Trust Mechanisms in IIoT Software Authenticity: Challenges and Emerging Solutions

Nana Onumah[1]*, *Ali Kashif Bashir*[1], *Md Israfil Biswas*[2], *Muhammad Atif Ur Rehman*[1]

[1]*Manchester Metropolitan University, Manchester, United Kingdom*
nana-kwesi.a.onumah@stu.mmu.ac.uk; a.bashir@mmu.ac.uk; M.Atif.Ur.Rehman@mmu.ac.uk
[2]*University of Bedfordshire, Luton, United Kingdom*
mdisrafil.biswas@beds.ac.uk

Corresponding author:* nana-kwesi.a.onumah@stu.mmu.ac.uk

Abstract

The Industrial Internet of Things (IIoT) represents a transformative frontier in modern industrial operations, offering improved efficiency, predictive maintenance, and enhanced safety. However, ensuring software authenticity within IIoT systems remains a paramount challenge. Traditional trust mechanisms such as digital signatures, certificate authorities, checksums, hash functions, and public key infrastructure exhibit limitations when applied in an IIoT environment. Vulnerabilities emerge, particularly in security, resilience, performance, and scalability, which become accentuated given the resource-constrained nature of many IIoT devices and the massive scale of IIoT deployments. This paper provides a comprehensive analysis of the strengths and limitations of applying conventional trust mechanisms to ensure software authenticity, while also elucidating the potential of emerging technologies. This analysis offers a roadmap for future research and implementation in securing IIoT ecosystems. This article posits integrating emerging technologies, including blockchain, machine learning, and artificial intelligence, to address these gaps in the IIoT security paradigm. These technologies promise decentralised, adaptive, and predictive security measures, potentially elevating the robustness of software authenticity in IIoT environments.

Keywords software security; trust mechanisms; IIoT; cyber-threat

1 Introduction

Insecure software and hardware vulnerabilities in the Industrial Internet of Things (IIoT) expose critical infrastructure to various threat actors [1]. This results in cybersecurity incidents that disrupt system stability and sometimes lead to life-threatening situations [2]. Therefore, we need to identify solutions to verify software authenticity before the software is executed to minimise the risk of compromise [3]. This requires identifying technical means to establish a trust mechanism for verifying software authenticity before deployment and execution on IIoT application systems.

The number of IoT devices is growing rapidly as industries continue to explore its benefits. Some of the benefits, as described in the following sections, make it more practical for an industrial environment. However, the security of IoT devices must be ensured to provide overall security, safety, and resilience in the operating environment. This means we need to enhance the software quality of IoT devices by updating or patching them with improved software and firmware, without compromising security or introducing additional security risks. This requires us to trust the software component we deploy to the IoT devices. This paper contributes a normalised evaluation of five trust mechanisms using a weighted Security Core Index (SCI), an Operational Index (OI), and a combined Software Authenticity Score (SAS). We utilise these to identify IIoT-specific constraints and research directions.

- **Increased Efficiency:** IoT devices can gather, analyse and provide real-time data about the performance of machines and processes, enabling businesses to identify bottlenecks, streamline their operations, and significantly improve efficiency [4]. Automated systems can also perform tasks more quickly and accurately than human operators, resulting in increased productivity.
- **Predictive Maintenance and Reduced Downtime:** IoT devices can continuously monitor the health and performance of equipment, allowing potential problems to be identified before they cause failure. This enables maintenance to be carried out only when needed (predictive maintenance)

rather than on a regular schedule (preventive maintenance), leading to cost savings and reduced downtime [5].

- **Improved Safety:** In industries such as manufacturing, mining, or oil and gas, IoT technology can monitor environmental conditions and alert workers or shut down equipment if hazardous conditions are detected [6]. This can help prevent accidents and improve worker safety. Furthermore, remote monitoring and control capabilities enable tasks to be carried out without the need for workers to be physically present in potentially hazardous areas.

The apparent need for a robust trust mechanism in IIoT, along with the limitations of existing trust mechanisms, has motivated the research presented in this paper. Section II provides a brief overview of the current landscape. Section III builds on Section II by examining the challenges and constraints of existing trust mechanisms. Section IV highlights a promising new research direction that will investigate the potential of emerging technologies to address the challenges mentioned earlier.

2 IIoT Trust Mechanisms

The concept of trust mechanisms in digital systems has evolved significantly over the years in response to the changing needs and challenges posed by technological advancements, which we briefly outline in this section. One of the earliest forms of trust mechanisms, checksums, was devised in the 1950s for error checking in telecommunications [7]. As the digital age progressed, the need for more robust trust mechanisms led to the development of hash functions in the late 1970s [8] and the introduction of cryptographic hash functions such as MD5 and SHA-1 in the 1990s [9]. These provided a more reliable method for verifying data integrity and authenticity. With the rise of the internet and digital communications, verifying identities in a network was necessary. This led to the establishment of Public Key Infrastructure (PKI) and Certificate Authorities (CA) in the 1970s and 1980s, providing a framework for users and devices to verify each other's identities using digital certificates [10]. Digital signatures, introduced around the same time, provided a way to authenticate digital messages or documents [11]. Over time, these trust mechanisms have been adapted and improved to meet the increasingly complex requirements of modern digital systems, including the Industrial Internet of Things (IIoT). Ensuring the authenticity of software has become paramount in this context, particularly in relation to the IIoT [12]. The following sections provide an overview of the various trust mechanisms, their relevance, and application in the current IIoT environment.

