RapidAtt: A Fast Attestation Technique for Industrial Internet of Things

Syed Owais Athar[†‡], *Student Member IEEE*, Muhammad Naveed Aman[†], *Senior Member IEEE* and Biplab Sikdar[§], *Fellow IEEE*

[†] *School of Computing, University of Nebraska-Lincoln, Lincoln, NE USA.*

[‡] *Balochistan University of IT, Engineering and Management Sciences, Quetta, Pakistan.*

[§] *Department of Electrical and Computer Engineering, National University of Singapore, Singapore.*

***Abstract*—The industrial Internet of things (IIoT) relies on programmable logic controllers (PLCs) for critical operations, therefore making them prime targets for cyber-attacks especially when the program is manipulated with malevolent intent. Current attestation methods either need ongoing monitoring of the PLC program during runtime which results in substantial computational burden, or rely on physical models that are challenging to accurately develop and maintain with precision. This paper introduces a novel and efficient attestation method exclusively developed for IIoT settings, which effectively combines efficiency and security, particularly in legacy PLCs that may not have sufficient computing capabilities. Contrary to continuous attestation, this approach conducts periodic attestation at intervals and selectively validates different parts of the PLC program randomly against the legitimate PLC program. Implementing this focused strategy decreases the computational load while ensuring a strong probability of detecting unauthorized modifications. Experimental verification demonstrates that our approach achieves a total verification time of 11.93 ms (i.e. improving execution time by up to 17.67% over existing techniques), and maintains detection accuracy above 90%, thereby offering superior efficiency and security for both contemporary and older Programmable Logic Controllers in industrial environments.**

***Index Terms*—Industrial Internet of things (IIoT), Remote Attestation, Programmable Logic Controller (PLC), Algorithm, Cyber-physical Systems, Hardware Security.**

I. Introduction

The industrial Internet of things (IIoT) is revolutionizing industrial processing through the integration of smart devices, sensors, and industrial controllers that communicate and coordinate to enhance productivity and operational efficiency. Among these, programmable logic controllers (PLCs) are essential for managing and automating most of the processes in manufacturing, energy, and other critical infrastructure sectors. Maintaining the integrity of the PLC program is critical for ensuring the security and operational dependability of IIoT systems [1], [2].

PLCs are highly vulnerable to cyberattacks especially when their controlling programs are compromised [1]. High-profile cases such as Stuxnet [3] that targeted Iranian nuclear facilities, Triton [4] that disrupted safety systems at a petrochemical plant, and BlackEnergy [5] that significantly disrupted Ukraine's power grid. These cyberattacks reveal the severe threat to critical infrastructure, while underscoring the urgent need for robust PLC security and advanced security protocols in IIoT environments.

Unauthorized changes to PLC programs can interrupt operations, create safety issues, and lead to financial losses. The existing work on verifying PLC code integrity, a process called attestation, is not only limited but also has key drawbacks [6]. Attestation is a security process that verifies software integrity (such as a PLC program) by comparing it to a trusted reference, where a trusted component that performs periodic check to detect tampering is known as attester [7]. Techniques that require continuous monitoring use too much processing power and are not feasible for legacy PLCs [8]. On the other hand, model-based methods require detailed system knowledge and are hard to apply across different environments. Other approaches rely on fixed schedules, making it easier for attackers to avoid checks. Most importantly these techniques rarely allow flexible attestation schedules suitable for real-time industrial systems [9].

These limitations highlight the need for an attestation technique that is not only lightweight and unpredictable in its checking mechanism but also flexible enough to accommodate legacy devices without sacrificing performance. Approaches based on selective or probabilistic attestation have been proposed to address computational overhead, but they are either unreliable or too application specific. Furthermore, their compatibility with low-power or outdated hardware remains limited [10].

To address these challenges this paper presents "RapidAtt", a novel attestation method designed specifically for IIoT environments. RapidAtt performs periodic and randomized validation of selected sections of the PLC program in regular intervals. This approach significantly reduces the system's processing load while preserving security. RapidAtt achieves a detection rate exceeding 90%, even against partial or intermittent code modifications. The proposed technique is compatible with both modern and legacy PLCs and supports configurable attestation intervals based on system constraints and threat models.

