and extend to application-layer risks such as unauthorized access, malicious trigger commands, and large-scale privacy violations. Traditional security solutions often fail to address these risks in a holistic and resource-aware manner.

A critical yet often underprotected element within this architecture is firmware—the core software that defines device behavior and connectivity. Firmware vulnerabilities such as hardcoded credentials, memory corruption bugs, and insecure update mechanisms have been widely exploited in real-world attacks. The growing use of OTA firmware updates, while essential for maintainability, also introduces new risks if updates are not properly authenticated, encrypted, or version-controlled. Once compromised, a device's firmware can be used to propagate malware, manipulate sensor readings, or even render the hardware unusable [3].

This study addresses the growing need for secure and scalable protection strategies in IoT by analyzing security threats through a layered architectural lens, with a particular focus on firmware integrity and the OTA update pipeline. Drawing from recent academic literature, the report surveys existing vulnerabilities and evaluates contemporary solutions such as federated learning, access control models, blockchain-based update verification, and lightweight cryptographic mechanisms. The overarching goal is to highlight the importance of designing integrated, resource-efficient security frameworks that are capable of defending IoT systems at both the software and system levels.

2 RESEARCH SCOPE AND APPROACH

This project is structured as a layered literature-based analysis of security and privacy challenges in IoT systems, with an emphasis on firmware vulnerabilities and the secure management of OTA updates. Rather than focusing on simulation or prototype development, the study relies on existing academic research to identify threat models, evaluate mitigation strategies, and highlight integration challenges across architectural layers. The core of the approach is based on four interconnected phases that define the scope and structure of the report:

2.1 Literature Selection and Review

The study began with the identification of two primary academic survey sources: one offering a broad review of IoT security and privacy issues across multiple system layers, and the other focusing specifically on firmware-level threats and OTA update security. These sources were selected based on their recency, citation impact, and alignment with the project's layered framework. Additional supporting literature on cryptographic protocols, access control models, federated learning, and blockchain integration was also reviewed to ensure that both foundational concepts and emerging technologies were covered.

2.2 Layer-Oriented Structuring of Threats and Defenses

IoT systems are classically structured into three functional layers: perception, network, and application. Each of these layers introduces distinct risks, influenced by factors such as hardware exposure, protocol limitations, and user-facing vulnerabilities. In this phase, each layer was examined independently, with attention given to the most common attack surfaces and technical constraints. The layered structure also enabled the identification of firmware as a cross-cutting component that affects security at all levels.

2.3 Mapping of Protection Techniques

Following the threat classification, the report evaluates a wide range of existing protection strategies. These include cryptographic solutions such as DTLS and ECC for network-layer encryption, firmware validation through signature-based integrity checks, and access control frameworks like RBAC, ABAC, and UCON at the application layer. Privacy-preserving models such as federated learning and searchable encryption were also studied as emerging tools for maintaining data confidentiality in distributed IoT environments. In addition, the report considers the practical feasibility of these techniques in resource-constrained environments, particularly regarding processing overhead, memory requirements, and protocol compatibility. Where available, real-world deployment examples and experimental evaluations were incorporated to assess the maturity and effectiveness of each approach.

2.4 Cross-Layer Integration and Analysis

The final dimension of the research approach is focused on cross-layer interactions. Many IoT security mechanisms operate across multiple layers—for example, an OTA firmware update relies on secure transport at the network layer, device integrity at the perception layer, and authorized execution from the application layer. By evaluating the dependencies between these layers, the study aims to identify both synergy and friction between techniques, as well as open gaps in current implementations and standardization efforts.

Through this approach, the report not only isolates layer-specific risks but also demonstrates how system-wide security can be improved through coordinated, lightweight, and firmware-centric solutions. The goal is to provide a complete view of IoT security that acknowledges architectural complexity while remaining focused on the practical realities of constrained, real-world deployments.

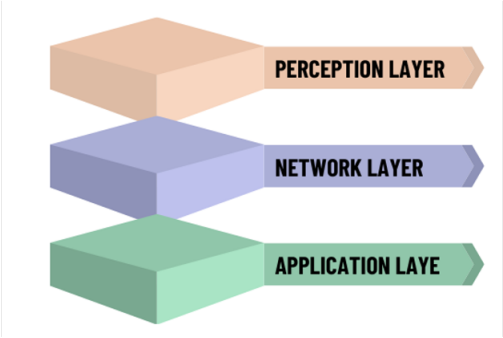

Figure 1: IoT layers overview.

3 LAYER BASED SECURITY EVALUATION

3.1 IoT Architecture and Security Fundamentals

Most IoT systems operate on a client-server communication model, where resource-constrained edge devices function as clients that sense, act, or transmit data, and centralized servers or cloud platforms act as control points that aggregate, process, and respond. These servers often provide remote management capabilities, including access control, data analytics, and firmware distribution. This architectural separation enables scalable deployment and forms the structural foundation for the layered design of IoT systems.

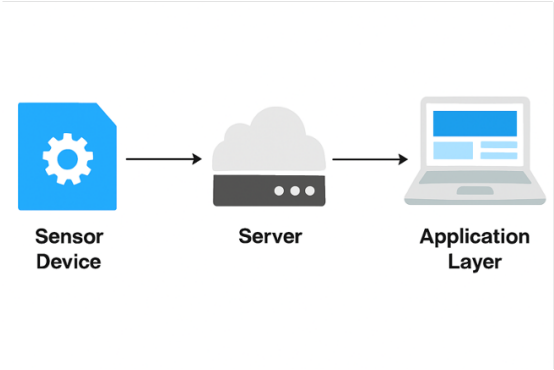

Figure 2: Basic client-server diagram of IoT.