2.1 Checksums

A checksum is a value calculated from a dataset and used to verify the integrity of the data [7]. It is a simple form of error detection calculated by summing the binary values in a block of data. It is typically used to verify data integrity during transmission or storage. The checksum is calculated before the data is transmitted or stored, and then calculated again when the data is used or received. Comparing the two checksum values can determine whether the data was altered during transmission or storage. If the checksums match, it is assumed that the data is intact; if they do not, it indicates that the data was altered. In an IIoT environment, checksums can ensure the integrity of software updates or data transmissions between devices.

For instance, consider a scenario where a centralised server is transmitting firmware updates to IIoT devices in a factory. To ensure the update has not been tampered with during transmission, a checksum of the firmware package can be calculated at the server before transmission. This checksum is then sent along with the firmware update. Upon receiving the update, each IIoT device calculates its checksum of the received data and compares it to the received checksum.

If the values match, the device can assume that the update was not tampered with during transmission and can proceed to install the update. If the values do not match, the device can reject the update and request a new transmission. While checksums are valuable for maintaining data integrity, they do not

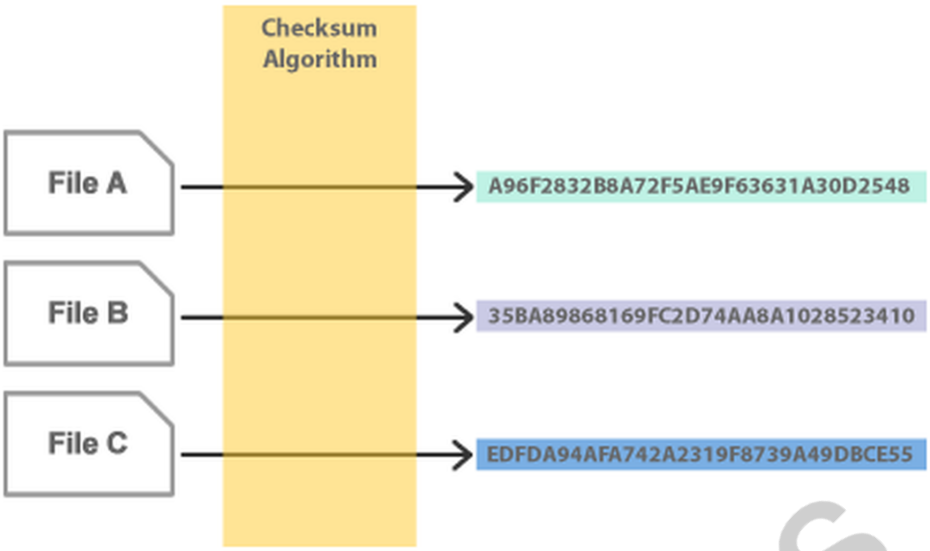

Figure 1: When the checksum algorithm is applied to the file, it generates a simple hexadecimal string - the checksum.

provide data authenticity independently of other verification methods [7]. Checksums are most effective when combined with other trust mechanisms that can authenticate the source of the data, like digital signatures or certificates from a Certificate Authority. This is because, while a checksum can indicate if data has been altered, it cannot verify if it originated from a legitimate source.

2.2 Hash Functions

A hash function is a unique function used in cryptography that takes an input (or 'message') and returns a fixed-size string of bytes [8]. The output, often referred to as the hash, is unique to each input; even a slight change in the input will produce a drastic change in the output, making the new hash appear uncorrelated with the old hash. Hash functions are "one-way", meaning it is computationally infeasible to reverse the process and derive the original input given only the hash output [8]. They are widely used in various applications, including verifying data integrity and securely storing sensitive information, such as passwords.

Hash functions play a crucial role in securing software authenticity in an IIoT environment. They can be combined with other cryptographic techniques to validate the authenticity and integrity of software updates, data transmissions, and device identities. For example, consider a scenario where IIoT devices in a factory need to download and install software updates from a central server regularly. To ensure the authenticity and integrity of these updates, the server could employ a method known as "hashing and signing". In this method, the server first calculates a hash of the software update and then uses a private key to digitally sign that hash. The software update and the digital signature are then sent to the IIoT devices. Upon receiving the software update, each IIoT device calculates the hash of the received software and uses the server's public key to verify the digital signature, which reveals the signer's hash. If the two hashes match, the device can confirm that the update is authentic and has not been tampered with during transmission [13]. It is important to note that while hash functions can ensure data integrity, they do not provide confidentiality. Hence, sensitive data should be encrypted before transmission, even when hash functions are used.