The paper is structured as follows: Section II discusses related work. Section III outlines the system and threat models. Section IV introduces the proposed RapidAtt technique. Security analysis is provided in Section V. Section VI describes the

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002).

implementation, followed by results and discussion in Section VII. Finally, Section VIII concludes the paper and outlines future directions.

II. Related Work

PLCDefender [11] proposes a hybrid attestation method that integrates a physical model to verify PLC behavior. Despite accurate modeling, its complexity and computational overhead make it challenging for legacy PLCs.

PAtt [12] combines remote software attestation and process validation by encoding memory state integrity into sensor readings. It effectively detects logic manipulation but requires detailed physical process modeling, limiting its scalability.

Chen et al. in [13] introduced a neural-network-based black-box model for verifying code integrity while preserving privacy. While the technique accurately detects code alterations, its reliance on machine learning introduces computational overhead and potential issues with model generalization.

Likely, [14] is a purely software-based remote attestation (SBRA) technique emphasizing lightweight verification suitable for resource-constrained IoT devices. Although SBRA effectively detects software integrity violations it lacks protection against physical tampering and real-time operational disruptions in industrial environments.

In contrast to the above mentioned techiques, RapidAtt emphasizes efficient and randomized periodic attestation that balances security and computational efficiency without the need for complex physical modeling or machine learning methods. A comparison summary of existing techniques, including RapidAtt is provided in Table I.

III. System and Threat Model

A. System Model

The system model under consideration consists of a typical IIoT environment where various industrial devices, including PLCs, are interconnected to perform critical operations. The system model is shown in Figure 1, where the PLCs serve as the central control units that execute predefined programs to manage industrial processes. The integrity of these PLC programs is crucial to ensure the reliability and security of the entire system.

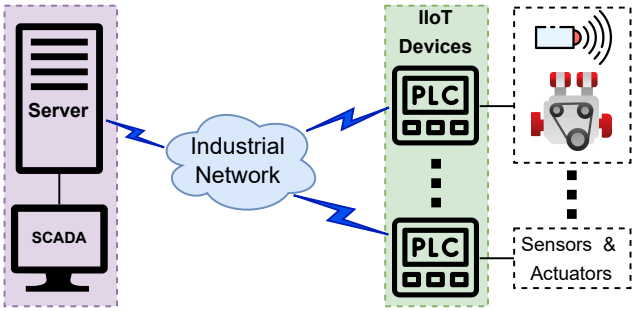

Fig. 1. The system model diagram depicting individual PLCs as IIoT devices within an industrial network that are managed and monitored through a SCADA system, connected to sensors, actuators, and field devices, where a periodic attestation mechanism verifies the PLC program's integrity against a secure and trusted baseline for robust industrial security.

B. Threat Model

The threat model explains how an attacker might tamper with PLC code to compromise the system integrity. These actions may target operations, safety, or data. The attacker may

- access and modify the PLC program to introduce malicious code.
- has knowledge of the system operation and may attempt to evade detection by making subtle changes.
- possesses the ability to introduce modifications at any point in time during the PLC scan cycle making it challenging to detect tampering using conventional methods.
- launch direct physical attacks on the PLC and IIoT components.
- take advantage of any flaws in the firmware, software, or communication protocols.
- Despite knowing the system's operation the adversary cannot predict the exact sections of the PLC program that will be verified during each attestation cycle.

In order to cover a broader range of possible attacks that can be performed the treat model considers both internal and external threats with a variety of expertise, knowledge, and resources [15]. The adversary aims to evade detection by remaining stealthy and introducing minimal or time-triggered changes that are difficult to detect using conventional attestation methods. The attacker may attempt to avoid being discovered by using evasion techniques such as modifying program outside attested regions, timing attacks between verification cycles, or relocating the malicious payload across different sections of the PLC program.

