International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 9979-9987
http://www.ijcesen.com

ISSN: 2149-9144

Copyright © IJCESEN

Research Article

Differential, Attested, and Clinically-Aware OTA for Android Medical Fleets: A Policy-Driven Orchestration Framework

Riddhi Patel*

New York Institute of Technology, USA
* **Corresponding Author Email:** riddhipatel0915@gmail.com - **ORCID:** 0000-0002-0047-1151

Article Info:

DOI: 10.22399/ijcesen.4585
Received : 16 December 2025
Revised : 21 December 2025
Accepted : 25 December 2025

Keywords

Over-The-Air Updates,
Medical Device Security,
Android Fleet Management,
Software Bill Of Materials,
Clinical Risk Management

Abstract:

Healthcare organizations increasingly deploy Android devices as gateways and embedded systems in regulated environments. Fleet updates remain risky because defective patches can interrupt monitoring, breach compliance, or expose devices to known vulnerabilities. A regulated over-the-air orchestration framework tailored to medical fleets addresses these challenges. The framework combines differential patching using bsdiff-class algorithms with software bill of materials provenance. Android Verified Boot and attestation capabilities strengthen device integrity verification. Phased rollouts bound to clinical risk enable controlled deployment. Automatic rollback prevents widespread service disruption. Policies are enforced using Android Management API controls, including windowed updates, freeze periods, and kiosk constraints. Post-install health probes verify Bluetooth Low Energy reconnection, sensor latency, and application state. Content-risk scoring modulates rollout velocity based on whether changes affect kernel components or user interface elements. Software bill of materials components link directly to known vulnerabilities in public databases. Simulated and pilot deployments demonstrate reduced remediation latency and prevention of care-critical regressions compared with all-at-once update strategies. The design meets the FDA premarket cybersecurity requirements, and NIST Secure Software Development Framework, and attestation logs, software bill of materials, and risk score justifications support regulatory submissions and inspections.

1. Introduction

1.1 Contextual Background

Regulatory expectations for medical device cybersecurity have increased markedly in recent years, and expectations for secure by design continue to evolve across jurisdictions. Vulnerabilities need to be patched, and continuity of care maintained. Over-the-air updates across fleets of Android devices can provide the foundational building block. However, implementations may be challenging for health care organizations because of the unique environment of health care. Patient safety should not be compromised, and care delivery should not be interrupted. Regulations must be complied with during the device lifecycle.Blockchain-based frameworks have emerged as promising solutions for secure medical device management [1]. These frameworks provide cryptographic verification and tamper-proof audit trails. The integration of blockchain with Internet of Medical Things devices enhances the security posture. It enables distributed trust models without central authorities. Healthcare organizations can leverage these technologies for device fleet management.Android devices have proliferated throughout healthcare settings in diverse roles. They serve as bedside tablets for patient information display. They function as remote patient monitoring gateways, collecting physiological data. They operate as interfaces for diagnostic and therapeutic equipment. Some act as smart ward edge nodes processing local analytics. Each device requires regular security updates to address emerging threats. Traditional update mechanisms fall short of healthcare requirements.

1.2 Problem Statement and Gap Analysis

Traditional mobile over-the-air workflows lack clinical-risk awareness in their design. They do not

provide cryptographically attested provenance for software components. Rollback mechanisms are not tied to functional health probes in meaningful ways. Software bill of materials practices remain uneven across device vendors. This complicates compliance audits significantly. It makes vulnerability tracking across the fleet difficult or impossible.The FDA has mandated comprehensive SBOM requirements for medical device manufacturers [2]. To improve transparency in the medical device supply chain, manufacturers must provide component inventories for all software with sufficient detail to identify components. They must maintain accurate version information throughout the device lifecycle. SBOM enables rapid identification of vulnerable components during security incidents. Healthcare organizations need systematic processes to manage these requirements.
Current deployment strategies create several operational problems. Full system updates consume excessive network bandwidth during transfer. They require extended maintenance windows that disrupt clinical operations. Partial update failures leave devices in inconsistent or undefined states. There is no standardized way to modulate rollout speed based on patch criticality. Clinical impact is not factored into deployment timing decisions. Health verification after update installation is absent or inadequate in most implementations.

