UDC 004.056

SECURITY-BY-DESIGN PRINCIPLES FOR EMBEDDED AND IoT SYSTEMS

F. Badshahzoi, B.A. Mailykhanova

Kazakh National Agrarian Research University, Almaty, Kazakhstan, farhadbadshahzoi.uni@gmail.com

Abstract: this paper proposes a five-layer security-by-design model for embedded and Internet of Things systems and validates its structural coverage against selected requirements from NISTIR 8259A, NIST SP 800-213, and ETSI EN 303 645. The layers cover hardware trust, device integrity, identity and access, communication and data protection, and lifecycle security. A traceability assessment for a smart-agriculture IoT architecture maps ten security requirement groups to design layers and implementation controls. All selected requirement groups are addressed, while the assessment also identifies device identification, data protection, secure updates, state awareness, and recovery as controls that must span multiple system components.

Keywords: security-by-design, IoT security, embedded systems, secure boot, authentication, firmware security, cybersecurity.

I. INTRODUCTION

Embedded and Internet of Things (IoT) systems support agriculture, healthcare, transport, industrial automation, and smart buildings. Their connectivity enables continuous data collection and remote control but creates additional attack paths through devices, gateways, communication links, cloud services, and software-update channels. Resource constraints and long operational lifetimes further complicate the use of conventional security mechanisms.

Security-by-design integrates protection into requirements, architecture, implementation, deployment, and maintenance. NISTIR 8259A defines a core baseline of device cybersecurity capabilities [1], NIST SP 800-213 connects device capabilities with organizational security requirements [2], and ETSI EN 303 645 establishes baseline provisions for consumer IoT products [3]. These publications define what capabilities are expected, but a compact architectural view is still useful when students and system designers must place controls within a concrete embedded solution.

The objective of this work is to develop a five-layer model that translates established security requirements into architectural design decisions and to evaluate its traceability using a smart-agriculture IoT example. The contribution is not a new cryptographic algorithm; it is a reproducible mapping from requirement groups to security layers, components, and verifiable controls.

II. THREATS AND DESIGN REQUIREMENTS

Relevant threats include device impersonation, unauthorized access, interception and modification of data, extraction of credentials, execution of altered firmware, installation of malicious updates, and loss of security visibility after deployment. A single control cannot mitigate all these threats. For example, transport encryption protects data in transit but does not prove that a device is executing authorized firmware; authentication restricts access but does not provide a recovery path after compromise.

The requirement set used for the assessment was constructed from the device capabilities in NISTIR 8259A, system-level guidance in NIST SP 800-213, firmware resiliency guidance in NIST SP 800-193 [4], and relevant ETSI baseline provisions. Closely related provisions were consolidated into ten requirement groups to avoid counting repeated wording across standards. This consolidation supports traceability analysis; it does not claim formal product certification [5].

III. PROPOSED FIVE-LAYER MODEL

The model separates controls according to the trust dependency they establish. Layer 1 provides hardware-supported trust and protects key material. Layer 2 verifies software integrity during boot and supports recovery. Layer 3 controls device and user identities, authentication, and authorization. Layer 4 protects data and communications. Layer 5 maintains security through updates, monitoring, vulnerability management, and incident recovery. Higher layers depend on lower-layer trust, while monitoring and recovery interact with the entire lifecycle (Fig. 1).

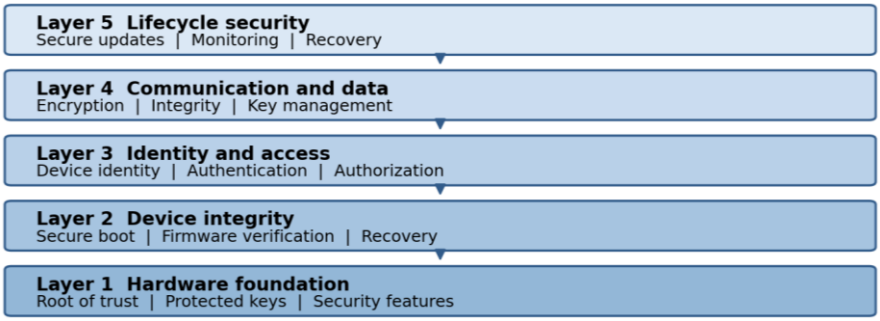


Figure 1. Five-layer security-by-design model

IV. TRACEABILITY ASSESSMENT

The model was applied to a smart-agriculture architecture containing soil and climate sensors, an IoT gateway, a cloud platform, and a management application. Each requirement group was mapped to a primary layer, a target component, and an observable implementation control. A requirement was counted as covered only when the model identified both its architectural location and a control that could be inspected or tested (Tab.1).