IoT architectures are typically composed of three logical layers: perception, network, and application. This layered model reflects the vertical flow of information and control through the system, from physical data acquisition to high-level decision-making and user interaction.

The perception layer consists of hardware components such as sensors, actuators, embedded processors, and radio-frequency identification tags. These devices operate at the physical edge of the network and are often subject to environmental exposure and resource constraints. The firmware that governs their behavior is particularly critical, as its compromise may result in sensor falsification, unauthorized data collection, or device reprogramming.

The network layer handles the communication between perception-layer devices and cloud or edge computing platforms. It comprises wireless and wired technologies such as Wi-Fi, Zigbee, LoRa, Ethernet, and 5G. Security threats at this layer include man-in-the-middle attacks, packet interception, identity spoofing, and denial-of-service attempts. Inconsistent protocol implementations and lack of end-to-end encryption further compound the risks.

The application layer is where aggregated data is visualized, interpreted, and acted upon. It includes cloud dashboards, mobile applications, API endpoints, and control platforms. This layer is frequently targeted through weak authentication schemes, hardcoded credentials, or flawed access control mechanisms. Improper sanitization of input data, misconfigured privilege levels, and insufficient logging also contribute to vulnerabilities at this level.

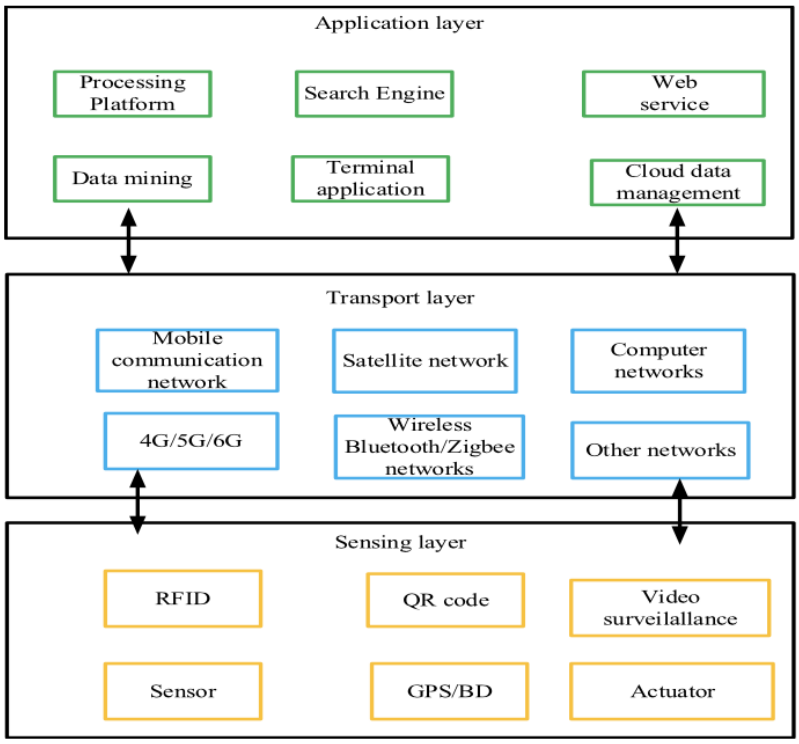

Figure 3: Layered IoT structure.

Firmware spans all three layers by acting as the interface between hardware and networked software. It dictates device boot behavior, communication routines, and policy enforcement, making it a systemic trust anchor. Without secure firmware, higher-layer protections can be bypassed or rendered ineffective. To build resilient systems, security must be distributed across all layers and must account for the interactions between them. A defense-in-depth approach—where each layer implements context-appropriate safeguards—is essential for limiting the scope of compromise and ensuring reliable operation in adversarial environments.

3.2 Perception Layer: Threats and Protections

The perception layer in IoT systems consists of physical components such as sensors, actuators, RFID modules, and embedded controllers that collect and interact with data from the physical world. These devices serve as the entry point to the entire IoT architecture, and any compromise at this layer can undermine the integrity of the entire system. Due to their direct exposure to the environment and limited processing capabilities, perception-layer components are particularly vulnerable to both physical and software-based attacks.

3.2.1 Threat Landscape

Devices at the perception layer often operate on low-cost MCUs with minimal security features. This makes them susceptible to hardware tampering, firmware extraction, or malicious reprogramming. One of the most common threats is signal spoofing, where an attacker forges false sensor input to manipulate system behavior. For example, in industrial automation, a forged temperature reading may trigger incorrect safety actions or resource allocation.

Another frequent threat is eavesdropping or RFID skimming, where an adversary intercepts sensitive data such as identity codes or sensor readings during transmission. Passive RFID systems and unencrypted serial communication protocols make this a significant concern, especially in logistics and smart access control systems.

Side-channel attacks also pose a risk. These attacks exploit physical characteristics of device operation—such as power consumption or electromagnetic emissions—to infer sensitive information like cryptographic keys or system commands. Since many perception-layer devices are designed for energy efficiency rather than security, these leakage channels are rarely addressed by default.

Firmware manipulation is perhaps the most severe threat, allowing attackers to implant persistent malware or alter device behavior in undetectable ways. Devices without secure boot or code verification mechanisms are particularly vulnerable to unauthorized firmware updates—either locally via serial ports or remotely via poorly protected OTA systems.

3.2.2 Protection Techniques

Effective protection of the perception layer begins at the hardware level. Devices should incorporate physical security measures such as casing seals, tamper-evident enclosures, or board-level protections. At the chip level, including secure elements or TPMs can enable hardware-based key storage and cryptographic operations, even in constrained environments.

Secure boot mechanisms verify firmware integrity during startup, preventing unauthorized code execution. This process typically involves cryptographic signature checks using embedded public keys and can be enforced at the bootloader stage.