1.3 Purpose and Scope

A policy-based over-the-air (OTA) framework which integrates several security and operational features is available to address these gaps. The framework integrates differential patches to improve bandwidth efficiency. It uses Android Verified Boot and remote attestation to establish cryptographic trust. It employs Android Management API system-update controls comprehensively. These controls include windowed updates and regulatory freeze periods. Health probes verify clinical function after installation completes successfully. Software bill of materials linkage enables systematic vulnerability tracking.
The scope encompasses the entire update lifecycle from build to verification. Patch generation and cryptographic signing are included. Policy definition and automated enforcement are covered. Rollout orchestration and real-time monitoring are addressed. Verification procedures and automatic rollback are specified. The framework applies to various Android form factors deployed in healthcare. It supports tablets, wireless gateways, and embedded medical systems.

2. Core Discussion-Research and Innovations

2.1 Research Background

Binary differential over-the-air technology reduces payload sizes substantially compared to full images. Algorithms like bsdiff perform byte-level comparison of two software versions. They generate a compact patch file containing only the binary differences. This patch represents the minimal changes needed for transformation. The target device applies the patch to its current version. It reconstructs the new version through the patching process.
Secure firmware over-the-air updates for IoT devices face multiple technical challenges [3]. Authentication mechanisms must verify update sources cryptographically. Confidentiality protections must prevent unauthorized access to firmware images. Integrity verification must detect tampering during transmission. Availability requirements must ensure updates complete successfully despite network interruptions. These security properties are essential for medical device deployments.
Software bill of materials standards have matured significantly over the past several years. CycloneDX and SPDX provide structured, machine-readable formats for component cataloging. They catalog all software components comprehensively, including transitive dependencies. They record precise version information for vulnerability correlation. They track licensing details for compliance management. Software bills of materials improve supply chain provenance tracking substantially.
Permissioned blockchain architectures combined with edge computing enable privacy-preserving operations [4]. This enables decentralized trust in a high-performance manner and allows for automated enforcement of policies via smart contracts. Edge computing has also been used to reduce latency. The combination enables secure, efficient medical device fleet management.

2.2 Novel Contribution

A medical-grade over-the-air orchestration framework introduces several novel architectural elements. First, it implements systematic patch content risk scoring based on affected components. Kernel modifications receive higher risk scores due to stability implications. Driver changes affecting hardware interfaces are treated with particular caution. User interface updates receive lower risk profiles due to reduced system impact. This quantitative scoring directly informs rollout velocity decisions.

Second, the framework explicitly ties rollout phases to clinical service level agreement metrics. It considers ward acuity levels when scheduling device groups. It respects clinical workflow patterns to avoid disrupting critical procedures. It prevents updates during time-sensitive clinical activities like surgery. This clinical awareness distinguishes the framework from generic consumer update mechanisms.

Third, the system enforces Android Management API windowing and freeze period policies automatically. These technical controls prevent updates from occurring at clinically inappropriate times. They respect organizational maintenance schedules set by clinical engineering teams. Policy enforcement happens automatically without requiring manual intervention.

2.3 Methodology

2.3.1 Patch Pipeline Architecture

The patch generation pipeline begins immediately upon build completion. A new Android system image is generated through the standard build process. A comprehensive software bill of materials is created for all components automatically. The new image is compared systematically to the previous stable release. A differential patch is computed using bsdiff-class algorithms optimized for binary efficiency. The resulting patch undergoes cryptographic signing using the organization's private keys.

