IMT School for Advanced Studies, Lucca
Lucca, Italy

Enabling the integrated and automated self-defense of Cyber-Physical Systems through an interdisciplinary approach

PhD Program in Cybersicurezza

Track in Software, System, and Infrastructure Security

XXXVIII Cycle

By

Giacomo Gori

2026

The dissertation of Giacomo Gori is approved.

PhD Program Coordinator: Prof. Mirco Tribastone, IMT School for Advanced Studies Lucca

Advisor: Prof. Franco Callegati, Alma Mater Studiorum - Universitá di Bologna

Co-Advisor: Prof. Marco Prandini, Alma Mater Studiorum - Universitá di Bologna

The dissertation of Giacomo Gori has been reviewed by:

Prof. Jorge Azorín-López, Dept. of Computer Technology and Computation, University of Alicante (Spain)

Julian Szymanski, Gdansk University of Technology (Poland)

IMT School for Advanced Studies Lucca
2026

Contents

List of Figures

List of Tables

Acknowledgements

This thesis is based on the following papers, sorted in the order they appear in the manuscript (C=Conference, J=Journal, S=Submitted):

[J1] Giacomo Gori et al. “A systematic analysis of security metrics for industrial cyber–physical systems”. In: Electronics 13.7 (2024), p. 1208.

[J2] Giacomo Gori et al. “GRAPH4: A Security Monitoring Architecture Based on Data Plane Anomaly Detection Metrics Calculated over Attack Graphs”. In: Future Internet 15.11 (2023), p. 368.

[SJ1] Gori Giacomo, Lorenzo Rinieri, Melis Andrea, Girau Roberto, Prandini Marco, and Callegati Franco. “Pk-IOTA: Blockchain empowered Programmable Data Plane to secure OPC UA communications in Industry 4.0”. Submitted to: Computers & Security, publisher Elsevier.

[SJ2] Giacomo Gori, Higinio Mora, Francisco J. Mora-Gimeno, and Marco Prandini. “Scalable Trust Management in Federated Learning for IoT Using DAG Structures”. Submitted to: IEEE Communications Standards Magazine on the special issue December 2025/Standardization and Integration of Blockchain and Federated Learning, publisher IEEE.

[SJ3] Giacomo Gori, Higinio Mora, Francisco J. Mora-Gimeno, and Marco Prandini. “DAGTrustFL: Scalable Trust Management for Federated Learning”. Submitted to: IEEE Transactions on Dependable and Secure Computing on the special issue Security and Privacy in Federated Learning and Unlearning, publisher IEEE.

[C1] Giacomo Gori et al. “Unraveling The Link Between Sociodemographics and Cybersecurity”. In: (2025) Proceedings of the 2025 Americas Conference on Information Systems (AMCIS), publisher AIS.

[C2] Shuyuan Mary Ho et al. “Synthetic Lies, Digital Truths: A Systematic Review of Computer-Mediated Deception Research in the Era of AI and Deepfakes”. In: (2025) Proceedings of the 2025 International Conference on Information Systems (ICIS), publisher AIS.

Chapter 2 is based on the research paper [J1].
Paper [J1] presents the result obtained from a systematic review of available security metrics for Industrial Cyber Physical Systems. The design and methodology of the review approach originated from me. The selection and filtering of the metrics, as well as the discussion of the results, were jointly performed by Rinieri Lorenzo and me. Feedback on the design and evaluation phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper.

Chapter 3 is based on the research papers [J2], [SJ1], [SJ2], and [SJ3].
Paper [J2] presents an architecture that exploits Attack Graph metrics to detect and automatically block attackers in distributed systems. The concept was ideated by Rinieri Lorenzo and me. The part regarding P4 was developed by Rinieri Lorenzo and Melis Andrea, whereas the part focusing on Attack Graphs and their generation was addressed mainly by me. The implementation of the use case and the experiments were carried out by Rinieri Lorenzo and me. Feedback on the design and evaluation phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper.
Paper [SJ1] introduces a novel approach that combines P4 programmable switches and blockchain technologies to achieve

similar goals, within the broader context of usable security. The original idea of applying blockchain and P4 switches to the OPC UA certificate distribution use case was conceived by Rinieri Lorenzo, with minor contributions from Andrea Melis and Amir Al Sadi. I proposed leveraging the two-layer architecture of IOTA (L1 and L2) for integrating blockchain and P4. I handled all the blockchain implementation and evaluation, while Rinieri Lorenzo was responsible for the P4-based components. Feedback on the design and evaluation phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper.
Paper [SJ2] reviews current trust-management approaches in Federated Learning, identifying limitations in DLT-based solutions and open challenges. It then introduces a scalable DAG-driven, context-aware framework that advances standardization efforts for trust in FL-for-IoT and outlines future research directions. The idea and concept of the article originated with me and the professors with whom I spent my period abroad in Alicante, Higinio Mora-Mora and Francisco Mora-Gimeno. The literature review, the analysis on limitations and open challenges, and the proposal of a model were done by me. Constant feedback on the design and modeling phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper.
Paper [SJ3] extends the previous proposal into a concrete formalization of the model and the consequent implementation with experiments on the performance and feasibility. The idea and concept of the article originated by me, Higinio Mora-Mora, and Francisco Mora-Gimeno. I conducted the formalization of the model, its implementation, and the experiments. Constant feedback on the design and modeling phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper.

Chapter 4 is based on the research papers [C1] and [C2].

Paper [C1] presents the results of a systematic literature review on the influence of socio-demographic factors and cybersecurity behavior, awareness, and attitude. The idea of the article and the methodology originated by me. The review process and the qualitative and quantitative analysis were performed by Martucci Giordana, Ganzaroli Alessia, and me. Feedback on the design and evaluation phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper. The article was presented at the AMCIS25 conference, held in Montreal (CA), by me.
Paper [C2] presents the results of a systematic literature review over information systems literature across physical and computer-mediated settings, with particular attention to the evolving role of multimodal data, especially audio-visual manipulation, in facilitating deception. The idea originated from Prof. Mary Ho Shuyuan. The literature review and the results analysis were performed by all co-authors. Feedback on the design and evaluation phases was provided by all co-authors. All authors contributed to the writing and reviewing of the paper. Prof. Mary Ho Shuyuan, Liu Yue, Manzoor Hussain Ghazal, and I presented the article at the ICIS25 conference, held in Nashville (USA).

In preparing this doctoral dissertation, artificial intelligence tools were used solely to support writing and language revision activities. Specifically, such tools helped improve clarity of expression, refine sentence structure, and correct grammatical errors and typographical mistakes. The development of the scientific content, critical analysis, methodological design, and conclusions presented herein remains entirely the independent work of the author.