To counter signal spoofing and eavesdropping, perception-layer devices must use encrypted communication—even if resource-constrained. Lightweight implementations of AES or stream ciphers can protect serial or radio communication channels with minimal performance cost.

In systems where devices are remotely managed, all firmware updates should be digitally signed and verified. OTA

update mechanisms must include rollback protection to prevent attackers from reinstalling vulnerable firmware versions. Secure key management practices are essential to ensure that only trusted entities can sign and deliver firmware. Runtime attestation, though still an emerging concept in embedded systems, can help detect unauthorized modifications during normal operation. Combined with periodic integrity checks and anomaly-based behavior monitoring, these techniques can improve the resilience of perception-layer components.

Ultimately, protecting the perception layer is about ensuring that the hardware, firmware, and communication pathways of edge devices cannot be easily manipulated, impersonated, or subverted. As the foundation of the IoT stack, security failures at this level can propagate upward and compromise the reliability of the entire system.

3.3 Network Layer: Secure Communication and OTA Risks

The network layer in IoT systems is responsible for the transmission of data between edge devices, gateways, and cloud services. It also serves as the primary channel for OTA firmware updates. Due to its central role in enabling both control and communication, the network layer is a frequent target for adversaries aiming to intercept, manipulate, or disrupt IoT traffic.

3.3.1 Threat Landscape

The most prevalent threat to the network layer is the MITM attack, where an attacker silently intercepts or alters messages between legitimate parties. In IoT contexts, this may allow the attacker to steal authentication tokens, modify sensor readings, or inject malicious control commands. Many IoT communication protocols, such as MQTT and CoAP, lack default encryption or authentication mechanisms, making them especially vulnerable to MITM when deployed in plaintext [4].

Replay attacks are also common, particularly in broadcast environments. In these attacks, a valid message (such as a firmware update command) is captured and resent at a later time to trick the device into executing unauthorized operations. Without message freshness validation or update version control, these attacks can compromise even authenticated systems. DoS attacks threaten the availability of IoT services by overwhelming network bandwidth or exhausting limited device resources through packet flooding. Since many IoT devices have limited buffer space and are unable to manage excessive traffic, DoS attacks can be initiated with minimal effort and hardware.

Furthermore, the increasing reliance on wireless communication, particularly low-power wide-area networks such as LoRa and NB-IoT, brings protocol-specific vulnerabilities including jamming, routing misdirection, and identity spoofing. In heterogeneous networks, inconsistent security implementations across protocol stacks also lead to weak links that adversaries can exploit to pivot between devices. Perhaps the most critical threat is the misuse of the network

layer to distribute malicious firmware through insecure OTA update mechanisms. If updates are not cryptographically validated, or if communication is not encrypted, attackers can inject corrupted firmware images into the update process, enabling persistent control of edge devices.

3.3.2 Protection Techniques

Securing the network layer requires a combination of transport-level encryption, authentication, and traffic validation mechanisms, designed with the constraints of embedded systems in mind. One of the most effective solutions is the use of DTLS, a datagram-friendly variant of TLS that offers confidentiality, integrity, and authentication over UDP-based protocols. DTLS has been successfully adapted to constrained devices, particularly when implemented with ECC, which provides strong encryption with shorter key lengths and reduced computational overhead. To prevent MITM and replay attacks, message integrity codes, nonces, and timestamps should be embedded in all communication flows. Sequence numbers or update version counters can also be used to reject outdated or duplicated packets, which is especially important in OTA firmware delivery pipelines.

Firmware updates should always be signed using asymmetric cryptography, and update packages must be transmitted over encrypted channels. Devices should validate both the origin and integrity of updates before installation. Additionally, broadcast or multicast OTA distribution schemes should be carefully managed, as they present unique risks related to spoofed dissemination and denial-of-update attacks. Rate-limiting mechanisms and lightweight intrusion detection rules can be employed at gateways to filter anomalous traffic patterns and throttle suspicious behavior. This is particularly important in scenarios involving shared radio spectrum or narrowband communication.

Finally, efforts should be made to standardize and unify the security configurations across mixed-protocol deployments. Secure gateways can act as translators and validators between devices using different communication technologies, ensuring that end-to-end security is preserved even when devices operate under different stacks.

Through these measures, the network layer can evolve from a common point of compromise into a robust foundation for data confidentiality, device integrity, and trusted firmware updates in IoT systems.

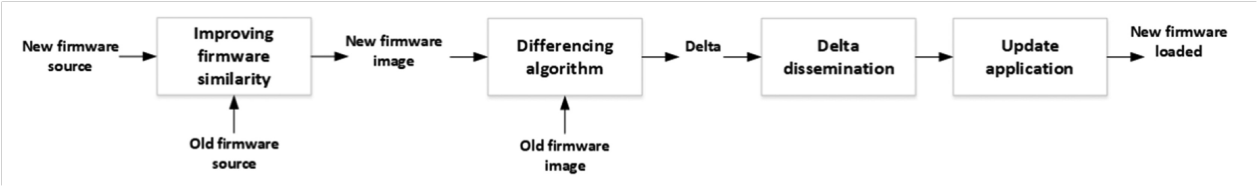

Figure 4: Firmware update process essential stages.

3.4 Application Layer: Access Control and Privacy

The application layer in IoT systems serves as the central interface between users and devices. It hosts the logic for automation, data analytics, visualization dashboards, and control commands. In many cases, it also manages access to device settings, OTA update scheduling, and system configurations. Due to its position at the top of the IoT stack, this layer governs how and by whom data and services are consumed—making it a frequent target for privilege escalation, privacy violations, and unauthorized control.