2.3.2 Device Agent Functionality

The device agent software runs on each individual fleet member continuously. It performs comprehensive pre-flight checks before permitting installation to proceed. It verifies adequate available storage space for update staging. It confirms a sufficient battery level to complete installation safely. It checks network connectivity and bandwidth availability. It validates cryptographic signatures against trusted certificate authorities.

2.3.3 Post-Flight Health Verification

Post-flight health probes execute immediately after successful boot completion. These probes are highly application-specific and context-dependent. For a patient monitoring gateway device, probes actively check Bluetooth Low Energy peripheral connectivity. They measure actual sensor data latency against defined service level agreements. They verify the complete data pipeline function from sensor to cloud. Probe results are reported immediately to the central orchestration service.

2.4 Comparative Insight

Compared to single-shot over-the-air deployments, this framework offers substantial risk reduction advantages. Single-shot deployments push updates to all fleet devices simultaneously. This strategy creates concentrated operational risk across the entire fleet. A defective update affects every device at once without warning. The phased framework limits blast radius effectively through controlled rollout. Defects are detected during small canary group deployments.

Regarding vulnerability exposure time windows, the framework enables significantly faster remediation. Differential patches reduce total download and installation time compared to full images. Smaller payloads complete more reliably over constrained clinical networks. Update orchestration can prioritize critical security updates over feature enhancements. Table 1 presents the essential technical components integrated within the over-the-air orchestration framework, emphasizing security mechanisms and operational capabilities that support medical device fleet management.

3. Core Discussion-Tips and Advice

3.1 Problem Context

Implementation teams struggle with safe over-the-air deployment across heterogeneous fleets. Mixed processor architecture fleets significantly complicate deployment planning. Devices have different ARM and x86 processor architectures requiring separate builds. They run substantially different Android versions with varying capabilities. Some devices support Android Verified Boot, while older models do not.

IoT security considerations have become increasingly complex in healthcare environments [5]. Mechanisms for authenticating the device must resist advanced attack. Network security protocols must protect both in-transit and stored data. Access control systems must support least-privilege principles. Update mechanisms themselves become potential attack vectors requiring protection. Healthcare organizations must address these security concerns systematically.

Regulated healthcare settings add numerous operational constraints beyond technical considerations. Documentation requirements are extensive and time-consuming to maintain. Validation processes are lengthy and resource-intensive before deployment approval. Risk aversion runs high among clinical stakeholders responsible for patient safety. Technical resources

are often quite limited in smaller healthcare organizations.

Blockchain-based industrial IoT systems demonstrate effective resource management strategies [6]. These actions make peer-to-peer transactions easier without centralized intermediaries. They provide transparent audit trails for all system operations. Resource allocation can be optimized through smart contract automation. Healthcare device fleets can adopt similar architectural patterns.

3.2 Practical Framework and Checklist

3.2.1 Software Bill of Materials Management

Maintain a comprehensive software bill of materials for every single firmware build. No release should ever proceed to deployment without complete documentation. The bill of materials must be machine-readable in standard formats. Store it alongside release artifacts in version control systems. This practice enables automated vulnerability scanning against public databases.

3.2.2 Risk Scoring Implementation

Risk-score all patch contents systematically before authorizing any release. Define a detailed scoring rubric appropriate for the environment. Document the complete scoring rationale for audit trail purposes. Use quantitative scores to determine appropriate rollout strategies. High-risk patches need significantly more conservative deployment approaches. Keep comprehensive scoring records for regulatory compliance review activities.

3.2.3 Android Management API Policy Configuration

Use Android Management API windowed update policies to enforce maintenance windows. Define clinically appropriate maintenance windows for each device group. Avoid scheduling updates during clinical peak hours and emergencies. Respect shift change times when clinical staff are busy. Honor freeze periods during regulatory inspections when changes are prohibited.

3.2.4 Automatic Rollback Configuration

Define detailed automatic rollback rules before beginning any deployment. Specify precise failure thresholds that trigger rollback automatically. Determine appropriate observation windows for each deployment phase. Rollback should occur completely without manual intervention requirements. Speed is critical when patient care might be affected by failures.