2.3 Public Key Infrastructure

PKI is a set of roles, policies, hardware, software, and procedures to create, manage, distribute, use, store, and revoke digital certificates and public-key encryption [10]. The primary purpose of PKI is to

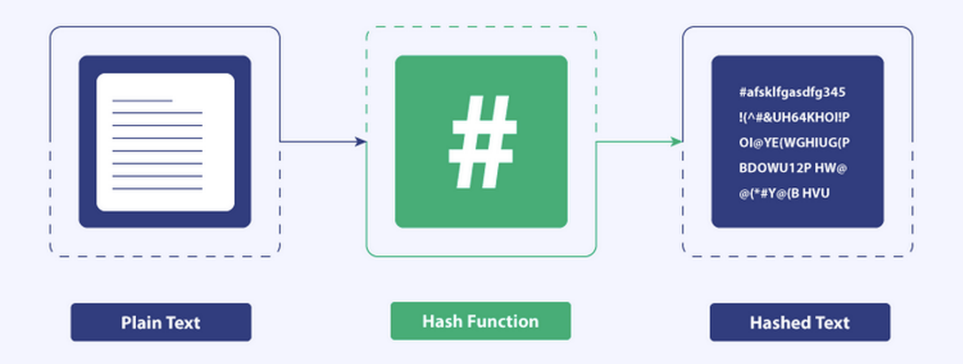

Figure 2: A Hash Function is a function that converts a given numeric or alphanumeric key to a small, practical integer value.

facilitate the secure electronic transfer of information in various activities such as e-commerce, internet banking, and confidential email. In a PKI system, a certificate authority (CA) issues digital certificates to verify the certificate holder's identity and associate that identity with a public key. This certificate can be used to authenticate the identity of a party during digital communications and for the secure exchange of encryption keys.

PKI plays a critical role in securing an IIoT environment, specifically in managing the identities of devices and ensuring the integrity and authenticity of data transmitted between them [12, 14]. For instance, consider a power grid system controlled by numerous IIoT devices that constantly communicate with each other. In this scenario, each device would possess a digital certificate issued by a trusted CA, affirming its identity and binding it to a public key. When a device needs to communicate with another, it shares its digital certificate. The receiving device can verify the sender's identity by checking the digital certificate's validity, which involves ensuring the certificate was issued by a trusted CA and has not expired or been revoked, and then confirming the CA's digital signature on the certificate. Once verified, the devices can securely exchange encryption keys for secure communication, confident in each other's identities. PKI, therefore, plays an integral role in managing the trust relationships necessary for secure communication within an IIoT environment. By providing a robust framework for device identity management and secure key exchange, PKI helps protect against various security threats, including man-in-the-middle attacks, data tampering, and device impersonation [14].

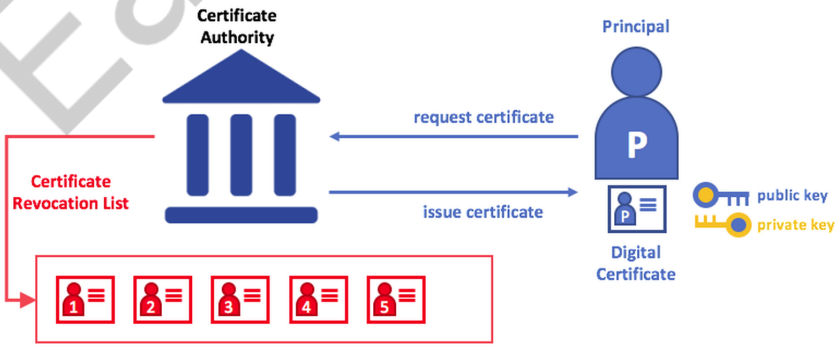

Figure 3: Public key infrastructure is the hardware, software, policies, processes, and procedures required to create, manage, distribute, use, store, and revoke digital certificates and public keys.

2.4 Certificate Authority

A CA is a trusted entity that issues digital certificates [15]. These certificates are used to certify the ownership of a public key in the context of a PKI. In other words, the CA vouches for the certificate holder's identity and binds that identity to a public key. A digital certificate typically contains information such as the owner's name, the certificate's public key, the certificate's expiration date, the certificate's serial number, and the digital signature of the CA. The digital signature of the CA allows the receiver of the certificate to verify its authenticity [16].

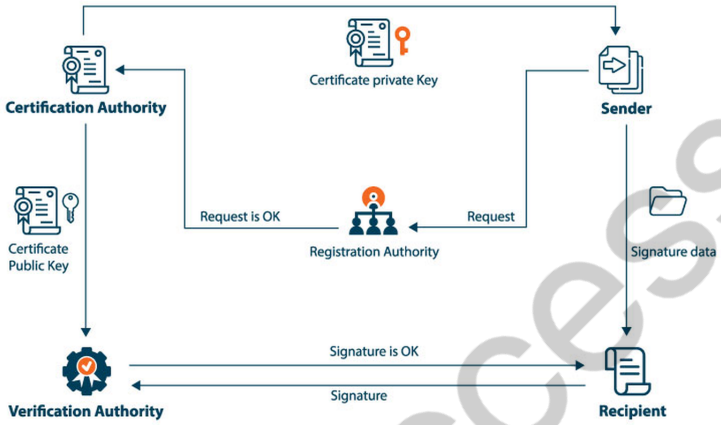

Figure 4: A certificate authority or certification authority is an entity that stores, signs, and issues digital certificates.