Table 1. Traceability of selected requirements to the proposed model

Requirement group	Layer	Target	Verifiable control
Device identification	3	Sensor, gateway	Unique device identifier
Authentication	3	All interfaces	Mutual authentication
Authorization	3	Gateway, cloud	Role-based access policy
Data confidentiality	4	Links, storage	Encryption enabled
Data integrity	4	Messages, logs	Integrity tag or signature
Software integrity	2	Sensor, gateway	Verified secure boot
Secure update	5	Sensor, gateway	Signed update accepted
Key protection	1	Device hardware	Protected key storage
Security state awareness	5	Gateway, cloud	Events and alerts recorded
Recovery and reset	2, 5	Device, operator	Trusted recovery procedure

All ten selected requirement groups were mapped to at least one layer and one observable control, giving structural coverage of 10/10 for the defined assessment set. Four groups require coordination across more than one component: authentication, confidentiality, integrity, and state awareness. Recovery spans Layers 2 and 5 because a trusted technical recovery mechanism must be accompanied by an operational response procedure. The result shows that the layered model is complete for the selected baseline, but it does not demonstrate resistance to every attack or replace platform-specific security testing.

The assessment also reveals a design distinction that is easy to miss in a flat checklist. Secure boot and protected key storage are device-local controls, whereas access policy, monitoring, and vulnerability response require gateway, cloud, and organizational participation. Therefore, assigning every requirement only to the sensor would leave system-level gaps even if the embedded firmware were correctly protected.

V. IMPLEMENTATION GUIDANCE

For implementation, each sensor should receive a unique identity during provisioning and authenticate before exchanging data with the gateway. Communication should use an appropriate secure protocol with managed keys. The gateway should enforce authorization, validate message integrity, record security-relevant events, and restrict administrative interfaces. Devices should verify firmware before execution and accept only authenticated updates. The cloud platform should protect stored data, manage privileges, support credential rotation, and retain monitoring information. Recovery procedures should define how compromised credentials are revoked and how a device returns to a trusted state.

Resource limitations require proportional controls. Hardware-backed key storage and secure boot are preferable where supported. On constrained devices, cryptographic algorithms, certificate profiles, logging volume, and update mechanisms must be selected according to memory, energy, bandwidth, and maintenance constraints. The traceability table can be used during design review: a missing target component or unverifiable control indicates that the corresponding requirement has not yet been translated into an implementable decision.

VI. THREAT-SCENARIO ANALYSIS

To examine how the layers cooperate, five representative attack scenarios were traced through prevention, detection, and recovery activities. The scenarios were selected because they affect different parts of the smart-agriculture architecture and cannot be addressed by a single mechanism. Table 2 records the principal layer interaction for each scenario. The entries describe expected security behaviour rather than measured attack success rates (Tab. 2).

Table 2. Layer interaction for representative attack scenarios

Attack scenario	Preventive control	Detection evidence	Recovery action
Cloned soil sensor	Unique identity and mutual authentication (L3)	Repeated identity or failed authentication alert (L5)	Revoke credential and reprovision device (L5)
Telemetry modification	Encryption and message-integrity protection (L4)	Integrity verification failure at gateway (L4, L5)	Reject message and rotate affected keys (L4, L5)
Altered firmware	Verified boot and protected trust anchor (L1, L2)	Boot verification failure (L2)	Load trusted recovery image (L2)
Malicious update	Signed update package and rollback protection (L2, L5)	Invalid signature or version event (L5)	Block update and retain approved version (L2, L5)
Compromised cloud account	Role-based access and strong authentication (L3)	Abnormal login or privilege-use alert (L5)	Disable account, revoke sessions, review logs (L5)

The scenario analysis demonstrates three recurring dependencies. First, identity mechanisms require lifecycle support: a unique credential is useful only if it can be revoked, replaced, and audited. Second, integrity must be checked at more than one boundary. Firmware verification protects execution on the device, whereas message verification protects data exchanged with the gateway and cloud. Third, recovery must preserve trust. Restoring an unverified image or reusing a compromised credential could return the system to operation without removing the original weakness.

For design review, the combined results of Tables 1 and 2 provide two complementary checks. The traceability matrix verifies that each selected requirement has an architectural location and observable control. The scenario matrix verifies that controls cooperate across the sequence of prevention, detection, containment, and recovery. A design fails the review when a requirement has no owner, when an event cannot be observed, or when no trusted recovery action is defined. This rule makes the model usable as a compact checklist before implementation and as a basis for later security testing.