3.4.1 Threat Landscape

One of the most common threats in the application layer is improper access control. Many IoT platforms rely on static or poorly defined permission models that fail to distinguish between different users, roles, or devices. This creates opportunities for unauthorized access to sensitive functions, such as disabling alarms, unlocking smart doors, or issuing firmware update commands. Weak or hardcoded credentials further expose application interfaces, especially when devices expose APIs or web dashboards over the internet. Adversaries can exploit these endpoints through credential stuffing, brute-force attacks, or injection vulnerabilities.

Data privacy violations also pose a significant risk at the application level. IoT systems often collect detailed personal, environmental, or behavioral information that, if mishandled, can lead to surveillance, identity inference, or regulatory non-compliance. In some cases, even metadata—such as device status logs or update timestamps—can reveal usage patterns or location information. Another challenge is the triggering of unauthorized OTA firmware updates through manipulated application logic. If update processes are linked to unsecured user actions or unverified external inputs, attackers can push devices to install corrupted or outdated firmware images.

Finally, the lack of audit logging and insufficient anomaly detection mechanisms at the application layer make it difficult to identify, trace, or respond to misuse after it occurs.

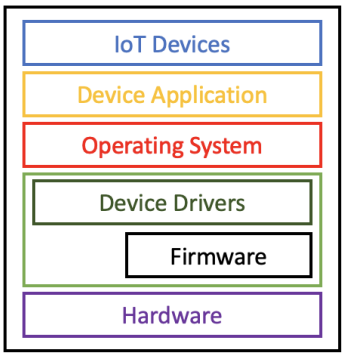

Figure 5: IoT device structure.

3.4.2 Protection Techniques

The most fundamental defense at the application layer is the implementation of structured access control frameworks. RBAC models define permissions based on predefined user roles (e.g., administrator, guest, service), which helps isolate responsibilities. ABAC extends this by evaluating access decisions based on contextual attributes, such as device location, time of request, or user identity. UCON introduces dynamic access policies that consider the ongoing usage state of the resource, supporting real-time revocation and adaptive control. To protect sensitive interfaces, strong authentication mechanisms must replace hardcoded or default credentials. Multi-factor authentication, token-based sessions, and secure API keys should be enforced for all user-device interactions, especially for remote or high-privilege operations.

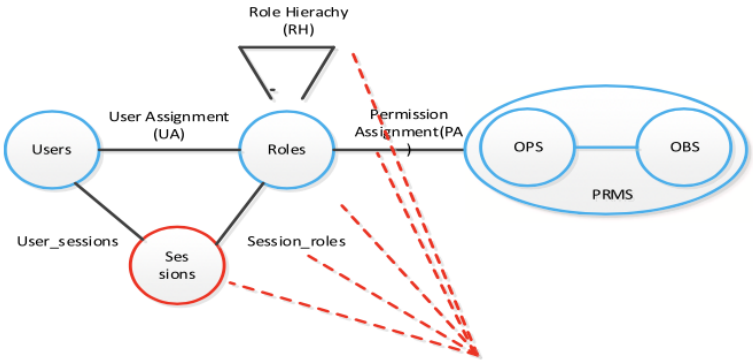

Figure 6: RBAC model.

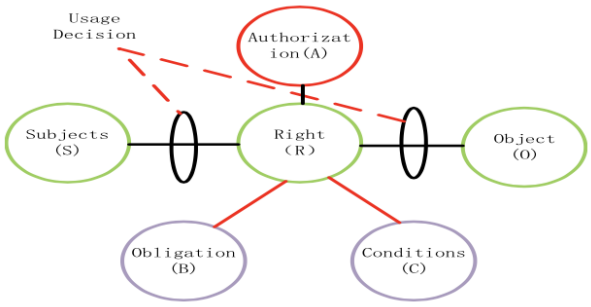

Figure 7: UCON model.

In terms of privacy, several emerging techniques have been proposed. Federated learning enables devices to collaboratively train machine learning models without sharing raw data, reducing the risk of centralized leaks. PEKS and other searchable encryption methods allow cloud-based systems to perform keyword searches on encrypted data without decrypting it—preserving confidentiality even during analytics. Blockchain technologies can be integrated at the application layer to create immutable audit trails of access requests and firmware updates. This not only provides traceability but also helps prevent rollback or forged updates initiated through the application logic.

Finally, application platforms should enforce secure software development practices, including regular penetration testing, input validation, and proper logging of all critical operations. Behavior-based anomaly detection systems can be deployed to detect unexpected usage patterns, which may indicate compromised accounts or misconfigured privileges.

By deploying these protection strategies, the application layer can enforce granular control, preserve user privacy, and serve as a secure endpoint for system-wide operations in IoT environments.

4 SYSTEM-WIDE SECURITY MECHANISMS

4.1 Firmware Update Pipeline and Device Trust

Firmware plays a foundational role in the operation and security of IoT devices. It initializes hardware components, manages communication protocols, and enforces local control logic. Unlike application-layer software, firmware typically operates with elevated privileges and has direct access to physical interfaces and memory. For this reason, firmware is often referred to as the root of trust in IoT systems. However, this same centrality makes it a high-value target for attackers. In many real-world deployments, firmware is rarely updated after installation—or worse, is updated through insecure channels. Without proper protections, malicious actors can modify firmware to implant persistent backdoors, alter device behavior, or disable security features entirely. The integrity of the firmware update pipeline is therefore critical to overall system trustworthiness.

4.1.1 Firmware Update Pipeline

The firmware update process in modern IoT systems generally consists of four key stages:

1. **Firmware Generation**

   A new firmware image is compiled, hashed, and digitally signed by a trusted source. Metadata such as version number, device model compatibility, and hash values are appended to the package.

2. **Dissemination**

   The firmware package is transmitted to the target device(s), either via direct connection or wirelessly through OTA protocols. In OTA scenarios, firmware may traverse multiple intermediate nodes or gateways.