Vita

September 2, 1998	Arezzo (AR), Italy
2020	Bachelor Degree in Computer Engineering Final mark: 110/110 cum laude Alma Mater Studiorum, University of Bologna (Italy)
2022	Master Degree in Computer Engineering Final mark: 110/110 cum laude Alma Mater Studiorum, University of Bologna (Italy)

Publications

1. Gori, Giacomo, et al. “Unraveling The Link Between Sociodemographics and Cybersecurity.” (2025), in Proceedings of the Americas Conference in Information Systems (AMCIS) 2025.
2. Gori, Giacomo, et al. “Synthetic Lies, Digital Truths: A Systematic Review of Computer-Mediated Deception Research in the Era of AI and Deepfakes.”, in Proceedings of the International Conference in Information Systems (ICIS) 2025.
3. Gori, Giacomo, et al. “A systematic analysis of security metrics for industrial cyber–physical systems.”, in Electronics 13.7 (2024): 1208.
4. Gori, Giacomo, et al. “GRAPH4: A Security Monitoring Architecture Based on Data Plane Anomaly Detection Metrics Calculated over Attack Graphs.”, in Future Internet 15.11 (2023): 368.
5. Gori, Giacomo, et al. “Towards the Creation of Interdisciplinary Consumer-Oriented Security Metrics.”, in 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023.
6. Gaiba, F., Bedogni, L., Gori, G., Melis, A., & Prandini, M. (2024, January). “Wi-Fi sensing for human identification through ESP32 devices: An experimental study”, in 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC) (pp. 206-209). IEEE.
7. Apollonio, F., Bedogni, L., Gori, G., Melis, A., & Prandini, M. (2024, January). “On the Trade-Off Between Privacy and Information Quality in Location Based Services.”, in 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC) (pp. 994-997). IEEE.
8. Gori, Giacomo et al. “Scalable Trust Management in Federated Learning for IoT Using DAG Structures” submitted on 31 March 2025 to IEEE Communications Standards Magazine. [Waiting for review]
9. Gori, Giacomo et al. “Pk-IOTA: Blockchain empowered Programmable Data Plane to secure OPC UA communications in Industry 4.0.”, submitted on 26 September 2025 to Future Generation Computer Systems, Elsevier. [Waiting for review]
10. Gori, Giacomo et al. “DAGTrustFL: Scalable Trust Management for Federated Learning”, submitted on XX XX 2025 to IEEE Transactions on Dependable and Secure Computing, special issue on Security and Privacy in Federated Learning and Unlearning, IEEE. [Waiting for review]

11. Gori, Giacomo et al. “An industrial network digital twin for enhanced security of cyber-physical systems”. In 2022 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-7). IEEE.

12. Gori, Giacomo et al. “Metrics for Cyber-Physical Security: A call to action”. In 2022 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, 2022.

Presentations

1. Gori, Giacomo, "Unraveling The Link Between Sociodemographics and Cybersecurity.", at Americas Conference in Information Systems (AMCIS) 2025, Montréal, Canada, August 2025.
2. Gori, Giacomo, et al. "Synthetic Lies, Digital Truths: A Systematic Review of Computer-Mediated Deception Research in the Era of AI and Deepfakes.", at International Conference in Information Systems (ICIS) 2025, Nashville, U.S., December 2025.
3. Gori, Giacomo, "A systematic analysis of security metrics for industrial cyber–physical systems.", at Italian Conference on Cybersecurity (ITASEC), Salerno, Italy, April 2024.
4. Gori, Giacomo, "GRAPH4: A Security Monitoring Architecture Based on Data Plane Anomaly Detection Metrics Calculated over Attack Graphs.", at 20th Italian Networking Workshop (INW), Madonna di Campiglio, Italy, January 2024.

Abstract

Cyber-physical systems underpin many of today's critical infrastructures, including industrial automation, energy production, healthcare environments, and smart-city ecosystems. These systems tightly couple software, networks, and physical processes, operate under real-time and safety constraints, and span heterogeneous technologies and organizational domains. Their growing interconnection has expanded the attack surface across enterprise, demilitarized, and operational networks, as well as across devices ranging from modern edge platforms to legacy industrial controllers. However, three major gaps limit the effectiveness of current security approaches. First, although existing standards and frameworks support governance and compliance, they provide few system-level security metrics that are meaningful in industrial environments, that can be measured continuously, and that satisfy formal conditions of soundness and reproducibility. Second, even when indicators exist, there is no clear architectural path to transform them into timely, auditable defensive actions without undermining availability or operational continuity. Distributed infrastructures must exchange trust, policy, and posture information in ways that resist tampering, avoid single points of failure, and remain scalable. Third, security in these environments depends not only on technical mechanisms but also on human behavior, organizational practice, and emerging forms of deception. Operators increasingly face social-engineering campaigns that exploit synthetic media and AI-generated content, while organizations lack clear evidence on how to tailor training and awareness programs in a scalable and context-appropriate manner. This thesis addresses these gaps through

an integrated research agenda that links measurable security, enforceable distributed architectures, and the human dimension within industrial and human-cyber-physical systems.

The thesis adopts a consistent methodological approach across its contributions: it first conducts a critical synthesis of prior literature to identify conceptual and practical gaps, then designs models and architectures that address these gaps, and finally implements and evaluates prototype systems under realistic constraints such as device heterogeneity, timing requirements, and resource limitations.

The first contribution addresses security metrics for industrial cyber-physical systems through a multi-stage systematic pipeline for collecting, classifying, filtering, and validating indicators from academic and industrial sources. Metrics are documented using a uniform schema, mapped to established taxonomies, and filtered according to inclusion criteria tied to industrial networks and constraints. The remaining candidates are evaluated against formal Conditions for Sound Security Metrics, covering clarity, measurability, monotonicity, and reproducibility. From 278 proposals, only 32 metrics satisfy all conditions, revealing that many published indicators lack clear definitions, units, or links to real security outcomes. Most validated metrics support system-level assessment across CIA properties, while notable gaps emerge for human-related and authentication metrics, and challenges remain around context-awareness and cross-environment comparability.

Building on the need to move from measurement to action, the second contribution introduces GRAPH4, a system that operationalizes security metrics through attack-graph–driven monitoring in programmable networks. The approach generates attack graphs in the control plane to identify the network components and flows that are relevant to a given threat scenario, based on knowledge of the full network topology. These

graphs are then translated into rules that are deployed in programmable data-plane devices, enabling metric collection, anomaly detection, and alerting to be performed directly at line rate. By distributing responsibilities between control and data planes, GRAPH4 supports timely and resource-aware responses to emerging conditions while preserving operational continuity. Experimental evaluation demonstrates that this architecture improves detection efficiency and responsiveness with modest overhead, showing the feasibility of in-network measurement and mitigation in industrial environments.

The third contribution addresses identity, trust, and resilience in distributed and collaborative environments where devices cannot be assumed to be trustworthy by default. The thesis explores two complementary directions. The first focuses on secure identification and authorization in industrial communication that uses OPC UA. Although OPC UA provides strong native security features, prior work shows that secure deployments remain difficult to achieve in practice, due to complex certificate management workflows and heterogeneous support for security extensions across devices. To address these challenges, the thesis introduces PK-IOTA, an automated framework that secures OPC UA communications by combining in-network certificate validation with decentralized credential coordination. Programmable data-plane switches validate certificates inline, while certificate issuance, revocation, and lifecycle events are recorded on a direct acyclic graph ledger, providing a tamper-evident and scalable distribution layer without reliance on a single authority. Evaluation on a physical industrial testbed demonstrates that Pk-IOTA adds only minimal overhead while enabling consistent, auditable, and resilient certificate management in operational environments.