In an IIoT environment, a CA can be used to ensure the authenticity of the devices and the data they produce [15]. Consider an IIoT system used in a smart factory, where hundreds of sensors and actuators are connected to collect data and control machinery. Each device in this environment could be issued a digital certificate by a trusted CA. This digital certificate would authenticate and associate the device's identity with a public key. When a device sends data to a server or another device, it can digitally sign it using its private key, including its digital certificate. The receiver can then use the public key in the sender's certificate to verify the digital signature, thus confirming the authenticity and integrity of the data [11]. However, to trust the certificate, the receiver must trust the CA that issued it. The receiver can verify the CA's digital signature on the certificate to ensure it is valid. This setup can protect the IIoT environment from various threats, such as man-in-the-middle attacks, where an attacker may attempt to impersonate a device or inject false data into the system. By using digital certificates from a trusted CA, the devices can authenticate each other and ensure the integrity and authenticity of their communications.

2.5 Digital Signatures

Digital signatures are a cryptographic technique used to authenticate the identity of the sender of a message or the signer of a digital document. They ensure the integrity of the transmitted data and provide non-repudiation, which means the sender cannot deny having sent the message or signed the document [11, 17]. Digital signatures work using public key cryptography. When a sender signs a document, they use their private key to generate the digital signature. The digital signature is then attached to the document. Upon receiving the document, the receiver uses the sender's public key to verify the signature. If the signature verification is successful, it confirms that the sender signed the document and that it has not been altered during transmission.

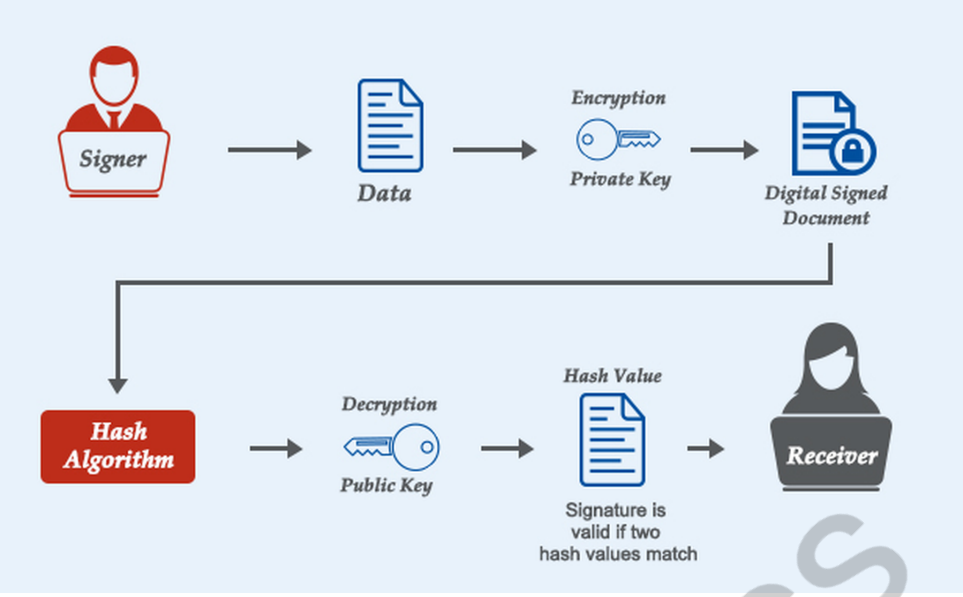

Figure 5: A digital signature is an electronic, encrypted stamp of authentication on digital information such as email messages, macros, or electronic documents.

In an IIoT environment, digital signatures can ensure the authenticity and integrity of software updates. For example, consider a manufacturing plant where various IIoT devices control different parts of the production line. These devices often require software updates to address bugs, implement security patches, or enhance functionality [18]. However, malicious actors could try to send fake software updates to these devices, causing them to malfunction or giving the actors control over them [13]. To protect against this, the developers of the IIoT devices' software can sign each software update with a digital signature before it is sent to the devices [19]. The digital signature is generated using a private key known only to the developers. When an IIoT device receives a software update, it uses the developers' public key to verify the digital signature. If the signature is valid, it confirms that the software update is authentic and has not been tampered with during transmission, allowing the device to proceed with the installation of the update. If the signature verification fails, the device rejects the update. This application of digital signatures helps to ensure the authenticity of software updates in the IIoT environment, protecting against threats such as fake updates or tampering during transmission [13, 19].

To move beyond qualitative comparisons, we normalise each mechanism's ratings and compute two summary indices: the Security Core Index (SCI) and the Operational Index (OI), as well as a headline Software Authenticity Score (SAS). Table 1 reports these values for the five mechanisms.

The weighting used for SAS is intentionally a paper-specific, security-first heuristic rather than a universal standard. Because the objective of this study is software authenticity in IIoT, the aggregate score gives greater emphasis to the Security Core Index (SCI), which captures the properties most directly tied to authenticity assurance, while still retaining the Operational Index (OI) to reflect deployment practicality in constrained environments. Accordingly, SAS is defined as $0.70 \cdot SCI + 0.30 \cdot OI$. A brief sensitivity check on the normalised scores shows that within a security-dominant range (SCI weight 0.65–0.80), the ranking remains unchanged (PKI > DS > CA > CS/HF), indicating that the comparative conclusion is not driven by a single point choice.