3. **Validation and Verification**

   Upon receipt, the device performs integrity checks using hash functions (e.g., CRC or SHA-256), verifies the digital signature using stored public keys, and compares the version metadata against its local configuration.

4. **Installation and Rollback Protection**

   If all checks pass, the firmware is written to flash and activated on reboot. Secure systems often retain a previous version for rollback, but must block reversion to vulnerable versions.

Any weakness at one of these stages can lead to critical compromise. For instance, unsigned firmware packages, unencrypted transport, or the absence of rollback protection can each be exploited to gain persistent unauthorized control.

4.1.2 Device Trust Anchors

Establishing device trust begins with secure boot, which ensures that only verified firmware is executed during system startup. This process involves cryptographically verifying the bootloader and subsequent firmware layers before handing over control to the operating system. Root keys used in this process are typically stored in hardware-protected areas, such as One-Time Programmable (OTP) memory or secure elements embedded in the MCU.

Firmware signature verification prevents unauthorized updates from being installed. This involves checking the digital signature of the firmware image using asymmetric cryptography. Only images signed by a trusted private key are accepted, and the corresponding public key is hardcoded or securely provisioned on the device. To ensure update integrity in transit, firmware packages should be encrypted or authenticated using schemes like HMAC or authenticated encryption with associated data (AEAD). These prevent tampering or man-in-the-middle manipulation during network transmission.

Version control and anti-rollback mechanisms are essential to prevent attackers from reinstalling older, vulnerable firmware images. Devices should store the current firmware version and reject any incoming package with a lower revision level. Lastly, systems with dual-bank firmware architecture can maintain a backup image that is known to be safe. This allows devices to recover gracefully from failed updates or partial installations without entering an unrecoverable state.

Together, these components form the basis of a secure firmware update pipeline. By embedding verification at every stage—generation, distribution, validation, and execution—IoT systems can protect themselves against a wide range of firmware-level threats and maintain long-term operational trust.

4.2 Cross-Layer Security Integration

IoT security cannot be effectively addressed by treating each architectural layer in isolation. The integrity and trustworthiness of the overall system depend on how security mechanisms are distributed and coordinated across the perception, network, and application layers. This interdependence, known as cross-layer security, is especially important in IoT environments due to the tight coupling between hardware behavior, communication protocols, and application-level logic.

Security failures in one layer often propagate to others. For example, a compromised sensor at the perception layer may inject false data that travels unmodified through the network and triggers incorrect behavior at the application layer. Similarly, a misconfigured access control policy in the application layer can override network-layer protections and expose low-level device functions to unauthorized actors. These cascading effects emphasize the need for a holistic, defense-in-depth approach where multiple layers reinforce each other rather than operate independently.

4.2.1 Data Flow and Trust Propagation

In typical IoT deployments, data originates from the perception layer and flows upward toward applications, while control commands flow downward. Trust must be maintained along this entire path. If the firmware governing sensor behavior is unverified or maliciously modified, even encrypted network protocols or access-controlled APIs cannot guarantee correctness. Therefore, establishing device trust through secure firmware and boot processes forms the foundation upon which higher-layer protections must build. Conversely, application-level decisions—such as who can trigger OTA updates or change device settings—must be enforced in a way that propagates securely down to the hardware level. This requires coordinated authentication, authorization, and validation mechanisms across all layers.

4.2.2 Cross-Layer Enforcement Scenarios

Several practical scenarios highlight the necessity of cross-layer security alignment:

- OTA Updates

  A secure OTA update process begins with authenticated user input (application layer), proceeds through encrypted delivery (network layer), and ends with signature verification on the device (perception layer). Any weak link in this chain can compromise the entire update pipeline.

- Access Control

  Policies defined at the application layer (e.g., using RBAC or ABAC) must correspond to real enforcement mechanisms within device firmware. For instance, disabling a sensor or changing a reporting frequency must not be executable unless validated and explicitly permitted by both the application logic and device-level access routines.

- Anomaly Detection

  Behavioral anomaly detection may involve collecting sensor logs (perception), aggregating them over the network, and running models in the cloud (application). For these models to be trustworthy, the underlying data sources and transport must be authenticated and tamper-resistant.

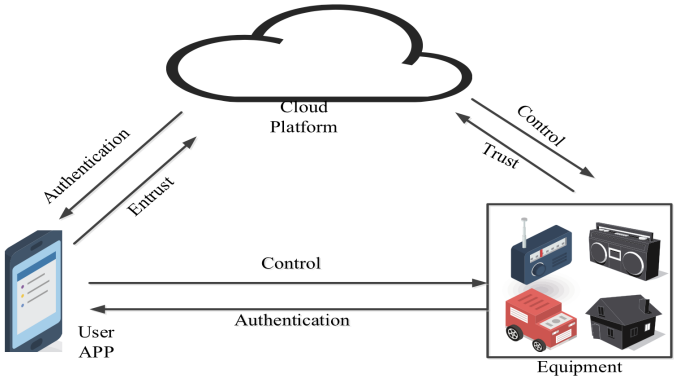

Figure 8: IoT access control based on cloud platform.

4.2.3 Blockchain as a Cross-Layer Tool

Blockchain and distributed ledger technologies offer promising solutions for enabling traceability and verifiability across layers. A blockchain ledger can store immutable records of firmware versions, access requests, and update approvals. Since each event is cryptographically signed and timestamped, this ledger can serve as a shared trust anchor for all layers. Devices can verify the legitimacy of commands or updates by consulting the ledger, and administrators can audit system history without relying on a centralized logging service.