The second direction addresses trust and resilience in collaborative machine learning among untrusted devices, a class of

scenarios that increasingly affects industrial and IoT ecosystems. Existing trust-management approaches for federated learning often rely on centralized or blockchain architectures that do not scale and are difficult to audit in adversarial settings. This raises the core research question of how to design a trust mechanism that remains both scalable and tamper-resistant while operating in decentralized environments. To answer this question, the thesis introduces a trust-management mechanism based on a Direct Acyclic Graph (DAG) ledger that records model updates and distribute trust assessment in the network. The design adapts the ledger's tip-selection logic so that the graph advances preferentially on contributions assessed as trustworthy according to context-aware indicators. These indicators are used to weight or filter local updates before aggregation, while smart-contract logic automates the verification and exclusion of suspected adversarial behavior. The system is evaluated using a lightweight convolutional model on Fashion-MNIST under controlled poisoning and collusion settings. Results show that the DAG-based trust layer improves selectivity against malicious contributions with limited computational cost, preserves model utility near the clean baseline under moderate adversarial pressure, and achieves linear scalability with essentially constant transaction time as the number of participants increases.

The final contribution concerns the human dimension of cybersecurity in industrial and socio-technical environments. The thesis conducts two systematic reviews to consolidate fragmented evidence from different disciplines and inform human-centered security practice. The first review examines the relationship between sociodemographic attributes and awareness, attitudes, behavior, and training outcomes in cybersecurity. Using a structured search and PRISMA-based screening process, the review synthesizes findings from 68 studies, providing both qualitative interpretations and quantitative sum-

maries. The analysis reveals that age, education, job sector, culture, and gender are associated with significant differences in how individuals perceive and manage cyber risk, although the effects are nuanced, context-dependent, and not deterministic. These factors are best understood as a neutral starting point for designing tailored training rather than as grounds for stereotyping or exclusion. The review also notes that advanced analytical approaches are still rarely applied in this field, largely due to the scarcity of suitable datasets, and suggests that future research will benefit from richer data and responsible modelling practices. The second review focuses on deception in computer-mediated contexts, with particular attention to deepfakes and other AI-generated content. It surveys 57 studies from the information systems field, maps linguistic, behavioral, physiological, and contextual cues associated with deceptive interactions, and analyzes the theoretical perspectives and methodological choices that underpin deception research. The review shows that techniques originally developed for face-to-face cues or simple textual analysis are insufficient in environments where attackers combine multiple media channels, exploit anonymity and asynchrony, and manipulate social and emotional signals. It argues for multimodal approaches that integrate technical detection capabilities with training, governance, and organizational practice.

Chapter 1

Introduction

Cyber–Physical Systems (CPS) blend software, communication networks, and physical processes into a single operational fabric [1]. Embedded controllers sense the environment, run analytics, and actuate responses on machines and infrastructures. This integration underpins critical domains, including manufacturing, energy, transportation, healthcare, and smart cities, where digital decisions have immediate physical consequences. As connectivity increases, so does exposure to faults, misconfigurations, and adversarial actions. Industry estimates report global cybercrime losses exceeding $400 billion and surpassing $1 trillion in broader assessments[2]. Pervasive IoT deployment in CPS expands the attack surface; recent examples include:

- Verkada (large-scale camera compromise and exposure of cloud-stored footage)[3];
- SolarWinds (supply-chain compromise)[4];
- BotenaGo (malware infecting routers and IoT devices)[5];

[1] https://www.nsf.gov/news/special_reports/cyber-physical/

[2] https://ir.mcafee.com/news-releases/news-release-details/new-mcafee-report-estimates-global-cybercrime-losses-exceed-1

[3] https://www.verkada.com/security-update/report/

[4] https://www.cisecurity.org/solarwinds

[5] https://www.iotworldtoday.com/2021/11/16/botenago-malware-targets-millions-of-iot-devices/

- Yarix (the cybersecurity company that discovered a commercial website for buying audio-video recordings illegally stolen from home/offices IoT cameras)[6];

Securing CPS, therefore, demands approaches that are technically rigorous, operationally realistic, and attentive to human work.

1.1 Cyber–Physical Systems: definition and security challenges

CPS connects sensors, controllers, and actuators through computation and communication to manage a physical process [11]. Typical industrial examples include process plants coordinated by programmable logic controllers and supervisory control systems, autonomous material handling in warehouses, and distributed monitoring of utilities. Outside the factory, CPS appear in connected vehicles, building automation, and medical devices. Despite their variety, these systems share a common architecture: field devices at the edge, cyber services that coordinate decisions, and interfaces where humans supervise, diagnose, and intervene.

Security in CPS encompasses more than just the confidentiality and integrity of data [11]. The availability and timeliness of control signals affect safety and production quality. Attacks that would be considered a nuisance in enterprise IT can have a disproportionate impact in control environments, where jitter, packet loss, or stale configuration may propagate to physical damage. The Science of Security agenda[7] emphasizes the need for measurable and testable security in such systems, with a particular focus on metrics that guide evaluation, design, and deployment. Traditional guidance for information security measurement (e.g., ISO 27004 [19] and NIST SP 800–55 [51]) offers valuable management indicators but does not provide straightforward metrics to evaluate cyber–physical coupling and real-time operation. Bridging this gap is central to this thesis.

[6] https://www.yarix.com/en/News/News/Chiuso-il-sito-che-vendeva-illecitamente-accessi-a-videocamere-di-sorveglianza

[7] https://cps-vo.org/group/SoS/about

Modern CPS seldom operate in isolation [11]. Industrial networks interconnect operational technology and enterprise services through demilitarized zones, remote maintenance channels, and cloud analytics. This interconnection improves efficiency and visibility, but it also expands the attack surface across legacy equipment, vendor ecosystems, and heterogeneous protocols. As systems become data-driven and service-oriented, adversaries have more opportunities to exploit supply chains, weak identities, or misaligned configurations. Measurement and assurance must therefore scale across layers and sites, remain comparable under diversity, and support automation without sacrificing safety [125].

The societal and economic relevance of CPS motivates stronger assurance. In manufacturing, disruptions translate into financial losses and contractual penalties. In healthcare and utilities, failures can endanger people or degrade essential services. Security controls must respect real-time constraints and quality-of-service guarantees, and they must be auditable to meet regulatory and certification requirements. A purely technical perspective is not sufficient. Human operators and engineers plan, supervise, and recover systems. Their cognition and coordination shape outcomes during both routine work and incidents.

1.2 Research gaps and motivation

Despite significant progress, three gaps persist at the intersection of security research and CPS practice.

Need for measurable security. A large body of work recommends the use of metrics for governance and reporting, yet validated and reproducible system-level measurements tailored to industrial contexts remain limited. Many proposals focus on single components, overlook dependencies and emergent behavior, or lack clear data sources and units of measurement. There is a need for metrics that can be collected with realistic effort in plants, that support comparison across deployments, and that are suitable for both design-time assurance and run-time operation [98].

Architectural resilience. Metrics deliver value when they close the loop from observation to action. Industrial distributed environments require architectures that can embed measurements in the network and at endpoints, validate their provenance, and use them to orchestrate safe responses [125]. Centralized authorities create bottlenecks and single points of failure, while heterogeneous devices and protocols complicate uniform enforcement [321]. Programmable data planes and trustworthy coordination substrates offer promising building blocks, provided they remain compatible with legacy constraints and support real-time performance.

Role of human and social factors. Security outcomes depend on people as much as on technology. Operators, engineers, and end-users are part of both the attack surface and the defense [314]. Organizational practices, training, and cognitive biases influence detection, escalation, and recovery. As synthetic media and automated influence campaigns proliferate, deception targets human judgment with increasing realism [229]. A comprehensive approach must therefore incorporate human factors and provide pathways for personalized, effective training that match roles, contexts, and risks.