The results show that PKI achieves the highest SAS (0.601), narrowly ahead of Digital Signatures (DS) (0.578), reflecting its stronger performance on authenticity-centric factors (SCI). Meanwhile, Checksums (CS) and Hash Functions (HF) tie at 0.481, despite having excellent operational characteristics (OI=0.900). The Certificate Authority (CA) sits in the middle (SAS = 0.530), indicating partial support for non-repudiation, with an added operational burden from certificate management. These trade-offs mirror the narrative that PKI provides robust foundations for identity and key exchange [12, 14]. In contrast, DS secures update authenticity, but both introduce overheads that lightweight methods (CS/HF)

Table 1: Composite indices and normalised property scores (0–1) for trust mechanisms in IIoT software authenticity.

	CS	HF	PKI	CA	DS
Indices (0–1)					
Security Core Index (SCI)	0.301	0.301	**0.602**	0.501	0.569
Operational Index (OI)	**0.900**	**0.900**	0.598	0.598	0.598
Software Authenticity Score (SAS)	0.481	0.481	**0.601**	0.530	0.578
Normalized property scores (0–1)					
Confidentiality	0.00	0.00	0.33	0.00	0.00
Integrity	0.67	0.67	0.67	0.67	0.67
Authenticity	0.00	0.00	0.67	0.67	0.67
Non-repudiation	0.00	0.00	0.67	0.33	0.67
Scalability	1.00	1.00	0.33	0.33	0.33
Availability	1.00	1.00	1.00	1.00	1.00
Performance	1.00	1.00	0.33	0.33	0.33
Resilience	1.00	1.00	0.33	0.33	0.33
Usability	1.00	1.00	0.33	0.33	0.33
Interoperability	1.00	1.00	1.00	1.00	1.00
Management	1.00	1.00	0.33	0.33	0.33
Adaptability	1.00	1.00	0.33	0.33	0.33
Cost-Effectiveness	1.00	1.00	0.33	0.33	0.33
Auditability	1.00	1.00	1.00	1.00	1.00
Life-Cycle Security	0.00	0.00	1.00	1.00	1.00

Methods note: Categorical ratings were normalized as no=0.00, part=0.33, yes=0.67, full=1.00; the Security Core Index (SCI) is 0.30·Integrity +0.30·Authenticity +0.20·Non-repudiation +0.10·Confidentiality +0.10·Resilience; the Operational Index (OI) is the mean of Scalability, Availability, Performance, Usability, Interoperability, Management, Adaptability, Cost-Effectiveness, Auditability, Life-Cycle Security; SAS combines them as 0.70·SCI +0.30·OI.

avoid at the cost of assurance.

3 Existing Trust Schemes: Challenges & Constraints

The composite indices make the central tension explicit: authenticity-first approaches (PKI/DS) score higher on the Security Core Index (SCI) but incur computational and management costs, whereas lightweight approaches (CS/HF) excel in Operational Index (OI) yet cannot guarantee authenticity or non-repudiation, motivating the security, performance, scalability, and resilience constraints discussed next.

To address limitations in current trust schemes for the IIoT environment, we must first identify the challenges and constraints. This will pave the way for crafting solutions, which we will explore in this section.

> **RQ1:** *How can the security in trust mechanisms be improved to prevent compromise and enhance overall system security in IIoT?*

3.1 Security Constraints

Security is the cornerstone of any trust mechanism. Digital signatures offer high security by encrypting the signature with a private key, ensuring that only the entity with the corresponding public key can verify it. While digital signatures provide authenticity for software, they can introduce performance overhead that impacts the overall trust verification schemes in an IIoT environment [20]. Certificate authorities are responsible for issuing digital certificates to verify the identity of an entity. However, the entire system's security can be imperilled if the CA itself is compromised, leading to the issuance of fraudulent certificates. This represents a central point of failure in the system, which is a significant security concern [21]. Checksums, while efficient for error checking, are not designed to withstand deliberate tampering.

Hash functions provide a one-way transformation of input data into a fixed-size string; at the same time, they can offer strong security against data tampering, but they cannot protect against replay attacks without additional measures [22]. By contrast, PKI provides a robust framework for ensuring the authenticity and integrity of software, but relies on the security of the private keys used.

> **RQ2:** *How can the computational efficiency of trust mechanisms be improved, particularly in resource-constrained IIoT environments, without compromising security and authenticity?*

3.2 Performance Constraints

This refers to the computational resources required for the mechanism. Digital signatures and PKI offer high security, albeit at the expense of computational intensity. This may be particularly relevant to resource-constrained IIoT environments, where lightweight processes are desirable. CAs also require significant computational resources, as the digital certificates they issue and manage are computationally expensive. Checksums and hash functions are less resource-intensive and thus perform well in this aspect. However, their lower computational overhead comes at the expense of reduced security.

> **RQ3:** *What scalable strategies can be developed to manage and update trust mechanisms in large-scale IIoT environments, considering the increased complexity and potential points of failure?*

3.3 Scalability Constraints

Scalability refers to a mechanism's ability to handle growth in the number of devices or data volume. PKI can be difficult to manage as the number of devices increases, especially if each device requires unique keys [21]. The administration of digital certificates by a CA can also become increasingly complex as the number of entities in the network grows. Checksums and hash functions scale well with increasing data volumes, but do not directly address device authentication in a growing network. Digital signatures also scale well in terms of data volume, but managing an increasing number of keys can present challenges.