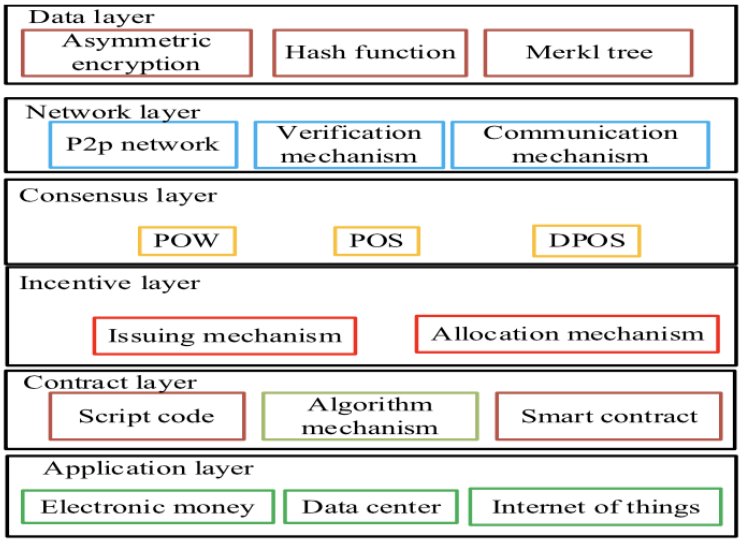

Figure 9: Blockchain technology architecture.

By integrating cross-layer validation mechanisms and enforcing consistency from firmware to cloud logic, IoT systems can achieve greater resilience, detect inconsistencies earlier, and reduce the impact of localized breaches. This layered alignment transforms the system into a security-aware ecosystem where protections are redundant, coordinated, and scalable.

4.3 Evaluation of Techniques and Open Challenges

The layered security strategies analyzed throughout this report present a comprehensive blueprint for addressing the diverse risks inherent in IoT systems. While many of these techniques are conceptually sound and have shown promise in controlled environments, their effectiveness and practicality in real-world deployments remain subject to critical limitations. This section evaluates the key protection mechanisms discussed and outlines several open research and deployment challenges that persist in the IoT security landscape.

4.3.1 Strengths and Contributions of Current Techniques

One of the major strengths of the layered approach is that it allows for compartmentalization of security responsibilities. By tailoring protection mechanisms to the specific threats at each layer—such as lightweight cryptography at the network layer or access control at the application layer—systems can remain efficient without overburdening resource-constrained devices.

Secure boot and firmware signature validation provide a solid foundation for trust at the hardware level, ensuring that only authenticated firmware is executed. OTA updates, when combined with encryption and version management, enable safe and scalable device maintenance without requiring physical access. Additionally, access control frameworks like RBAC and ABAC bring structured policy enforcement to user and system interactions, which is essential for preventing privilege escalation and unauthorized actions.

Emerging technologies such as blockchain offer immutability and transparency in access tracking and firmware update history. Federated learning and other privacy-preserving ML techniques also allow devices to participate in intelligent decision-making without exposing sensitive data to centralized systems.

4.3.2 Practical Limitations

Despite these strengths, several implementation barriers hinder the widespread adoption of these solutions. Many IoT devices operate under severe constraints in terms of memory, processing power, and energy availability. This makes it difficult to integrate traditional cryptographic protocols or public key infrastructure without significant performance degradation.

Another limitation is the lack of standardization across vendors and communication protocols. Diverse device ecosystems lead to inconsistent security implementations, especially in multi-vendor deployments. This fragmentation not only complicates integration but also creates weak links that can be exploited by attackers [5].

While blockchain and distributed ledgers offer exciting opportunities, they also introduce new challenges related to latency, energy consumption, and scalability—particularly in low-bandwidth or intermittently connected environments. Similarly, federated learning requires reliable synchronization and model aggregation mechanisms, which may not be feasible for all edge nodes.

Cross-layer enforcement is also difficult to manage in practice. Although ideal in theory, coordinating authentication, access control, and update validation across layers demands system-wide design coherence, which is rarely achievable in commercial deployments.

4.3.3 Open Research Challenges

Several areas require further research to close the gap between theoretical models and practical deployment:

- Lightweight Security Protocols: More work is needed to develop ultra-efficient encryption and authentication schemes that balance security with minimal resource usage.
- Standardized OTA Frameworks: The lack of unified, cross-platform OTA update protocols remains a critical vulnerability in modern IoT systems.
- Scalable Key Management: Secure, decentralized key provisioning and lifecycle management in large-scale IoT deployments remains largely unsolved.
- AI-Assisted Threat Detection: Integrating real-time ML or AI models at the edge for anomaly detection must overcome limitations in data availability, interpretability, and model robustness.
- Secure Multi-Tenant Architectures: As shared IoT infrastructure becomes more common (e.g., in smart buildings), secure isolation between tenants and policy enforcement must be guaranteed.
- Post-Quantum Readiness: Most IoT cryptographic schemes are not yet designed to resist quantum-level attacks. Research into quantum-safe algorithms for constrained environments is urgently needed.

By addressing these challenges, the IoT security field can move toward systems that are not only secure in theory, but also practical, scalable, and resilient in real-world deployments.