C. Assumptions

- It is assumed that the legitimate PLC program (golden reference) is securely stored and cannot be altered by the adversary.
- PLCs have limited computation capabilities.
- The next PLC section to be verified is randomly selected.

IV. Proposed Attestation Technique

A. Overview of RapidAtt

An overview of RapidAtt and its associated sections is illustrated in Figure 2. RapidAtt maintains a high probability of detecting unauthorized modifications while remaining computationally efficient. This is done by implementing a randomized periodic attestation mechanism. The attester attests at an interval of τ seconds against a Legitimate PLC program, where the attestation interval τ depends on the required attestation frequency and the security level. Algorithm 1 presents the attestation logic, while Figure 3 illustrates its operational flow.

B. Periodic Attestation

The proposed method periodically verifies random segments of the PLC program every τ seconds. The randomness ensures unpredictability, making it difficult for attackers to

TABLE I
COMPARISON OF ATTESTATION TECHNIQUES

Technique	Attestation Strategy	Computational Complexity	Hardware Dependence	Supports Legacy PLCs
PLCDefender [11]	Model-based verification	$O(d^3)$	Yes	No
PAtt [12]	Sensor-data-based validation	$O(s^2)$	Partial	Limited
NN Prediction [13]	Neural network-based prediction	$O(s \log s + m)$	No	No
SBRA [14]	Software-based memory checking	$O(s \log s)$	No	Partial
RapidAtt [Proposed]	Randomized periodic checking	$O(k)$	No	Yes

Note: d is the dimensionality, s is the input data size, m is the number of important inputs. k is the number of sections of the PLC program in the proposed technique. Also, $k \ll d$, $k \ll s$.

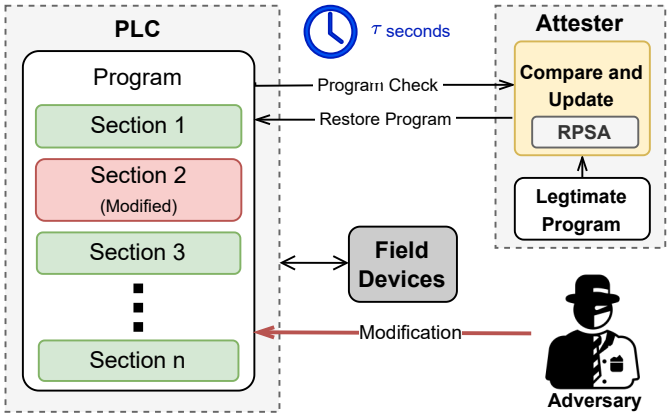

Fig. 2. Overview of RapidAtt.

Algorithm 1: Proposed Algorithm for PLC Program Validation

```
Procedure RapidAttestation
    PLC_program, golden reference R
while True do
    Wait for τ seconds;
    random_sections ← SelectRandomSections();
    foreach section ∈ random_sections do
        section_data ←
          ReadSectionData(PLC_program, section);
        if section_data ≠ R[section] then
            raise_flag ← True;
            send_alert(section, raise_flag);
            RestorePLCProgram(section, baseline);
            return CompromiseDetected();
        end
    end
end
```

anticipate checks. The default interval of 10 seconds balances system load with security but can be adjusted to meet specific security or resource constraints requirements.

C. Random Program Section Attestation (RPSA)

RapidAtt randomly selects small portions of the PLC program, typically a few hundred instructions per check. These segments are compared to a securely stored golden reference. Detected discrepancies trigger alerts and signal potential unauthorized changes. This random polling of program sections reduces reduces an attacker's chances of evading detection, even with subtle modifications.