> **RQ4:** *What measures can be implemented to increase resilience to reduce the potential for cascading failures in case of a compromise?*

3.4 Resilience Constraints

Resilience is another critical consideration in the face of compromise or failure. Digital signatures and PKI are designed to guarantee authenticity, but their strength is contingent on safeguarding private keys. A breach of these keys can lead to widespread system failure. For instance, if an attacker gains access to a private key, they could sign malicious software, making it appear genuine. CAs represent a central point of failure, as their compromise could lead to the issuance of fraudulent certificates. Checksums and hash functions, while resilient against accidental corruption, lack the robustness to withstand targeted attacks designed to produce a matching checksum or hash.

4 Discussion

In the IIoT era, where critical systems and infrastructures are increasingly connected, trust mechanisms are integral in verifying software authenticity and maintaining overall cybersecurity [23]. Despite the value of existing mechanisms such as digital signatures, CAs, checksums, hash functions, and PKI, their inherent limitations have left significant security gaps unaddressed, calling for new strategies in our quest for a robust, secure, and efficient IIoT environment.

Either individually or collectively, current trust mechanisms struggle to cope with the unique and evolving challenges of IIoT. These include the massive scale, the heterogeneous and dynamic nature of devices, the criticality of services they facilitate, and the limited resources available on the

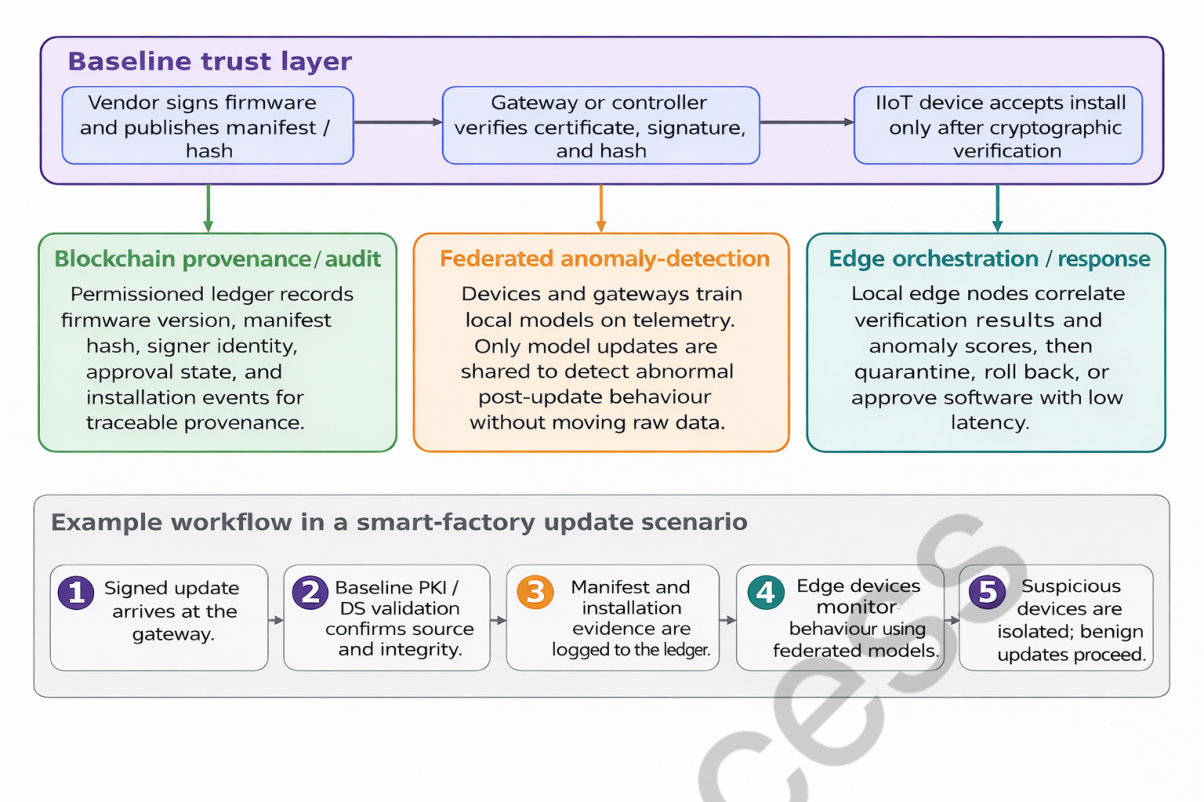

Figure 6: Layered augmentation of existing IIoT software trust mechanisms with blockchain provenance, federated anomaly detection, and edge-based response.

devices themselves. While pivotal in maintaining cybersecurity, existing mechanisms often fall short in their capacity to provide robust security, ensure data integrity, and support system availability without compromising performance, particularly on resource-constrained devices. Furthermore, these mechanisms may lack resilience against advanced cyber threats, limiting their ability to adapt swiftly and efficiently to an ever-changing threat landscape.