1.3 Thesis objectives and contributions

This thesis advances an integrated view of CPS security that unites metric-driven evaluation, deployable architectures, and human factors. The first objective is to consolidate a validated catalogue of security metrics for industrial CPS, together with a methodology for collection, classification, filtering, and soundness validation. The second objective is to design and assess architectural mechanisms that operationalize metrics through in-network enforcement and distributed trust substrates, supporting identity, policy, and aggregation at scale. The third objective is to extend the scope from CPS to Human–Cyber–Physical Systems, clarifying how human cognition and social dynamics affect security, and how training and deception research can be aligned with industrial realities.

The value added lies in an interdisciplinary approach that connects

quantitative measurement, programmable networking, distributed ledgers, machine learning, and human factors to deliver security that is auditable, scalable, and aligned with real work.

1.4 Approach and artifacts

The work combines systematic synthesis with design and empirical evaluation, and for each topic, the thesis follows a consistent arc. It first maps the state of the art, mainly through systematic literature reviews, and analyzes the evidence to surface gaps and unmet requirements. From this basis, it proposes a resolution model, designs and implements the corresponding mechanism, and evaluates effectiveness and efficiency under realistic conditions. This process yields a vetted set of security metrics suitable for industrial CPS, networked prototypes that enact metric driven enforcement with programmable data planes and distributed ledgers, a distributed based trust mechanism that scales Federated Learning and resists poisoning, and human centered results that quantify the role of sociodemographics and consolidate evidence on computer mediated deception, including deepfakes and multimodal cues, to guide tailored training and organizational practice.

This thesis is structured as follows. Chapter 2 defines security metrics for industrial CPS, presents the collection and validation methodology, and reports the resulting taxonomy and gaps. Chapter 3 translates metrics into action through in-network enforcement and trustworthy coordination, and evaluates the impact on performance and assurance. Chapter 4 extends the lens to Human–Cyber–Physical Systems, examines the influence of sociodemographics on cybersecurity outcomes, and surveys deception in the AI era with a focus on multimodal indicators. Chapter 5 synthesizes findings and outlines future research on standardization, deployment, and interdisciplinary validation in real infrastructures.

Chapter 2

Security Metrics for CPS

2.1 Definition and role of security metrics

2.1.1 Definition

Security metrics are quantitative or qualitative measurements, produced over time, used to assess specific security properties and to support risk assessment, performance evaluation, and continuous improvement. In practice, metrics underpin agendas for management, compliance, and reporting (e.g., ISO 27004 [19], NIST SP 800–55 [51]). These frameworks primarily highlight organizational processes and policy performance; complementary, system-level measurements are required in settings where cyber events may propagate to the physical world (e.g., CPS).

2.1.2 Role

The role of security metrics includes:

- Quantification: producing consistent, comparable measures of security-relevant properties (e.g., counts, rates, scores, state variables).
- Operational linkage: providing machine-readable signals that connect analysis to action for governance, assurance, and operations (e.g., threshold-based policies, resource reallocation, configuration updates).

European initiatives, such as the Cybersecurity Act, seek to harmonize assurance through certification [273], highlighting the need for metrics that remain meaningful across heterogeneous devices and operational contexts.

2.1.3 Basic properties

Effective metrics exhibit the following characteristics:

- Well scoped: they target clearly identified properties (e.g., confidentiality, integrity, availability) with explicit data sources and units.
- Feasible and automatable: they enable continuous assessment at sustainable cost. Improvements in measured values correspond to genuine security gains across deployments.
- Reproducible: repeated measurements under the same conditions yield consistent values across operators and deployments.

Operationally, a metric can be implemented as a structured questionnaire or computation that maps observations to a value or score [4]. To aid selection and comparison, taxonomic descriptors are useful, such as result type (nominal, ordinal, interval, ratio, absolute, distribution), scope (users, software, hardware, network, organization), automation (manual vs. automatic computation), and measurement (static vs. dynamic/real-time) [4]. Domain-specific taxonomies may be required, as shown for embedded systems in [169].

2.2 State of the Art

Defining standard, quantitative security metrics that are valid across contexts is non-trivial. Philippou, Frey, and Rashid [219] explicitly criticizes widely used guidance for insufficient contextualization and weak alignment with business objectives. They advocate a goal-driven strategy that traces each metric to concrete objectives, yielding more precise and suitable outcomes at the cost of substantial upfront effort to establish and maintain the traceability between objectives and measures.

Because contextualization is domain-dependent, several works narrow their scope to specific areas, enabling a transition from qualitative desiderata to quantitative, operational metrics. Wang, Jajodia, and Singhal [286] focus on network security metrics, reviewing advantages and limitations of common measures. Longueira-Romerc et al. [169] address embedded systems, filtering an initial pool of >200 metrics and selecting 169 for evaluation using SMART [62], PRAGMATIC [34], and quality characteristics from Savola [245]; their analysis targets comparability, cost-effectiveness, measurability, repeatability, and reproducibility [169].

Other contributions introduce domain-specific quantitative constructs (e.g., metrics for "stealthy" attacks in control settings [197]). Survey-based works (e.g., Pendleton et al. [218]) compare proposals across system security and assess effectiveness with respect to vulnerabilities, attack severity, and defense strength, concluding that notable gaps persist between research outcomes and desirable metric properties. For certain subfields, conformance perspectives (e.g., Hauet on ISA99/IEC 62443 [105]) provide clearer definitions of what constitutes good metrics, although guidance remains uneven across domains. Policy-level frameworks (e.g., the EU Cybersecurity Act [273] and its recently approved amendment[1]) aim for harmonization through certification; however, the heterogeneity of devices and operational contexts limits the applicability of one-size-fits-all prescriptions.

A recurring difficulty is that the first coarse selection of suitable metrics is both critical and under-standardized. The literature documents three complementary strategies:

- classification and selection, which defines a domain-coherent taxonomy and selects metrics to cover required aspects (e.g., [261], [195]);
- automatic generation from contextual goals and security objectives (e.g., [13]);

[1] https://digital-strategy.ec.europa.eu/en/library/proposed-regulation-managed-security-services-amendment

- multivocal literature review (MLR) that mines academic and grey literature via snowballing and staged filtering (e.g., [75]);

Context also modulates metric behavior, even for seemingly similar applications. Traditional Biometric Systems and Wearable Biometric Systems exhibit different threats and vulnerabilities, motivating the use of different metrics [262].

To make the space navigable, a widely used technical taxonomy (referenced from Pendleton et al. [218]) partitions metrics into four types:

- defense metrics: quantify the strength, coverage, and operational cost of preventive, reactive, and proactive mechanisms (e.g., adaptation dynamics [122]).
- Vulnerability metrics: characterise exposure (e.g., password weaknesses, attack surface [174], software flaws as in CVSS[2]).
- Attack metrics: quantify attacker capability and activity (e.g., botnet bandwidth in DDoS, malware obfuscation prevalence, packer complexity in layers or granularity [238]).
- Situation metrics: capture the evolving security state under attack–defense dynamics (e.g., incident frequency, security investment [315]); these include data-driven indicators (e.g., Network Maliciousness [317]) and model-driven constructs (e.g., fraction of compromised hosts).