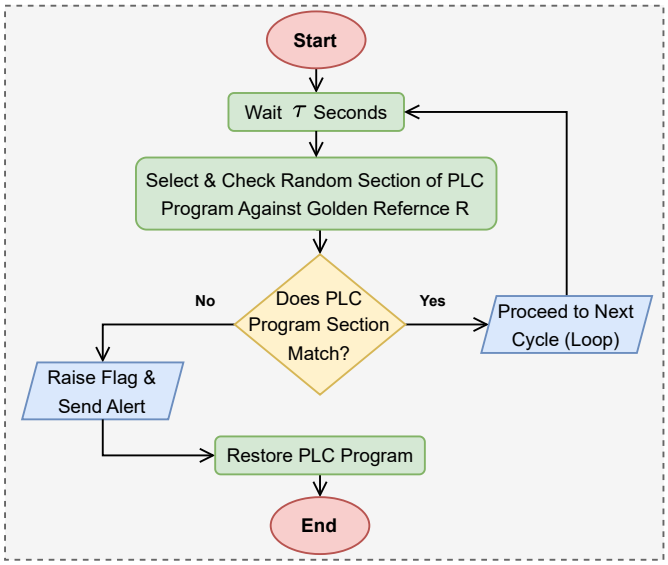

Fig. 3. Flowchart of the RapidAtt: fast attestation algorithm for PLC program validation.

D. Attestation Frequency

The attestation frequency is critical to achieve a balance between resource efficiency and the required security level [16]. RapidAtt uses a 10-second default interval, providing timely detection with low computational overhead. Depending on specific operational needs, this attestation interval can be shortened for enhanced security or extended to conserve power and computations resources.

E. PLC Program Comparison

Each attestation cycle involves comparing selected PLC program sections against a trusted and immutable golden reference. Any mismatch identifies possible tampering that leads to prompting immediate action. This focused approach reduces computational requirements while ensuring efficient operation suitable for both modern and legacy PLC environments.

F. Adaptability to Legacy PLCs

RapidAtt effectively supports legacy PLCs (which typically lack resources for intensive security protocols). Its design ensures minimal processing overhead by verifying small code segments per cycle. This makes RapidAtt suitable for industrial setups combining older and newer PLCs, and improving the overall system security without impacting performance.

V. Security Analysis

In this section, we present a formal security analysis of the proposed technique. We focus on analyzing the detection probability and the resilience of the method against potential attacks. To support this analysis, we define a "section" as a logical subset of the PLC program selected during each attestation cycle. Each section comprises multiple fixed-size "segments," which are the units used to calculate detection probability based on bit-level verification.

In the IIoT environment, the adversary seeks to remain unnoticed while existing within the PLC program. By randomly selecting and attesting segments within different sections, RapidAtt ensures ongoing program integrity verification while complicating the attacker's evasion strategy. The goal of the attacker is to adjust its location in order to evade detection during the attestation procedure. An essential metric for evaluating the effectiveness of the attestation mechanism is the likelihood of identifying such an opponent.

Theorem 1: The optimal approach for the adversary is to move itself once every software segment has been verified. The likelihood of discovery in this scenario is determined by:

$$P_D = 1 - e^{-1} \tag{1}$$

Proof: Consider that the PLC program is divided into n sections. To complete one iteration of the attestation process, n time intervals are required, denoted as $t_1, t_2, ..., t_n$. During each time interval t_i, a specific section p_i is randomly selected and attested. Let M_{ϕ_i} represent the section containing the adversary at time t_i. If the adversary evades detection at time t_i, it can either stay in the current section or move to another section during t_{i+1}. The probability that the adversary is detected when moving to a new section after t_i is:

$$P_{\text{move},\zeta_i} = \frac{1}{n}\left(1 - \left(\frac{1}{2}\right)^{\gamma}\right) \tag{2}$$

where $\gamma = m\beta$, m and β are the number of segments per section and the number of bits verified per segment, respectively. Let

$$\Upsilon = 1 - \left(\frac{1}{2}\right)^{\gamma} \tag{3}$$

Then, equation 2 can be rewritten as:

$$P_{\text{move},\zeta_i} = \frac{\Upsilon}{n} \tag{4}$$

Similarly, the probability of detection if the adversary decides to stay in its current section after t_i is:

$$P_{\text{stay},\zeta_i} = \frac{\Upsilon(n-j)}{n(n-i)} \tag{5}$$

where j denotes the last interval when the adversary moved. If $j < i$, then $P_{stay,i} > P_{move,i}$. Hence, the adversary's optimal strategy to evade detection is to move after every interval. The overall probability of evading detection is:

$$P_{\text{Adv}} = \left(1 - \frac{\Upsilon}{n}\right)^{n-1} \times \left(1 - \frac{1}{n}\right) \approx e^{-1} \tag{6}$$

where the probability that the adversary is not detected in the first section is $\Pr[M_{\phi_1} \neq \rho_1]$. The probability of detection is then:

$$P_D = 1 - P_{\text{Adv}} = 1 - e^{-1} \tag{7}$$

To enhance the likelihood of detection, several iterations might be executed. The probability of detection rises in proportion to the number of iterations and the number of bits in each segment. Figure 4 illustrates this trend by highlighting RapidAtt's increasing effectiveness with repeated checks. This demonstrates the technique's robustness against adaptive adversaries and affirms its suitability for securing both modern and legacy PLCs.

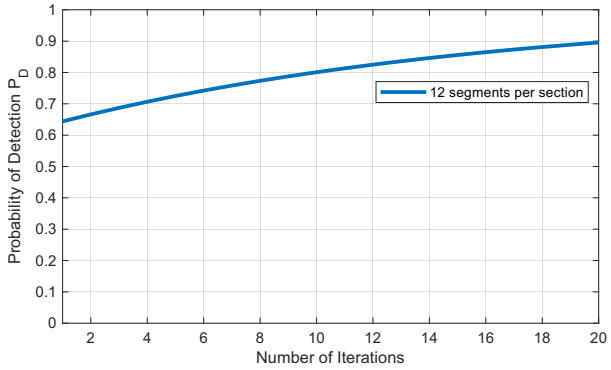

Fig. 4. Plot showing the probability of detection (P_D) versus the number of iterations for $n = 16$, $\beta = 5$, and $m = 12$ segments per section.

VI. Implementation and Experimentation

A. Experimental Setup

The implementation includes a testbed comprising of a real-time IIoT environment in which multiple PLCs initiate communication and undergo attestation using RapidAtt. Figure 5 shows all the components from the core of the setup.

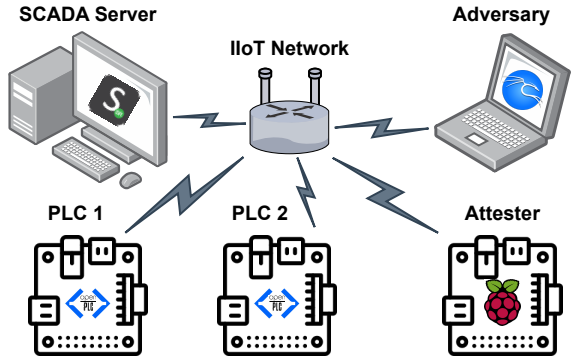

Fig. 5. Block diagram of experimental setup.

We used three Raspberry Pi devices, a WiFi router, and two computers (see Figure 6).

1) Attester: One Raspberry Pi runs the Raspbian OS and acts as the attester. It contains the authentic golden reference PLC program which is used for verifying the integrity of operational PLCs.

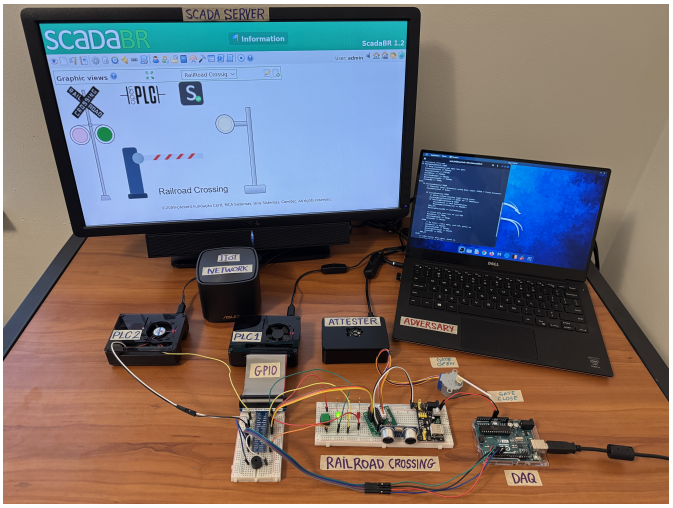

Fig. 6. Experimental Setup.