The proposed layers also clarify the different roles of the reference documents. NISTIR 8259A is centred on technical capabilities that an IoT device should provide or support, including identification, configuration, data protection, interface restriction, software update, and awareness of cybersecurity state. NIST SP 800-213 places these capabilities in an organizational risk-management context and helps a purchaser express device requirements. ETSI EN 303 645 adds product-oriented provisions such as avoiding universal default passwords, maintaining update mechanisms, protecting sensitive parameters, minimizing exposed attack surfaces, and supporting secure deletion. The layered model does not replace these sources; it provides a common architectural index through which their provisions can be assigned to components.

This distinction is important in the smart-agriculture example. The sensor manufacturer is responsible for hardware trust, firmware verification, protected credentials, and update support. The system integrator configures identities, communication protection, gateway filtering, and cloud permissions. The operator monitors events, manages credentials, applies updates, and executes recovery procedures. If responsibility is not assigned, a technically available capability may remain disabled or unmanaged. Accordingly, the design-

review record should identify a responsible role, evidence source, and verification method for every row of the traceability matrix.

The verification method can be refined during implementation. Device identity can be checked by attempting enrollment with duplicated credentials; communication protection can be reviewed through protocol configuration and packet inspection; secure boot can be tested with an altered firmware image in a controlled environment; update protection can be checked using an unsigned package; and security-state awareness can be assessed by confirming that failed authentication, integrity errors, and update failures generate timestamped events. Recovery should be tested only on a non-production device using an approved image and documented credential-revocation procedure. These tests would convert structural coverage into measurable implementation evidence.

The analysis has limitations. The ten requirement groups are a consolidated educational baseline and do not reproduce every provision of the cited standards. Structural coverage shows completeness of the mapping, not resistance to real adversaries. Hardware platform, protocol, cryptographic configuration, and operational environment can change the effectiveness and resource cost of each control. These factors should be evaluated experimentally when a target device and software stack are selected.

VII. CONCLUSION

A five-layer security-by-design model was developed for embedded and IoT systems. The model organizes hardware trust, device integrity, identity and access, communication and data protection, and lifecycle security into dependent architectural layers. A traceability assessment for a smart-agriculture IoT architecture mapped ten requirement groups derived from NIST and ETSI guidance to system components and verifiable controls. All ten groups were structurally covered, while the analysis identified several controls that must span the device, gateway, cloud platform, and operational process. The model can support early design reviews and teaching activities. Future work should implement the controls on a selected IoT platform and measure security effectiveness, latency, memory use, and energy overhead.

REFERENCES

[1] Fagan M., Megas K. N., Scarfone K., Smith M. IoT Device Cybersecurity Capability Core Baseline. NISTIR 8259A. National Institute of Standards and Technology, 2020. DOI: 10.6028/NIST.IR.8259A.

[2] Fagan M., Marron J., Brady K., Cuthill B., Megas K., Herold R., Lemire D., Hoehn B. IoT Device Cybersecurity Guidance for the Federal Government: Establishing IoT Device Cybersecurity Requirements. NIST SP 800-213. National Institute of Standards and Technology, 2021. DOI: 10.6028/NIST.SP.800-213.

[3] ETSI. Cyber Security for Consumer Internet of Things: Baseline Requirements. ETSI EN 303 645 V3.1.2, 2024.

[4] Regenscheid A. Platform Firmware Resiliency Guidelines. NIST SP 800-193. National Institute of Standards and Technology, 2018. DOI: 10.6028/NIST.SP.800-193.

[5] Schiller E., Aidoo A., Fuhrer J., Stahl J., Ziörjen M., Stiller B. Landscape of IoT Security // Computer Science Review. 2022. Vol. 44. Art. 100467. DOI: 10.1016/j.cosrev.2022.100467.

ПРИНЦИПЫ БЕЗОПАСНОГО ПРОЕКТИРОВАНИЯ ВСТРОЕННЫХ И IoT-СИСТЕМ

Бадшахзои Ф., Майлыханова Б. А.

Казахский национальный аграрный исследовательский университет, Алматы, Казахстан,
farhadbadshahzoi.uni@gmail.com

Аннотация: предложена пятиуровневая модель безопасного проектирования встроенных систем и Интернета вещей и выполнена проверка структурного покрытия требований NISTIR 8259A, NIST SP 800-213 и ETSI EN 303 645. Модель включает аппаратную основу доверия, целостность устройства, управление идентификацией и доступом, защиту связи и данных, а также безопасность жизненного цикла. Для архитектуры умного сельского хозяйства десять групп требований сопоставлены уровням, компонентам и проверяемым механизмам защиты. Все выбранные группы требований получили архитектурное соответствие; при этом показано, что ряд механизмов должен охватывать устройство, шлюз, облачную платформу и эксплуатационные процессы.

Ключевые слова: безопасное проектирование, безопасность IoT, встроенные системы, безопасная загрузка, аутентификация, безопасность прошивки, кибербезопасность.