Turning qualitative notions into quantitative measures requires sound evaluation criteria. Ahmed [3] underline that a valid metric should rely on properties that are measurable through consistently accessible data, that data collection must be feasible and preferably automated while accounting for costs, that the method of quantification should be clearly defined (for example, using counts, rates, or percentages), and that the metric must include explicit units of measurement. Savola [245] argue that effective metrics should reduce to interpretable scores when appropriate; CVSS is a canonical example of a composite score that mixes intrinsic,

[2]https://nvd.nist.gov/vuln-metrics/cvss

temporal, and environmental factors via a published, open method[3]. Beyond checklists, Yee [309] formalize Conditions for Sound Security Metrics (CSSM), requiring metrics to be well-defined (meaningful, objective, unbiased, sufficiently complete, and affordable), progressive (monotone with respect to actual security), and reproducible (strongly or weakly, across environments).

CPS introduces additional risks that influence metric design. Threat modelling frameworks (e.g., STRIDE [251]) help systematize properties to be measured (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege). However, empirical analyses suggest that many metrics still focus on isolated components. Aigner and Khelil [4] benchmark CPS metrics against CPS-specific conditions and find good coverage of many desired features, yet persisting gaps in attack detection and, crucially, insufficient treatment of dependencies and side effects in System-of-Systems (SoS) contexts: emergent properties from component composition are often neglected, hindering accurate system-level assessment. More generally, flawed metrics can stem from measuring the wrong factors, omitting relevant ones, or introducing subjectivity and bias; Yee [308] documents such pitfalls and advocates rigorous design-and-test cycles. Community efforts (e.g., SecurityMetrics.org[4]) reflect the ongoing debate at the intersection of policy and technology.

In summary, progressing from qualitative indicators to quantitative, operational measures in CPS/ICPS entails:

- anchoring metrics to context and objectives, even when this increases specification effort ([219]);
- using structured selection pipelines (classification, generation, MLR) to mitigate ad hoc choices ([261, 195, 13, 75]);
- adopting taxonomies that cover defense, vulnerability, attack, and situational perspectives ([218, 174, 238, 315, 317, 122]);

[3] https://www.first.org/cvss/
[4] https://www.securitymetrics.org

- validating soundness via criteria ([3, 245]) and formal conditions (e.g. CSSM [309]).

Desired characteristics for CPS/ICPS include efficiency and cost-effectiveness, for both design-time certification and run-time reconfiguration, agility to keep pace with evolving vulnerabilities, interpretability, including for non-expert stakeholders, and a true system-level focus that extends beyond individual components. Where appropriate, environmental and impact-sensitive factors should be integrated to differentiate semantically similar vulnerabilities whose physical or financial consequences diverge across deployments.

2.3 A systematic analysis of security metrics for ICPS

The literature consistently indicates that, although security metrics are widely recommended for governance and reporting, validated, system-level measures tailored to CPS, and especially ICPS, remain limited. Survey and position papers highlight persistent gaps between available proposals and desirable properties of metrics [218], while policy-oriented frameworks primarily focus on organizational processes rather than on measurements that capture the cyber–physical interplay typical of CPS [19, 51]. Further critiques emphasize insufficient contextualization and weak alignment with operational or business objectives, calling for goal-driven strategies that explicitly trace each metric to clearly defined objectives [219]. Domain-specific efforts confirm the difficulty of moving from qualitative desiderata to quantitative, operational measures: targeted filters for embedded systems require substantial curation [169], and evaluation checklists underline the need for measurability, feasibility, and explicit units and methods [3, 245].

In CPS settings, additional constraints compound these shortcomings. Benchmarks reveal that many metrics focus on isolated components but fail to sufficiently address dependencies, emergent behaviors, and side effects in System-of-Systems contexts; coverage of attack detection and

system-level posture remains incomplete [4]. At the same time, the heterogeneity of industrial devices and operating environments limits the applicability of one-size-fits-all approaches, reinforcing the need for domain-aware, reproducible metrics [273].

Motivated by these observations, this chapter undertakes a systematic analysis aimed at identifying, classifying, and vetting security metrics that are applicable to ICPS. The objective is to consolidate a coherent, context-sensitive set of measures that can be collected with realistic effort in industrial environments, support comparability across deployments, and are suitable for both design-time assurance and run-time operation. The following sections detail the methodology for collection, filtering, and validation, present a taxonomy tailored to ICPS, and discuss representative use cases and limitations.

2.3.1 Methodology: collection, classification, filtering, validation

This section outlines the methodology for the systematic collection, selection, and validation of security metrics tailored to ICPS. By restricting the domain, the process yields a concrete, domain-aware set of metrics, while remaining general enough to be re-applied to other CPS contexts. The workflow is presented in algorithmic form in Fig. 0 and summarized in Fig. 1. The initial dataset, intermediate reduction steps, and the final metric set are publicly available[5]. Throughout the process, selections were reviewed with domain experts to ensure accuracy and consistency.

Collection To assemble a comprehensive pool of candidates, the methodology begins with the pre-filter dataset reported by Longueira-Romerc et al. [169] (i.e., before their embedded-systems-specific reductions), which aggregates over 500 metrics from multiple sources [218, 230, 69, 194]. This seed was extended through additional searches in IEEE Xplore, Elsevier, ACM Digital Library, Springer, and Google Scholar, using terms such as "security metric," "security assessment," "ICPS," and "CPS." Inclusion

[5]https://doi.org/10.5281/zenodo.10142113

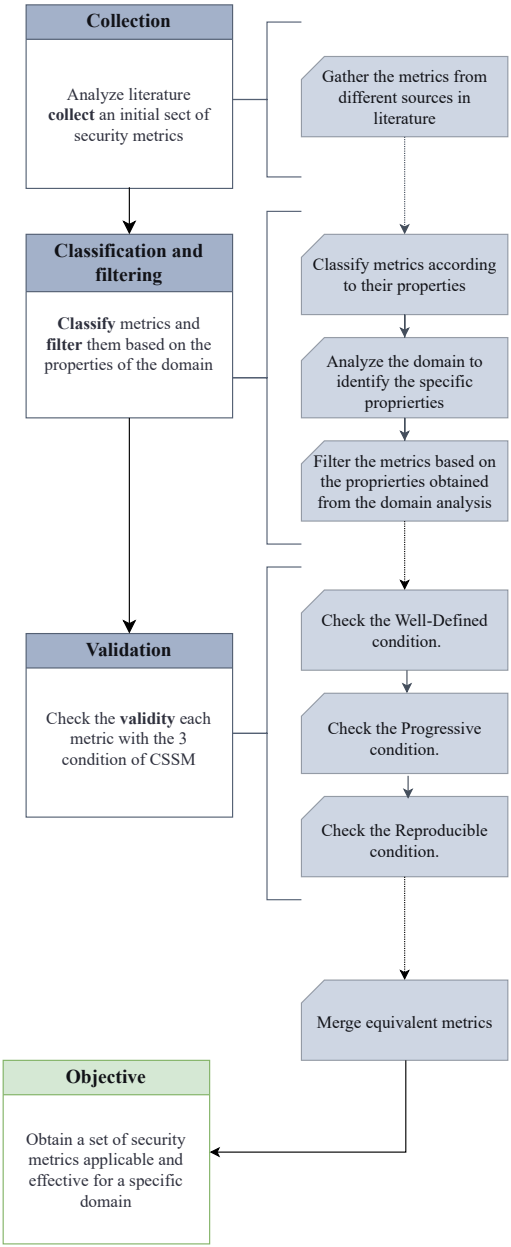


Figure 1: A graphical summarization of our objective with the several steps required for the achievement of the set of security metrics for a specific domain.

prioritized works that present security measurements or metrics with sufficiently clear definitions for real use cases and surveys that collect or compare security metrics. This step additionally incorporated 14 metrics from Boyer and McQueen [33] and 29 from Bhol, Mohanty, and Pattnaik [27]. After pruning duplicates and not-referenced entries, the initial set comprised 278 distinct metrics.