2) PLCs: The remaining two Raspberry Pi devices run the OpenPLC runtime, effectively functioning as PLCs. They manage operations for a railroad crossing scenario in a simulated IIoT network. Raspberry Pi was chosen over commercial PLCs due to its flexibility, low cost, ease of integration in testbed environments, and to avoid legal or ethical concerns associated with attacking real industrial hardware.

3) Wireless Router: The wireless Wi-Fi router establishes an industrial network that connect all devices. This allows the attester to periodically (every 10 seconds) check the integrity of the PLC program running on PLCs.

4) SCADA Server: A SCADA server running ScadaBR continuously monitors and controls these PLCs, providing real-time system feedback.

5) Adversary: A Kali Linux laptop on the same network simulates an attacker attempting to compromise PLC 1 by altering its Program.

B. Random Attestation and Adversarial Detection

Every 10 seconds, the attester randomly selects portions of the PLC program for verification against the golden reference. During the test scenario, the attacker modifies a segment of the PLC 1 program. The attester detects this anomaly in the next attestation cycle, raises an alarm, and informs the SCADA server. Immediately after detection, the attester restores the compromised segment with the authentic version to maintain system integrity.

C. Real-Time Monitoring

The SCADA server continuously monitors the status of both PLCs and observes the input-output status of all PLCs. Upon being informed of a breach by the attester the SCADA server records the incident and alerts the system administrator for necessary intervention. This configuration effectively demonstrates the resilience and effectiveness of the RapidAtt approach in realistic IIoT scenarios, ensuring the integrity and continuous functionality of PLC systems.

VII. Results and Discussion

A. Detection Effectiveness

RapidAtt consistently achieved a high detection rate for unauthorized PLC program modifications. In the experiments, an adversary tampers with PLC 1's program to simulate an attack. The attester operates at every τ seconds intervals, randomly verifying code segments to detects the modification. In our experiments, we observed a detection probability exceeding 90%, as anticipated in the design phase. We simulated a railroad crossing where PLC controls lights and gates. When the adversary altered its program RapidAtt detected the change in the next cycle, alerted the SCADA server, and restored the correct segment. The flow of events of a scenario involving this attack and subsequent response is illustrated in Figure 7.

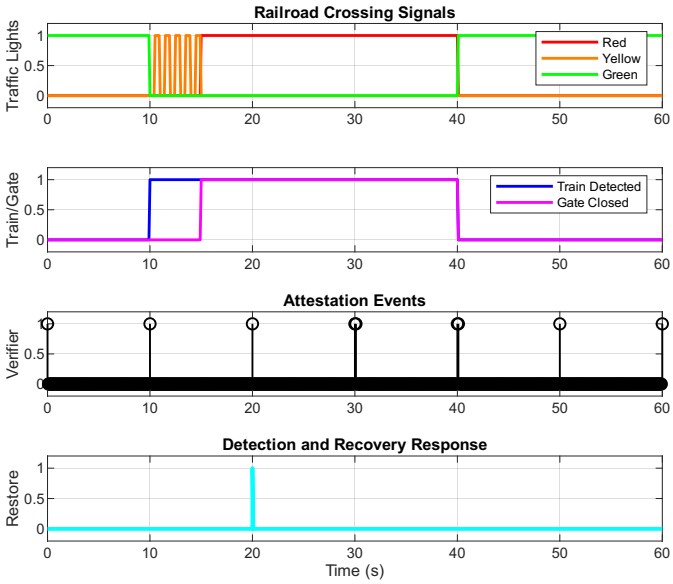

Fig. 7. Timeline plot showing railroad crossing events and the attestation process. The top subplot illustrates the traffic light transitions (green, yellow blinking, and red) in response to train detection. The second plot sho ws train proximity and gate closure status. The third subplot marks periodic verifier activations every 10 seconds. The final subplot highlights the detection and restoration event triggered by RapidAtt, demonstrating real-time recovery following tampering.