Moreover, as IIoT networks become increasingly complex, traditional trust management strategies are proving inadequate. Current mechanisms may struggle to deliver resource-efficient, resilient, and scalable solutions that ensure software authenticity in a decentralised yet interoperable manner [24]. The urgent need for advanced, automated trust management and real-time threat response solutions has become more apparent. Given these limitations, a growing consensus is that a new cybersecurity approach is essential for IIoT [1]. Such an approach should address the challenges mentioned earlier and anticipate future developments in the rapidly evolving field of IIoT. It should provide robust security, resilience to attacks, efficient performance even on constrained devices, scalability to accommodate growing numbers of devices and adaptability to changes in the threat landscape and operational environment.

In this paper, emerging technologies such as machine learning, blockchain, and edge computing are best interpreted as layered enhancements to existing trust mechanisms rather than as direct replacements for PKI, CAs, hashes, and digital signatures. The baseline cryptographic controls still provide the root of trust, identity binding, and software-signing assurance [25, 26]. The emerging layers extend provenance, runtime visibility, automation, and response in areas where conventional mechanisms alone become difficult to scale [27, 28].

A practical IIoT example is a smart-factory software update workflow. In such a deployment, the vendor still signs the firmware image and publishes its manifest and hash using the existing PKI/DS process. A gateway or plant controller verifies the certificate chain, signature, and hash before the update is released to edge devices. Blockchain can then be used as a permissioned provenance layer to record firmware versions, signer identity, approval state, and installation events, creating an immutable audit

trail for software lineage and rollback decisions [25, 26]. In parallel, federated learning can be used to train anomaly-detection models across gateways or production cells so that post-update deviations in software behaviour are detected without exporting raw operational telemetry from each site [29, 30]. Edge orchestration closes the loop by correlating the verification result with the anomaly score and then deciding whether the software should be approved, quarantined, or rolled back locally with minimal latency [25, 28]. Figure 6 illustrates this layered augmentation workflow.

This layered model is particularly relevant in resource-constrained IIoT deployments because it preserves lightweight endpoint verification while shifting heavier analytics and coordination functions to gateways or edge nodes. Recent work shows that federated and privacy-preserving approaches can improve anomaly detection without centralising sensitive data [29, 30]. Blockchain-assisted architectures can strengthen provenance, auditability, and distributed trust management across heterogeneous IoT environments [25, 26].

Recent advances also indicate two important directions for future IIoT software trust. First, post-quantum cryptography is becoming relevant for long-lived IIoT deployments because device identities, stored firmware artefacts, and software-signing workflows may require assurance against future quantum-capable adversaries. The release of NIST's ML-KEM and ML-DSA standards makes hybrid migration planning for key establishment and digital signatures a realistic consideration for next-generation IIoT trust mechanisms [31, 32]. Second, homomorphic encryption offers a path to privacy-preserving anomaly detection and collaborative analytics on encrypted IIoT telemetry, although current computational overhead means that such methods are more practical at the gateway or edge tier than on highly constrained endpoints [29]. These developments do not invalidate current PKI/DS approaches; rather, they show how future IIoT trust architectures can evolve by combining strong cryptographic roots of trust with adaptive and privacy-preserving security services.

However, these potentials cannot be fully realised without substantial research and development efforts. Therefore, the focus should be on fostering innovation and continuous learning to avoid potential threats and enable a more secure future for IIoT. Such endeavours could lead to the development of novel trust mechanisms that leverage these cutting-edge technologies, outperforming their predecessors and setting new standards for IIoT security.

5 Conclusion

As we delve into IIoT, ensuring software authenticity becomes critical to operational integrity. Five widely used trust mechanisms discussed in Section II above play a substantial role in securing software authenticity. However, a deep dive analysis of these mechanisms in Section III above reveals intrinsic limitations, mainly when implemented within the unique context of IIoT environments. As the need to secure IIoT environments intensifies, we must look beyond traditional trust mechanisms. This necessity propels us into emerging technologies such as blockchain, machine learning, and artificial intelligence. Blockchain's decentralised nature, tamper-proof records, and peer-to-peer trust model make it a promising candidate for IIoT security. It can potentially address scalability and single-point-of-failure issues by providing a distributed trust framework. Similarly, ML and AI can help develop predictive and adaptive security systems that learn from patterns and behaviours to identify anomalies, thus enhancing the resilience and performance of IIoT security. These technologies can also help efficiently manage IIoT devices for large-scale IIoT applications.

While necessary, existing trust mechanisms may not be sufficient for the complex needs of IIoT environments. Emerging technologies promise a new frontier in IIoT security, paving the way for a future where software authenticity is robustly ensured. The evolution and adoption of these new approaches is an area ripe for exploration, necessitating further research and experimentation. Future investigations should also assess how post-quantum cryptography and privacy-preserving techniques such as homomorphic encryption can be incorporated into IIoT trust workflows without overwhelming constrained devices.

References

[1] M. A. Khan and K. Salah. "IoT security: Review, blockchain solutions, and open challenges". In: *Future Generation Computer Systems* 82 (2017), pp. 395–411.

[2] S. H. Mekala et al. "Cybersecurity for industrial IoT (IIoT): Threats, countermeasures, challenges and future directions". In: *Computer Communications* (2023).

[3] W. Z. Khan et al. "Industrial internet of things: Recent advances, enabling technologies and open challenges". In: *Computers & Electrical Engineering* 81 (2020), p. 106522.