Classification Each metric is described through a unified schema adapted from established taxonomies ([283, 244]):

- Name: concise label.
- Definition: what is measured.
- Meaning: why it is informative for security.
- Weakness: assumptions/limitations and required external inputs.
- Scope: locus (network, device, user, organization, system).
- Result type: qualitative vs. quantitative; scale (nominal/ordinal/interval/ratio/absolute/distribution).
- Automation: manual vs. automatic computation.
- Measurement: static vs. dynamic (run-time).
- Construction: directly measured vs. modelled via artifacts (e.g., Attack Graphs [207]).

This schema standardizes documentation, enabling downstream filtering and validation.

Domain analysis (ICPS context). The dataset is structured by classifying metrics based on the most common attributes present in the literature, as shown in Section 2.2. Classifying metrics according to specific characteristics enables the selection of them based on criteria that depend on the context. In this case, I refer to the domain of ICPS, so it is necessary to study its intrinsic characteristics before proposing a set of metrics that

fully capture the security issues that may arise. ICPS are composed of interconnected Cyber and Physical components that monitor and manage physical processes. They are responsible for the safety and operation of the industrial process, which involves managing heterogeneous hardware and software. They include devices such as sensors, actuators, Supervisory Control and Data Acquisition (SCADA) systems, Human-Machine Interfaces (HMI), and dedicated subsystems, such as Programmable Logic Controllers (PLC) [53]. This heterogeneity obviously translates into system complexity, which implies more effort to manage and prevent anomalies. Additionally, ICPS networks utilize a diverse range of protocols, tailored to the specific objectives of each system. Real-time constraints and legacy hardware are two of the most important challenges that industrial protocols are specifically made to address. The Purdue Enterprise Reference Architecture [297] is the reference networking architecture for ICPS systems, adopted in the ANSI/ISA-95 standard, and divides ICPS network into three logical segments: the lower layer is the Manufacturing Zone, also known as Operational Technology (OT), while the upper layer constitute the Enterprise Zone, also referred to as Information Technology (IT), with a Demilitarized Zone of convergence between them. The OT network includes hardware and software used to monitor and manage industrial equipment, assets, processes, and events. On the other side, the traditional Information Technology IT network contains workstations, databases, and other typical machines used to manipulate information. From this perspective, the main concerns of IT systems are the confidentiality and integrity of the data, whereas for the OT part, availability is fundamental, as it can guarantee human safety and fault tolerance [318].

Filtering and reductions. Based on the domain analysis and the classification attributes, the following inclusion criteria guide filtering:

- The metric definition is applicable to IT/OT networks, components, protocols, or devices.
- The meaning explicitly relates to at least one security property among Confidentiality, Integrity, and Availability.

- The weakness indicates requirements or issues that are resolvable within the ICPS domain.
- The scope is one of: Network, Device, System, or User.

Two reduction steps are applied. First, metrics outside the target scopes are removed. Second, the remaining metrics are checked against the other three criteria. As an illustration (Tab. 1), Infection Rate [50] is retained (malware spread is a relevant IT-side concern in industrial contexts [93]), whereas ISP badness metric [135] is excluded due to lack of linkage to ICPS components.

Table 1: Example metrics that match or do not match the first three inclusion criteria (IC) regarding the second reduction step.

Metric	Definition	Ref	Match IC
Infection Rate	Average number of computers that can be infected by a compromised computer (per time unit) at the early stage of spreading	[50]	Yes
ISP badness metric	Quantifies the effect of spam from one ISP or Autonomous System on the rest of the Internet, comparing the "spamcount" with its "disconnectability"	[135]	No

Validation (CSSM) The filtered set is then vetted using the CSSM ([309]):

- Well-defined: meaningful, objective, unbiased, sufficiently complete, affordable.
- Progressive: metric evolution aligns monotonically with actual security posture.
- Reproducible: strong and weak reproducibility across environments.

Following guidance in [309], assessments were first performed individually and then discussed collectively to ensure completeness and reduce bias. Table 2 reports representative outcomes: Vulnerability lifetime fails Well-defined condition due to difficulties in reliably establishing introduction time (cf. VCC pitfalls [188, 9]); Network maliciousness fails the Progressive condition (blacklist counts need not track security posture [317]); Worst case loss fails the Reproducible condition (estimation variability [33]); VEA-bility, instead, passes all conditions (system-level aggregation of CVSS with standardized procedure [276]).

Table 2: Example metrics that respect (V) or not (X) the three CSSM conditions: Well-defined (WD), Progressive (P), Reproducible (R).

Metric	Description	Ref	WD	P	R
Vulnerability lifetime	Measures how long it takes to patch a vulnerability since its introduction	[218]	X		
Network maliciousness metric	It estimates the fraction of blacklisted IP addresses in a network	[317]	V	X	
Worst-case loss	Maximum dollar value of the damage/loss that could be inflicted by malicious personnel via a compromised control system	[33]	V	V	X
VEA-bility	Aggregating scores from CVSS for the overall system, identifying all the (well-known) vulnerabilities on hosts	[276]	V	V	V

Deduplication and merging. Finally, equivalent metrics, such as those with the same objective/concept, and matching scope, result type, automation, measurement, construction, and type, are merged into unified entries. CSSM soundness is preserved under additive aggregation when

component metrics are sound [309], ensuring the final catalogue is non-redundant and validated for ICPS adoption.

2.3.2 Results and taxonomy of metrics for Industrial CPS

This section presents the security metrics that successfully passed all stages of the selection and validation process. The characteristics and properties of the validated metrics are outlined to provide a foundation for practical application in the ICPS domain.

Figure 3 illustrates the distribution of security metrics across the categories Vulnerability, Attack, Defense, and Situation, both before and after filtering and validation. This view highlights how category proportions evolve through successive steps.

The selection process began with 278 security metrics. Non-useful or invalid entries were removed at each step; results are reported in Fig. 4. The first CSSM condition (Well-Defined) accounts for most of the pruning, reflecting that many metrics, which later fail the Progressive and Reproducible tests, already lack a sufficiently well-defined formulation, primarily due to missing units, undefined data sources, or non-actionable definitions.

After validation, a total of 32 security metrics constitutes the final set. To assess coverage, the contribution to Confidentiality, Integrity, and Availability (CIA) was examined. Specifically, 87.5% of the metrics cover all three CIA aspects, while the remaining 12.5% cover only one aspect. Table 3 reports the complete list and CIA coverage. These metrics meet the specified criteria and are positioned as reliable indicators of security within the scope of this study.