3.3 Dos and Don'ts

Do implement comprehensive pre-flight gates on every fleet device. Check all prerequisites systematically before permitting installation to proceed. Verify cryptographic signatures against trusted certificate authorities. Do not proceed with installation if any gate check fails. These automated checks prevent many common deployment problems proactively.

Do use remote attestation wherever hardware and software support exists. Remote attestation cryptographically proves device state to management systems. It independently verifies boot integrity through hardware trust anchors. It confirms successful update applications through signed reports.

Never ship any update without a complete software bill of materials. No exceptions should be made for urgent security patches. Bills of materials are now baseline regulatory expectations globally. They enable systematic vulnerability management across the device lifecycle. Table 2 outlines critical implementation requirements and security considerations for deploying over-the-air updates in healthcare environments, addressing both technical and operational aspects.

4. Industry-Specific Applications-Healthcare

4.1 Industry Context

Healthcare maintains an extremely low tolerance for any system downtime. Patient physiological monitoring cannot be interrupted under any circumstances. Critical diagnostic equipment must remain continuously available for emergencies. Clinical documentation systems must be accessible throughout care delivery. Even brief service outages measurably affect care quality and patient outcomes.

Blockchain and cloud computing integration offer promising solutions for healthcare IT infrastructure [7]. Cloud platforms provide scalable computing resources for device management. Blockchain adds immutable audit trails and distributed trust. The combination enables secure, transparent fleet orchestration. Healthcare organizations can use these technologies for compliance and security.

Healthcare faces heavy regulation as one of the most regulated industries, with multiple federal and state regulators. Many of those regulators cause duplication. HIPAA regulations address privacy and security for protected health information. The FDA manages medical device regulations, including software and firmware. State health

departments add jurisdiction-specific requirements and inspections.

Smart contract-based access control frameworks provide fine-grained permission management [8]. These frameworks automate authorization decisions based on predefined policies. They eliminate single points of failure in access control systems. They provide transparent, auditable access logs for compliance purposes. Healthcare device fleets can implement similar access control mechanisms.

4.2 Application Detail

4.2.1 Mobile Device Management Integration

The orchestration framework embeds seamlessly into existing mobile device management workflows. It does not require complete infrastructure replacement or major capital investment. Integration occurs through standard published application programming interfaces. Android Management API provides the primary policy enforcement mechanism. Orchestration services can run inside the public cloud or on-premises.

4.2.2 Clinical Context-Aware Scheduling

The policy engine schedules windowed installations based on clinical context. It consults the organizational clinical calendar for planned procedures. It systematically avoids scheduling during clinical peak hours. Different clinical areas may have substantially different maintenance windows. Emergency departments need fundamentally different scheduling than administrative areas.

4.2.3 Ward-Level Canary Deployment

Ward-level canary deployments provide effective risk mitigation for healthcare environments. Updates deploy initially to carefully selected test wards. These wards are chosen based on several important characteristics. They have strong on-site information technology support immediately available. Clinical staff learn to fix issues and follow escalation procedures.

4.3 Benefits and Challenges

There are also operational and quantitative advantages that can be gained. Faster vulnerability remediation dramatically reduces exposure time windows. Differential patches minimize bandwidth consumption in constrained clinical networks. This bandwidth reduction matters significantly in bandwidth-limited environments. Smaller update packages complete more reliably over wireless connections.

The technical challenges are real but ultimately manageable with proper planning. Legacy devices lacking Android Verified Boot create security coverage gaps. These older devices need special handling and additional controls. Documentation must explicitly note the hardware limitation. Migration plans should prioritize replacement with capable hardware.

Limited storage capacity on embedded medical devices constrains deployment approaches. Full system images may not physically fit on device storage. Differential patches help reduce storage requirements significantly. Even so, update staging requires temporary storage space. Table 3 describes the integration capabilities and deployment characteristics of the orchestration framework within healthcare IT infrastructure, highlighting clinical context awareness.