[4] L. Fetahu, A. Maraj, and A. Havolli. "Internet of things (IoT) benefits, future perspective, and implementation challenges". In: *45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, 2022, pp. 399–404.

[5] S. Ayvaz and K. Alpay. "Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time". In: *Expert Systems with Applications* 173 (2021), p. 114598.

[6] S. Misra et al. "Industrial internet of things for safety management applications: A survey". In: *IEEE Access* 10 (2022), pp. 83415–83439.

[7] A. Meylan et al. "A study on the use of checksums for integrity verification of web downloads". In: *ACM Transactions on Privacy and Security (TOPS)* 24.1 (2020), pp. 1–36.

[8] H. Ahmed. "A review of hash function types and their applications". In: *Wasit Journal of Computer and Mathematics Science* 1.3 (2022), pp. 120–139.

[9] S. Ghoshal et al. "A journey from MD5 to SHA-3". In: *Trends in Communication, Cloud, and Big Data: Proceedings of 3rd National Conference on CCB, 2018*. Springer, 2020, pp. 107–112.

[10] O. Albogami et al. "Public key infrastructure traditional and modern implementation". In: *International Journal of Network Security* 23.2 (2021), pp. 343–350.

[11] S. Aggarwal and N. Kumar. "Digital signatures". In: *Advances in Computers*. Vol. 121. Elsevier, 2021, pp. 95–107.

[12] J. Astorga et al. "Revisiting the feasibility of public key cryptography in light of IIoT communications". In: *Sensors* 22.7 (2022), p. 2561.

[13] S. Paul et al. "A cryptographic method for defense against MITM cyber attack in the electricity grid supply chain". In: *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2022, pp. 1–5.

[14] J. Hoglund et al. "PKI4IoT: Towards public key infrastructure for the internet of things". In: *Computers & Security* 89 (2020), p. 101658.

[15] C. Boudagdigue et al. "Trust-based certificate management for industrial IoT networks". In: *IEEE Internet of Things Journal* (2023).

[16] A. Atutxa et al. "Improving efficiency and security of IIoT communications using in-network validation of server certificate". In: *Computers in Industry* 144 (2023), p. 103802.

[17] M. R. Alagheband and A. Mashatan. "Advanced digital signatures for preserving privacy and trust management in hierarchical heterogeneous IoT: Taxonomy, capabilities, and objectives". In: *Internet of Things* 18 (2022), p. 100492.

[18] G. Banegas et al. "Quantum-resistant software update security on low-power networked embedded devices". In: *International Conference on Applied Cryptography and Network Security*. Springer, 2022, pp. 872–891.

[19] A. Ghosal, S. Halder, and M. Conti. "Secure over-the-air software update for connected vehicles". In: *Computer Networks* 218 (2022), p. 109394.

[20] L. Leonardi et al. “On the hardware–software integration in cryptographic accelerators for industrial IoT”. In: *Applied Sciences* 12.19 (2022), p. 9948.

[21] S. Belattaf et al. “Reliable and adaptive distributed public-key management infrastructure for the internet of things”. In: *Wireless Personal Communications* 120 (2021), pp. 113–137.

[22] J. Hajny et al. “Cryptographic protocols for confidentiality, authenticity and privacy on constrained devices”. In: *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2020, pp. 87–92.

[23] L. Chen, Z. Ye, et al. “A security, privacy and trust methodology for IIoT”. In: *Tehnicki Vjesnik* 28.3 (2021), pp. 898–906.

[24] A. Dua et al. “Trustful: A decentralized public key infrastructure and identity management system”. In: *2020 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2020, pp. 1–6.

[25] Sasikumar Asaithambi et al. “A Secure and Trustworthy Blockchain-Assisted Edge Computing Architecture for Industrial Internet of Things”. In: *Scientific Reports* 15 (2025), p. 15410.

[26] Giuseppe D’Aniello and Lidia Fotia. “Blockchain and AI-Based Methods for Trust Management in IoT: A Comprehensive Survey”. In: *Internet of Things* 34 (2025), p. 101755.

[27] Rachid Alami et al. “Blockchain Enabled Federated Learning for Detection of Malicious Internet of Things Nodes”. In: *IEEE Access* 12 (2024), pp. 188174–188185.

[28] Claudio Savaglio, Pasquale Mazzei, and Giancarlo Fortino. “Edge Intelligence for Industrial IoT: Opportunities and Limitations”. In: *Procedia Computer Science* 232 (2024), pp. 397–405.

[29] Marco Arazzi, Serena Nicolazzo, and Antonino Nocera. “A Fully Privacy-Preserving Solution for Anomaly Detection in IoT using Federated Learning and Homomorphic Encryption”. In: *Information Systems Frontiers* 27 (2025), pp. 367–390.

[30] Mohammad Shahin, Ali Hosseinzadeh, and F. Frank Chen. “A Two-Stage Hybrid Federated Learning Framework for Privacy-Preserving IoT Anomaly Detection and Classification”. In: *IoT* 6.3 (2025), p. 48.

[31] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Tech. rep. FIPS 203. National Institute of Standards and Technology, 2024.

[32] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. Tech. rep. FIPS 204. National Institute of Standards and Technology, 2024.