A final perspective is provided in Fig. 5. The initial set exhibited an imbalance between static and dynamic metrics; as filtering and validation progress, the difference narrows, approaching an almost balanced partition.

```
1: procedure Obtaining-Metrics
2:                                                                    ▷ collection
3:     fullMetricsSet ← analyzeLiterature()
4:     fullLength ← length of fullMetricsSet
5:                                                                    ▷ classification
6:     N ← 0
7:     while N ≤ fullLength do
8:         classify(fullMetricsSet[N])
9:         N ← N + 1
10:    end while
11:                                                                   ▷ filtering
12:    properties ← domainAnalysis()
13:    N ← 0
14:    reductionSet1 ← ∅
15:    while N ≤ fullLength do
16:        if fullMetricsSet[N] respects properties then
17:            reductionSet1.add(fullMetricsSet[N])
18:        end if
19:        N ← N + 1
20:    end while
21:                                                                   ▷ validation
22:    firstLength ← length of reductionSet1
23:    N ← 0
24:    reductionSet2 ← ∅
25:    while N ≤ firstLength do
26:        if reductionSet1[N] respects CSSM then
27:            reductionSet2.add(reductionSet1[N])
28:        end if
29:        N ← N + 1
30:    end while
31:                                                                   ▷ merge
32:    secondLength ← length of reductionSet2
33:    Ni ← 0
34:    Nj ← 0
35:    finalSet ← ∅
36:    while Ni ≤ secondLength do
37:        while Nj ≤ secondLength do
38:            if i ≠ j then
39:                if reductionSet2[Ni] is equivalent to reductionSet2[Nj] then
40:                    mergedMetric ← mergeMetrics(reductionSet2[Ni], reductionSet2[Nj])
41:                    finalSet.add(mergedMetric)
42:                end if
43:            end if
44:            Nj ← Nj + 1
45:        end while
46:        Nj ← 0
47:        Ni ← Ni + 1
48:    end while
49: end procedure
```

Figure 2: Algorithm to obtain the filtered and validated set of security metrics.

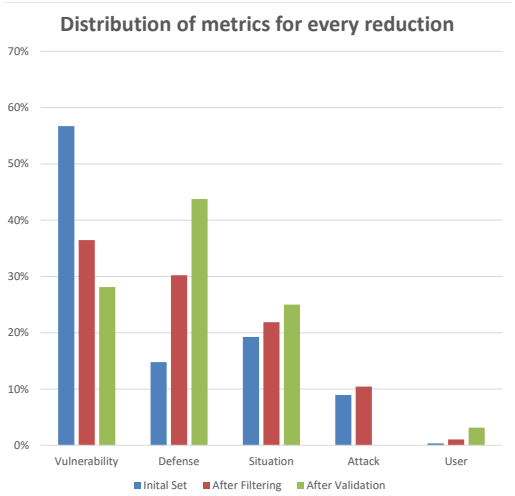


Figure 3: Percentage of security metrics by category after each step of the filtering and validation procedure.

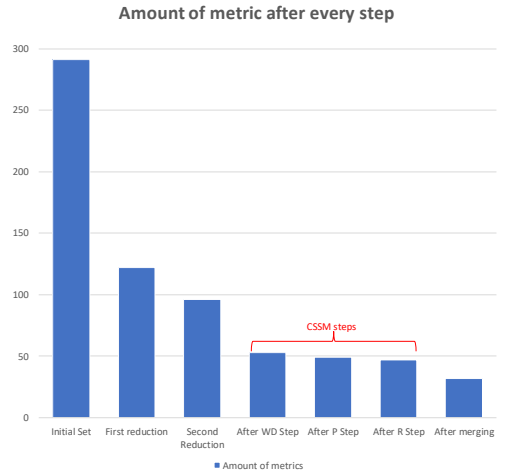


Figure 4: Number of security metrics after each step of CSSM-based filtering and validation (Well-Defined, Progressive, Reproducible).

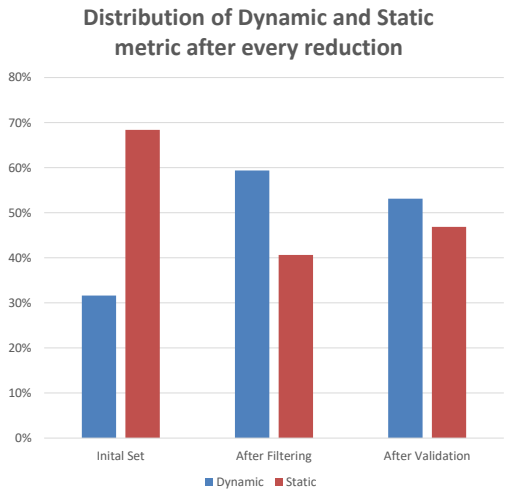


Figure 5: Percentage of dynamic and static security metrics after each step of the filtering and validation procedure.

2.3.3 Discussion and limitations

The effort aimed to assemble a comprehensive, validated set of security metrics for ICPS. The systematic methodology relies on published literature for metric collection; consequently, completeness is bounded by the breadth and depth of existing work. In addition, the resulting set may not fully capture emerging threats or future changes in the ICPS landscape. The domain is inherently dynamic, shaped by technological evolution and by the interconnection of systems with distinct vulnerabilities and constraints.

Classifying security metrics requires interpreting and applying criteria to assign them to types. Despite the use of explicit criteria and careful review of definitions, this step remains susceptible to subjectivity. Different assessors may interpret the criteria differently, resulting in variations in categorization. This does not undermine the methodology; rather, it promotes transparency in the classification process and acknowledges its interpretative nature. Future refinements of the criteria and consensus-

building among experts could help reduce subjectivity and strengthen reproducibility.

The coverage of the resulting metrics primarily concerns technical aspects. Yet, the deployment and operation of ICPS, and of CPS more broadly, encompass broader considerations, including social dimensions, the protection of fundamental rights (e.g., privacy), ethical issues, physical safety, and the integration of threat intelligence. An additional observation is that only 3% of the final set fall under the User category, underscoring the relatively limited attention that user-related issues receive in security evaluations, despite the user often being identified as the weakest link in cybersecurity [131].

Table 3: The list of the final set of security metrics that were obtained as a result of the collection, filtering, and validation for the ICPS domain.

Name	Scope	Result	Auto	Measure	Construction	Type	Ref.	C	I	A
Attack Impact	network	qualitative	manual	static	model	vulnerability	[145]	X	X	X
Attack surface	network	quantitative	auto	static	model	vulnerability	[174]	X	X	X
Component Test Count	device	quantitative	auto	dynamic	measure	situation	[33]	X	X	X
Cost metric	network	quantitative	auto	dynamic	measure	defense	[218]	X	X	X
d1-Diversity	network	quantitative	auto	static	model	defense	[319]	X	X	X
Data transmission exposure	network	quantitative	auto	dynamic	measure	situation	[33]	X		
Defense depth	network	quantitative	auto	static	model	vulnerability	[33]	X	X	X
Detection mechanism deficiency count	system	quantitative	auto	static	measure	defense	[33]	X	X	X
Historically exploited vulns metric	device	quantitative	auto	static	measure	vulnerability	[218]	X	X	X
Incident rate	network	quantitative	auto	static	measure	situation	[218]	X	X	X
Intrusion detection capability metric	network	quantitative	auto	dynamic	measure	defense	[101]	X	X	X
k-zero-day-safety metric	system	quantitative	auto	static	model	vulnerability	[289]	X	X	X
Mean of Attack Path Lengths	network	quantitative	auto	static	model	defense	[230]	X	X	X
Mean Effort To Failure (METF)	device	quantitative	manual	dynamic	model	situation	[230]	X	X	X
Mean-time-to-compromise (MTTC)	network	quantitative	manual	dynamic	model	vulnerability	[218]	X	X	X
Median of Path Lengths	network	quantitative	auto	static	model	defense	[230]	X	X	X
Minimum Password Strength	user	quantitative	auto	dynamic	measure	user	[33]	X		
Moving Target Defense evaluation	network	qualitative	manual	dynamic	model	defense	[249]	X	X	X
Network Compromise Percentage	network	quantitative	auto	dynamic	model	vulnerability	[230]	X	X	X
Number of Attack Paths	network	quantitative	auto	static	model	defense	[230]	X	X	X
Penetration resistance	system	qualitative	manual	dynamic	model	defense	[218]	X	X	X
Reachability count	network	quantitative	auto	static	model	situation	[33]	X	X	X
Reaction time metric	network	quantitative	auto	dynamic	measure	defense	[218]	X	X	X
Relative effectiveness	network	qualitative	manual	dynamic	model	defense	[230]	X	X	X
Restoration time	system	quantitative	manual	static	model	defense	[33]			X
Return on Investment	system	quantitative	manual	static	model	situation	[218]	X	X	X
Rogue Change Days	system	quantitative	auto	dynamic	measure	situation	[33]	X	X	X
Root privilege count	user	quantitative	auto	dynamic	measure	situation	[33]	X	X	X
SDPL and MoPL	network	quantitative	auto	static	model	defense	[230]	X	X	X
Side-channel Vuln Factor	device	quantitative	manual	dynamic	model	vulnerability	[56]	X		
VEA-bility	network	quantitative	auto	dynamic	model	defense	[276]	X	X	X
Vulnerable Host Percentage	network	quantitative	manual	dynamic	model	vulnerability	[33]	X	X	X