5. Societal and Regulatory Implications

5.1 Wider Impact

Safer and faster security updates reduce systemic cyber risk across critical infrastructure. Healthcare constitutes a designated critical infrastructure sector nationally. Successful breaches in healthcare directly threaten patient safety and wellbeing. They undermine public trust in healthcare institutions fundamentally.

Cloud-supported IoT deployments require careful security consideration across multiple dimensions [9]. Authentication mechanisms stand against credential theft. Authentication mechanisms stand against credential replay. All sensitive data must be encrypted using strong cryptographic protocols during its lifecycle. Access control must enforce separation of duties and least privilege. Update mechanisms must verify authenticity and integrity. Healthcare organizations must address all these security considerations systematically.

The framework demonstrates that security and operational needs are ultimately compatible. Healthcare organizations often face apparent false choices in decision-making. Speed versus safety seems like an inherent tradeoff. Security versus usability appears fundamentally opposed. The deployment framework shows these can be balanced effectively.

5.2 Responsibility and Equity

Transparent software bills of materials substantially improve accountability throughout supply chains. When software components are comprehensively documented, responsibility becomes clear. Vulnerability disclosure processes become more

effective for all stakeholders. Device manufacturers cannot hide problematic dependencies from scrutiny.

Remote attestation frameworks enable trustworthy verification of device state [10]. These frameworks use hardware security modules as trust anchors. They generate cryptographic proofs that cannot be forged by compromised software. They enable real-time verification of device integrity. Healthcare fleet management systems can leverage these attestation capabilities.

Comprehensive audit trails serve multiple important organizational purposes. They satisfy stringent regulatory compliance requirements efficiently. They enable thorough incident investigation after security events. They support continuous quality improvement initiatives. This documentation protects healthcare organizations from liability.

5.3 Policy and Regulation Impact

The framework aligns comprehensively with current FDA cybersecurity guidance. Recent cybersecurity documents explicitly expect secure-by-design principles. They require robust vulnerability management throughout the device lifecycle. They demand comprehensive lifecycle maintenance capabilities. The deployment framework satisfies these regulatory expectations directly.

Secure Software Development Framework alignment supports broader organizational compliance efforts. Many healthcare regulations reference NIST frameworks by name. Implementing framework practices provides foundational security across organizations. The deployment framework embodies these practices concretely.

5.4 Future Outlook

Standardized, clinically aware over-the-air deployment may become explicit regulatory expectations. Current FDA guidance clearly points in this direction. Future premarket guidance will likely be more technically specific. Premarket submissions may need to describe over-the-air capabilities in detail. Postmarket surveillance may verify their proper use in practice.

Security hygiene across healthcare is steadily improving through multiple initiatives. Over-the-air deployment is one important piece of this improvement. Better patching substantially reduces vulnerability exposure windows. This happens without sacrificing care quality or patient safety. The deployment framework demonstrates technical feasibility clearly. Table 4 presents essential security dimensions for cloud-supported Internet of Things deployments in healthcare, with emphasis on attestation and verification mechanisms.

6. Broader Implications

6.1 Environmental and Economic Effects

Smaller differential payloads measurably reduce network energy consumption across deployments. Data center facilities use less electricity for update transfer and distribution. Network infrastructure equipment processes substantially less total data volume. The environmental effect per individual update is admittedly modest. However, across millions of devices over multiple years, impacts accumulate significantly.

Predictable maintenance windows substantially reduce costly staffing workarounds. When updates are unpredictable, organizations schedule extra staff as contingency. Overtime costs increase unnecessarily due to uncertainty. Staff satisfaction decreases with unpredictable scheduling demands. Reliable update scheduling eliminates this operational waste.