B. Latency and System Performance

RapidAtt introduces minimal latency, completing attestation cycles within sub-second intervals, thus ensuring real-time operational performance remains unaffected. To assess the computational efficiency of RapidAtt a 68 KB PLC program file was modified. This results in a total two-way transfer time of 10.88 ms over a 100 $Mbps$ Wi-Fi connection. The processing time of RapidAtt was estimated at 1.05 ms (for $P_D = 90\%$), which leads to a total verification time of 11.93 ms. Table II compares the execution times of RapidAtt with existing techniques while demonstrating notable improvements. Specifically, RapidAtt achieved total verification times faster by 6.59%, 9.76%, 11.18%, and 17.67% compared to NN-based Prediction [13], SBRA [14], PAtt [12], and PLCDefender [11], respectively. This highlights RapidAtt's suitability for resource-sensitive IIoT applications.

TABLE II
EVALUATION OF END-TO-END ATTESTATION TIMES ACROSS DIFFERENT ATTESTATION TECHNIQUES

Technique	RapidAtt	[13]	[14]	[12]	[11]
File Size (KB)	68	68	68	68	68
File Transfer Time (ms)	10.88	10.88	10.88	10.88	10.88
Processing Time (ms)	1.05	1.89	2.34	2.55	3.61
Total Time (ms)	11.93	12.77	13.22	13.43	14.49
Improvement (%)	–	6.59%	9.76%	11.18%	17.67%

C. Computational Overhead and Scalability

As illustrated in Figure 8, RapidAtt operates with a bounded computational complexity of $O(k)$, where $k \ll N$, due to its randomized and selective attestation approach that verifies only a few sections per cycle in contrast to its counterparts. This not only ensures less computational burden, but also highlights the superior scalability of RapidAtt, making it suitable for resource constrained and time-sensitive IIoT deployment.

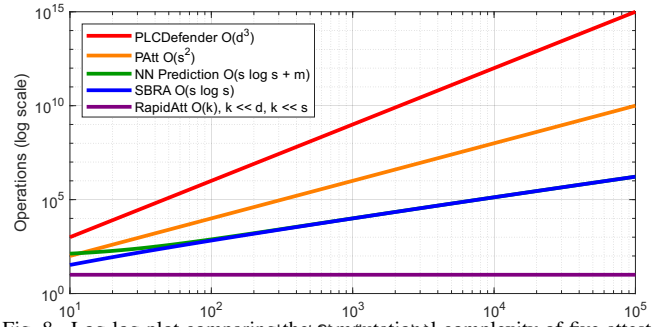

Fig. 8. Log-log plot comparing the computational complexity of five attestation techniques. PLCDefender exhibits cubic complexity $O(d^3)$ due to model-based verification [11]. PAtt scales quadratically with $O(s^2)$ by modeling sensor data [12]. NN Prediction has a complexity of $O(s \log s + m)$, driven by neural network inference [13]. SBRA demonstrates quasi-linear behavior with $O(s \log s)$ based on software checksums [14]. RapidAtt operates with a bounded complexity of of $O(k)$, where $k \ll d$, $k \ll s$, by verifying a limited number of randomly selected sections per cycle.

D. False Positives and Reliability

Across all experimental runs, RapidAtt exhibited a low false positive rate. No legitimate PLC activity was flagged, confirming high reliability. Randomized section checks maintained security without triggering false alerts, demonstrating a strong balance between accuracy and performance.