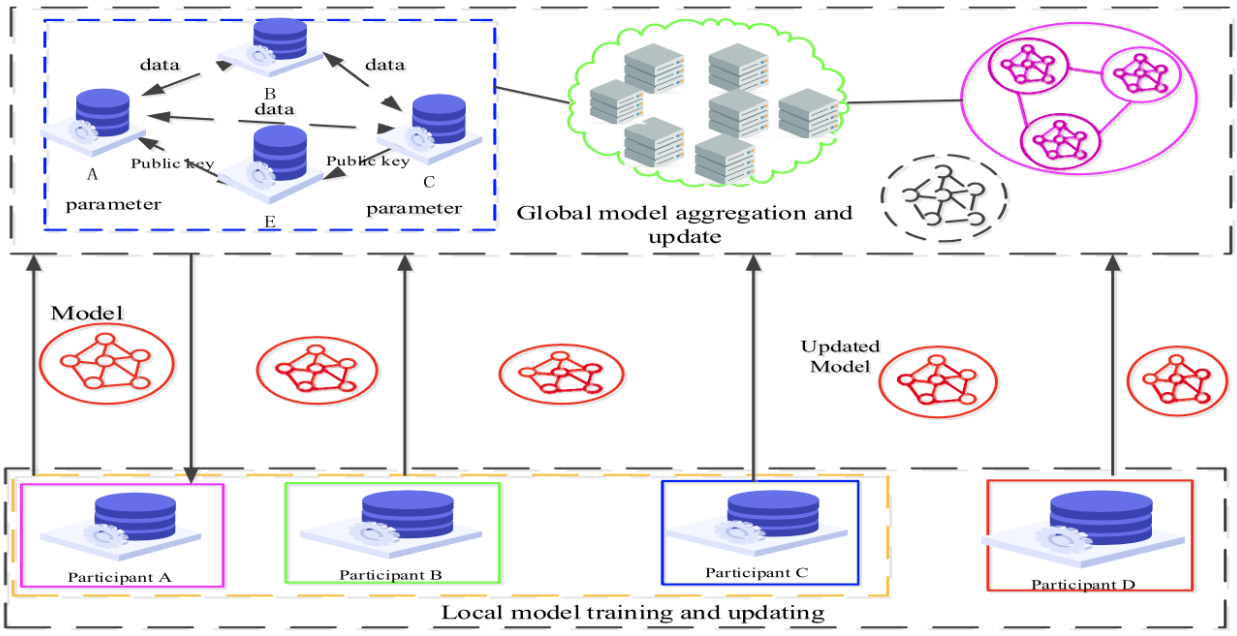

Figure 10: Federated learning model.

5 MAJOR REAL-WORLD IOT SECURITY INCIDENTS

To reinforce the importance of securing IoT systems across different architectural layers, this section presents a selection of high-impact real-world security incidents. Each case highlights the attack vectors used, the architectural layers affected, and whether firmware-level vulnerabilities were involved.

5.1 Mirai Botnet Attack (2016)

The Mirai botnet was responsible for one of the largest distributed denial-of-service (DDoS) attacks ever recorded. It targeted poorly secured IoT devices such as IP cameras and home routers by exploiting default telnet credentials, which were hardcoded into the firmware. The compromised devices were turned into a botnet army that overwhelmed DNS provider Dyn, disrupting access to major websites like Twitter and Netflix. This attack primarily affected the network and application layers and demonstrated how insecure firmware configurations could be weaponized at scale.

5.2 Verkada Camera Breach (2021)

In this incident, hackers accessed the internal networks of Verkada Inc., a provider of cloud-based surveillance systems. They exploited administrative credentials to gain entry to the company's camera infrastructure, exposing live video feeds from over 150,000 devices installed in sensitive locations such as hospitals and prisons. While the breach occurred at the application and network layers, it also involved firmware in an indirect manner, as some access persistence was achieved via device-level configurations that were not securely hardened.

5.3 Philips Hue Lightbulb Worm (2016)

Security researchers demonstrated a vulnerability in Philips Hue smart bulbs that allowed a malicious firmware update to be propagated wirelessly over the Zigbee protocol. Because the over-the-air (OTA) update mechanism lacked cryptographic signature verification, an attacker could spread a worm from bulb to bulb across a city-wide mesh network. This incident directly involved the perception and application layers and highlighted the importance of securing firmware update processes.

5.4 Mozi Botnet (2019–2022)

Mozi is a peer-to-peer botnet that infected hundreds of thousands of IoT devices, including routers and digital video recorders. It exploited open Telnet and SSH services to brute-force access using weak or default credentials, targeting devices running outdated firmware. The botnet persisted across reboots, maintained command-and-control channels,

and enabled further attacks. This threat primarily affected the network layer and directly involved firmware-level vulnerabilities that had not been patched or disabled.

5.5 Jeep Cherokee Remote Hijack (2015)

Researchers Charlie Miller and Chris Valasek remotely hijacked a Jeep Cherokee by exploiting a zero-day vulnerability in its Uconnect infotainment system. The system's firmware allowed remote code execution through the cellular network, giving the attackers control over steering, brakes, and transmission. This attack involved both the perception and network layers and was a clear example of the critical safety risks that can result from firmware vulnerabilities in automotive IoT systems.

6 CONCLUSION

6.1 Conclusion

The rapid expansion of IoT technologies has created a new generation of connected systems that are deeply embedded in critical infrastructure, industrial environments, and personal spaces. However, this unprecedented level of integration brings equally significant security and privacy risks that demand specialized, layered protection strategies. This report has examined those risks through a comprehensive evaluation of IoT system architecture, with an emphasis on firmware security and OTA update integrity. By analyzing threats and defenses at the perception, network, and application layers, the study highlights how each layer presents unique challenges that cannot be solved with traditional IT security models. From physical tampering and signal spoofing at the device level to data interception and malicious control at higher layers, IoT systems require targeted solutions that are both lightweight and context-aware.

Firmware has emerged as a central point of trust—and vulnerability—within the IoT stack. Compromised firmware can bypass higher-layer defenses and persistently alter system behavior. The secure management of firmware updates, especially over wireless OTA channels, is therefore critical to maintaining device integrity, operational stability, and system trust. To mitigate these risks, the report has surveyed a variety of security mechanisms: secure boot and signature validation to protect firmware; DTLS and ECC for securing communication channels; RBAC, ABAC, and UCON for fine-grained access control; and blockchain and federated learning for decentralized integrity and privacy protection. While these techniques provide a strong starting point, their practical deployment remains limited by constraints such as resource availability, lack of standardization, and complex cross-layer dependencies.