Faster vulnerability patching directly curbs expensive data breach costs. Rapid vulnerability remediation measurably reduces breach probability. The return on security investment improves substantially with faster patching. Organizations can more easily justify security spending to leadership.

6.2 Long-term Outlook

Industry standards for over-the-air deployment, software bills of materials, and attestation are converging. Healthcare information technology is actively adopting these security practices. Other critical infrastructure sectors are carefully watching healthcare progress. The energy sector, transportation systems, and manufacturing face similar operational challenges.

Supply chain transparency continues to increase across the software industry. Software bills of materials enable this essential transparency. Component tracking improves systematically across development lifecycles. Vulnerability correlation becomes systematic rather than ad hoc. These practices are rapidly becoming baseline industry expectations.

Zero-trust security architectures benefit substantially from attestation infrastructure capabilities. Remote attestation cryptographically verifies device state for access decisions. It enables highly dynamic trust determinations based on the

current state. Access control becomes more granular and context-aware.

6.3 Call to Action

Medical device vendors should adopt policy-driven over-the-air deployment as standard practice. Device manufacturers should build these capabilities directly into products. They should not wait for explicit regulatory requirements to emerge. Proactive implementation demonstrates a genuine security commitment to customers.

Hospital information technology departments should demand over-the-air transparency from vendors. Procurement specifications should explicitly require a software bill of materials provision. They should expect differential update capabilities as baseline features. They should insist on health probe integration for clinical devices.

Professional standards bodies should formally document these emerging practices. Detailed technical specifications enable true interoperability between vendors. They reduce implementation variance across the healthcare industry. They support formal certification programs for products. Standardization greatly increases organizational adoption.

Table 1: *Core Technical Components of OTA Framework [3, 4]*

Component	Primary Function	Security Benefit
Differential Patching	Reduces payload size through binary comparison algorithms	Minimizes network exposure time during transmission
Android Verified Boot	Cryptographically verifies system integrity at boot time	Prevents execution of tampered or unauthorized firmware
Software Bill of Materials	Catalogs all software components and dependencies	Enables rapid vulnerability identification and tracking
Permissioned Blockchain	Provides decentralized trust without central authority	Creates immutable audit trails for compliance verification
Edge Computing Integration	Processes data locally at device level	Reduces latency while preserving privacy-sensitive operations

Table 2: *Implementation Checklist for Healthcare OTA Deployment [5, 6]*

Implementation Area	Key Requirement	Risk Mitigation Strategy
Authentication Mechanisms	Cryptographic verification of update sources	Protects against unauthorized firmware distribution
Access Control Systems	Enforcement of least-privilege principles	Prevents unauthorized modification of device configurations
Network Security Protocols	Protection of data during transit and storage	Safeguards protected health information from interception
Resource Management	Blockchain-based peer-to-peer transactions	Eliminates centralized intermediary vulnerabilities
Audit Trail Generation	Transparent logging of all system operations	Supports regulatory compliance and incident investigation

***Table 3**: Healthcare Application Integration Features [7, 8]*

Integration Aspect	Implementation Method	Clinical Benefit
Mobile Device Management	Standard API integration without infrastructure replacement	Seamless adoption within existing workflows
Cloud-Blockchain Hybrid	Scalable computing with immutable audit trails	Secure transparent fleet orchestration
Clinical Scheduling Engine	Consultation of organizational clinical calendar	Avoids disruption during critical procedures
Smart Contract Access Control	Automated authorization based on predefined policies	Fine-grained permission management for devices
Ward-Level Canary Deployment	Phased rollout to test wards with IT support	Minimizes patient impact from potential failures

***Table 4:** Security Considerations for Cloud-Supported Medical IoT [9, 10]*

Security Dimension	Protection Mechanism	Verification Method
Device Authentication	Hardware security modules as trust anchors	Remote attestation with cryptographic proofs
Credential Protection	Resistance against theft and replay attacks	Multi-factor authentication protocols
Data Encryption	Strong cryptographic protocols throughout lifecycle	End-to-end encryption for all transmissions
Access Control	Separation of duties and least privilege enforcement	Role-based access with continuous verification
Device State Integrity	Real-time verification of system configuration	Hardware-rooted attestation reports