E. Overall Impact on IIoT Systems

Results show RapidAtt is a lightweight, cost-effective solution for PLC security in IIoT. It works with both modern and legacy PLCs and detects code tampering in real time, making it well-suited for industrial systems focused on safety and uptime.

VIII. CONCLUSION

This paper introduced RapidAtt, a light wight attestation technique designed to secure PLC programs in IIoT environments. By using periodic, randomized checks of selected program sections, RapidAtt ensures high detection accuracy with minimal computational overhead, making it ideal for legacy PLCs and resource-constrained settings. Experimental validation demonstrated RapidAtt's ability to rapidly detect and restore compromised code segments, achieving a total verification time of just $11.93\ ms$ and reducing execution latency by up to 17.67% compared to existing approaches, ensuring uninterrupted industrial operations. Future work includes exploring dynamic attestation frequencies based on system activity, evaluating scalability across larger and diverse IIoT networks. Extending RapidAtt to secure various heterogeneous IIoT devices beyond PLCs and assessing its robustness against sophisticated adversaries are also valuable future directions.

REFERENCES

[1] K. Stouffer, J. Falco, K. Scarfone *et al.*, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2011.

[2] M. A. Sehr, M. Lohstroh, M. Weber, I. Ugalde, M. Witte, J. Neidig, S. Hoeme, M. Niknami, and E. A. Lee, "Programmable logic controllers in the context of industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3523–3533, 2020.

[3] K. Stouffer, J. Falco, K. Scarfone *et al.*, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2011.

[4] S. Gibbs, "Triton: Hackers take out safety systems in 'watershed' attack on energy plant," *The Guardian*, December 2017, accessed: 2025-04-03. [Online]. Available: https://www.theguardian.com/technology/2017/dec/15/triton-hackers-malware-attack-safety-systems-energy-plant

[5] F.-S. Labs, "Blackenergy and quedagh: The convergence of crimeware and apt attacks," London, U.K., 2016.

[6] S. O. Athar, M. N. Aman, and B. Sikdar, "Duatt: A dual-layer attestation scheme for plc-based industrial internet of things," *IEEE Internet of Things Journal*, 2025.

[7] D. Feng, *Trusted Computing: Principles and Applications*. Walter de Gruyter GmbH & Co KG, 2017, vol. 2.

[8] W. Alsabbagh and P. Langendörfer, "Security of programmable logic controllers and related systems: Today and tomorrow," *IEEE Open Journal of the Industrial Electronics Society*, vol. 4, pp. 659–693, 2023.

[9] M. Alabadi, A. Habbal, and X. Wei, "Industrial internet of things: Requirements, architecture, challenges, and future research directions," *IEEE Access*, vol. 10, pp. 66 374–66 400, 2022.

[10] M. A. Baig, A. Iqbal, M. N. Aman, and B. Sikdar, "Leveraging ai to compromise iot device privacy by exploiting hardware imperfections," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 01, pp. 1–15, 2025.

[11] M. Salehi and S. Bayat-Sarmadi, "Plcdefender: Improving remote attestation techniques for plcs using physical model," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7372–7379, 2020.

[12] H. R. Ghaeini, M. Chan, R. Bahmani, F. Brasser, L. Garcia, J. Zhou, A.-R. Sadeghi, N. O. Tippenhauer, and S. Zonouz, "{PAtt}: Physics-based attestation of control systems," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, 2019, pp. 165–180.

[13] Y. Chen, C. M. Poskitt, and J. Sun, "Code integrity attestation for plcs using black box neural network predictions," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2021, pp. 32–44.

[14] J. Cao, T. Zhu, R. Ma, Z. Guo, Y. Zhang, and H. Li, "A software-based remote attestation scheme for internet of things devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1422–1434, 2023.

[15] D. Van Landuyt and W. Joosen, "A descriptive study of assumptions in stride security threat modeling," *Software and Systems Modeling*, pp. 1–18, 2022.

[16] M. Usama, M. N. Aman, and B. Sikdar, "Run-time self attestation of fpga based iot devices," *IEEE Internet of Things Journal*, 2024.