Moving forward, IoT security must be approached as a system-wide design concern, not as an afterthought or add-on. Success will depend on continued research into scalable, efficient security models that balance protection with performance—especially in constrained, distributed environments. By integrating layered defenses, establishing trustworthy firmware infrastructures, and aligning application policies with device capabilities, the IoT ecosystem can evolve toward secure, privacy-preserving, and resilient deployments.

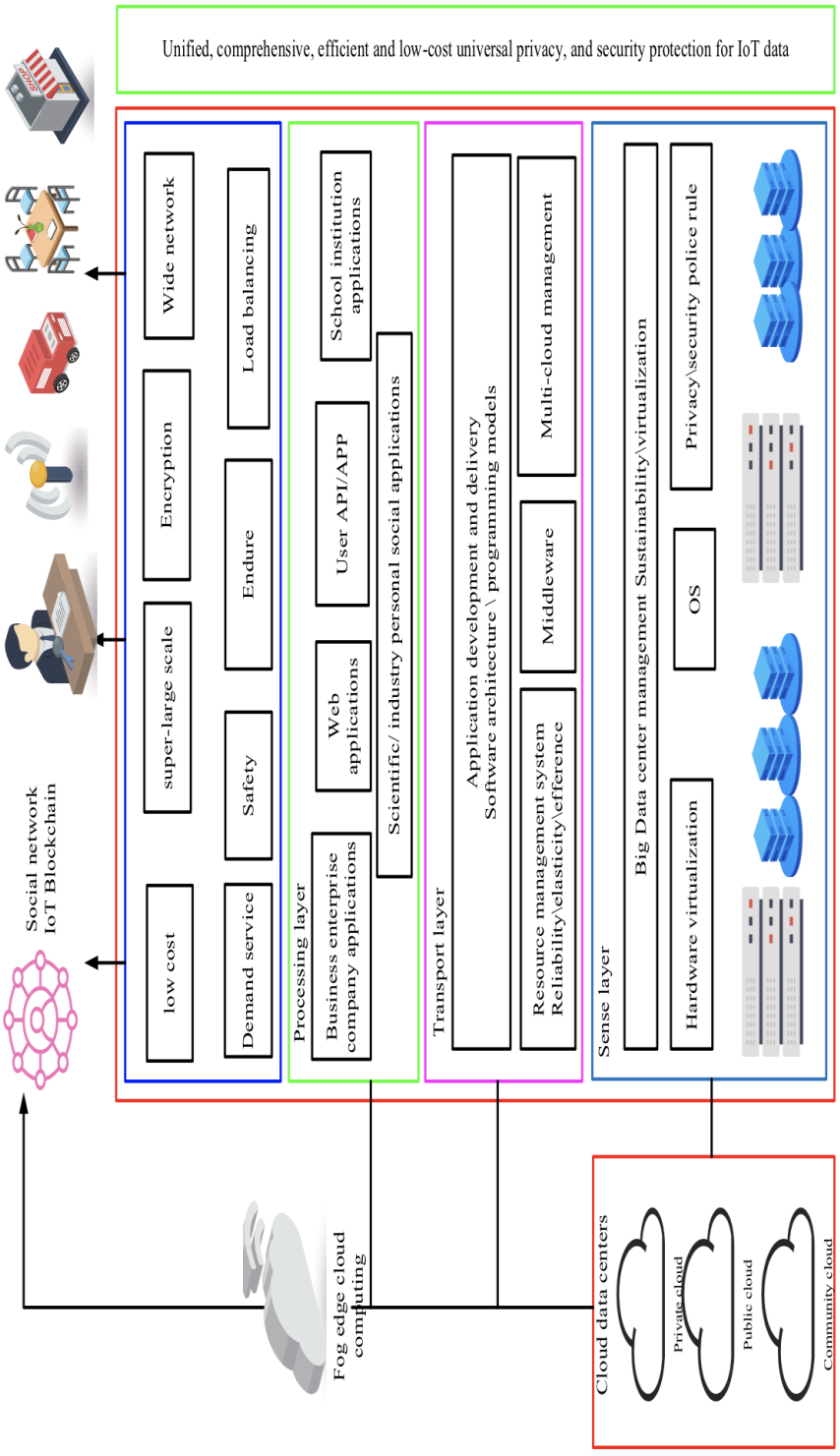

Figure 11: A future architecture for privacy and security protection of the IoT integrating cloud, edge computing.

REFERENCES

[1] Statista, "Number of connected iot devices worldwide 2019–2025," https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, 2024, accessed: 2025-05-10.

[2] P. Sun, Y. Wan, Z. Wu, Z. Fang, and Q. Li, "A survey on privacy and security issues in iot-based environments: Technologies, protection measures and future directions," *Computers Security*, vol. 148, p. 104097, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824004024

[3] K. Arakadakis, P. Charalampidis, A. Makrogiannakis, and A. Fragkiadakis, "Firmware over-the-air programming techniques for iot networks – a survey," 2020. [Online]. Available: https://arxiv.org/abs/2009.02260

[4] LoRa Alliance, "Lorawan® specification v1.0.4," https://lora-alliance.org/resource_hub/lorawan-specification-v1-0-4/, 2023, accessed: 2025-05-10.

[5] National Institute of Standards and Technology (NIST), "Considerations for managing internet of things (iot) cybersecurity and privacy risks," U.S. Department of Commerce, Tech. Rep. NIST SP 800-213, 2020.