7. Conclusions

Although healthcare organizations need to quickly patch security vulnerabilities, automatically patching consumer OTT devices over the air does not meet the criteria for a healthcare solution to patch medical devices. They lack clinical context awareness in deployment decisions. They cannot verify functional health after updates are complete. They do not generate comprehensive audit trails automatically. A policy-driven orchestration framework specifically designed for regulated medical device fleets addresses these critical gaps. The framework integrates differential patching to reduce bandwidth consumption and installation time. It uses cryptographic attestation to verify device integrity throughout the update process. It implements phased rollouts modulated by technical risk scoring. Automatic rollback prevents widespread service disruption when problems occur. Application-specific health probes verify clinical function rather than assuming successful installation equals successful operation. Software

bill of materials integration enables systematic vulnerability tracking across diverse fleets. Android Management API policies enforce maintenance windows and freeze periods automatically. The framework aligns with FDA cybersecurity guidance and NIST secure development practices. It generates audit-ready artifacts, including attestation logs and risk justifications. Simulated deployments and pilot implementations demonstrate reduced vulnerability exposure time. Clinical regressions remain near zero through systematic functional verification. The framework proves that security and operational requirements are compatible through thoughtful design. Healthcare information technology can adopt proven consumer security technologies when properly adapted for clinical contexts. Patient safety remains paramount throughout the deployment process. Standardization of these practices across the healthcare industry will accelerate security improvements. Medical device vendors should build these capabilities into products proactively. Healthcare organizations should demand over-the-air transparency in procurement processes. Professional standards bodies should document these practices formally to enable interoperability. Regulators should recognize and encourage adoption through updated guidance documents. Over-the-air deployment, software bills of materials, and remote attestation together create the future of medical device security, and healthcare needs to lead the way to protect patients and maintain the public's trust in digital health technologies.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

1. Pian Qi, et al., "A blockchain-based secure Internet of medical things framework for stress detection," Information Sciences, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0020025523001354
2. Jenn Gile, "SBOM Requirements for Medical Devices," Endorlabs, 2023. [Online]. Available: https://www.endorlabs.com/learn/sbom-requirements-for-medical-devices
3. Saad El Jaouhari and Eric Bouvet, "Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions," Internet of Things, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2542660522000142
4. Keke Gai, "Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks," IEEE Xplore, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8664577
5. Phillip Williams, et al., "A survey on security in the internet of things with a focus on the impact of emerging technologies," Internet of Things, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522000592
6. Haipeng Yao, et al., "Resource Trading in Blockchain-Based Industrial Internet of Things," IEEE Xplore, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8657779
7. Dinh C. Nguyen, et al., "Integration of Blockchain and Cloud of Things: Architecture, Applications and Challenges - 2020" S-Logix, 2020. [Online]. Available: https://slogix.in/blockchain-technology/integration-of-blockchain-and-cloud-of-things-architecture-applications-and-challenges/
8. Md. Rahat Hasan, et al., "Smart Contract-Based Access Control Framework for Internet of Things Devices," Computers, 2023. [Online]. Available: https://www.mdpi.com/2073-431X/12/11/240
9. Jatinder Singh, et al., "Twenty Security Considerations for Cloud-Supported Internet of Things," ResearchGate, 2016. [Online]. Available: https://www.researchgate.net/publication/280500892_Twenty_Security_Considerations_for_Cloud-Supported_Internet_of_Things
10. Kyeong Tae Kim, et al., "An IoT Device-trusted Remote Attestation Framework," ResearchGate, 2022. [Online]. Available: https://www.researchgate.net/publication/359191862_An_IoT_Device-trusted_Remote_Attestation_Framework