2.4 Open challenges

The systematic study yields a multi-stage methodology, from broad collection to classification, domain-aware filtering, and CSSM-based validation, resulting in a consolidated set of ICPS-specific security metrics that spans a wide range of technical concerns [309, 3, 245]. Despite these results, important challenges remain. System-of-Systems composition is still weakly represented: dependencies, side effects, and emergent dynamics across layers receive limited attention, even though they shape real risk in interconnected plants. Coverage is skewed toward technical artefacts, while user-facing aspects and operational routines appear only marginally, despite their role in incident causation and recovery. Finally, the diversity of industrial protocols, legacy equipment, and timing constraints complicates the direct comparison of metric values across sites, unless the context is modeled explicitly.

2.4.1 Context-awareness and scalability

ICPS operates across heterogeneous OT/IT stacks, legacy devices, real-time constraints, and diverse industrial protocols [297, 318, 53]. A metric that is well defined in one plant or vendor ecosystem may require re-scoping or re-calibration in another; domain tailoring improves relevance [169, 219] but complicates comparability and benchmarking across sites [218]. Feasible data collection at scale must respect operational costs and timing constraints while preserving measurability and automation properties [3, 245]. Even when CSSM conditions are met, local process dynamics and maintenance practices introduce variability that challenges strong or weak reproducibility [309].

Scalability pressures also arise in the measurement pipeline itself: aggregating over large attack graphs, spanning Enterprise, DMZ, Manufacturing zones, or federating across distributed plants stresses computation, storage, and normalization. Hierarchical collection and placement-aware sensing mitigate part of the burden, yet end-to-end scalability ultimately depends on architectural choices. This observation motivates the architectural mechanisms developed in the next chapter, where programmable

data planes, orchestrators, and trustworthy coordination substrates efficiently and consistently embed metrics.

2.4.2 Integration with automated decision-making

Metrics deliver value when they close the loop from sensing to control. Moving beyond observability requires mapping measurements to policies, orchestrating responses, and maintaining safety and stability in the face of uncertainty. Soundness conditions, such as CSSM, validate the metric itself [309], but do not guarantee that the induced control logic will be effective or safe. Thresholds and guards require local calibration and continuous revalidation to prevent drift or proxy effects [3, 245]. Most importantly, assessing whether a metric improves security is difficult outside a complete architecture that supports repeatable actuation paths and feedback; applicability and utility are best demonstrated within concrete deployments or faithful testbeds that exercise the full sense–decide–act loop. Metrics should be treated as first-class control variables, measured in-network or at endpoints, attested and shared when needed, and consumed by controllers to enact timely responses.

2.4.3 Authentication metrics: a missing layer

The analysis surfaced a specific blind spot. None of the authentication-related candidates made it to the final set. One metric tracked the number of authentications required before the exploit failed the Well-Defined condition, as establishing a consistent introduction time and exploit preconditions was not defined. Another metric that enumerated session-handling requirements failed applicability for ICPS, as it lacked a clear operational mapping to OT contexts and device constraints. As a result, authentication, identification, and authorization remain under-metricated for industrial environments. This absence is notable because identity and session assurances underpin any trustworthy posture in networks where cyber events can propagate to the physical layer.

2.4.4 Implications

Four implications follow. First, ICPS require authentication and authorization metrics that align with device capabilities, protocol behavior, and real-time constraints, and that remain measurable without intrusive instrumentation. Second, trustworthy transport and validation of metric signals are necessary to prevent tampering and to support cross-site comparability. Third, automated controllers need canonical, auditable formats for ingesting metrics and enacting decisions in a way that is explainable, reversible, and safe. Fourth, measurement must extend to the human dimension: user-oriented metrics are needed to capture operator workload, procedural adherence, error propensity, and security culture signals, collected ethically and with minimal disruption, so that sociotechnical risk becomes observable and improvable.

The next chapters build on these observations by exploring architectural mechanisms that can embed metric collection and validation into the network fabric, coordinate authentication mechanisms without relying on fragile central points, and consider the human factors. The aim is to transform metrics from static artefacts into operational primitives that support scalable assurance in heterogeneous industrial settings.

Chapter 3

Architectures

3.1 The link between metrics and architectures

Security metrics become effective when embedded into the control fabric of a system, influencing configuration, orchestration, and recovery. This section outlines how run-time measurements are transformed into enforceable decisions and which architectural properties, i.e., distribution, programmability, and verifiability, make that transformation scalable and dependable in CPS and ICPS. The next subsections provide the conceptual bridge; the following chapter instantiates these ideas in three complementary designs.

From measurement to enforcement

Moving from observation to action requires a well-defined loop of sensing, interpretation, and control [329]. In operational terms, measurements are mapped to policy constructs (e.g., thresholds, guards, weights), composed when multiple indicators are relevant, and consumed along placement-aware enforcement paths at endpoints, in the network, and within controllers. Stability, rollback capability, and the cost of false decisions are integral to these paths. Feedback closes the loop by checking whether actions improve the targeted posture and by recalibrating parameters as operating conditions evolve. The practical consequence is that the testa-

bility of a metric depends on the execution substrate that consumes it: effectiveness is demonstrated where sensing, decision, and actuation are exercised together [329].

The role of distributed and programmable infrastructures

Architectural support determines whether metrics can be collected and enforced at the required scale and speed. Three capabilities are central:

- Programmable data planes and control: in-network telemetry and match–action pipelines enable timely, low-overhead collection and first-line reactions close to traffic sources and sinks [255], while software-defined control provides a global view for correlating indicators and orchestrating changes across zones with real-time constraints [140].

- Distributed coordination substrates: when decisions rely on shared state (e.g., policy status, identity/credential posture, trust attributes), tamper-evident and auditable coordination mechanisms support consistent dissemination and verification among distinct components or administrative domains [274].

- Orchestration and lifecycle integration: controllers and configuration managers turn metric-driven policies into durable outcomes; exposing metrics as first-class control variables (schemas, units, update semantics) enables staged responses, safe fallbacks, and post-incident auditing [312].

Together, these capabilities turn metrics from static indicators into operational levers. The next chapter instantiates this linkage in three steps: first, metric-guided monitoring and response built around attack-graph reasoning and programmable planes; second, identity- and policy-aware coordination for industrial middleware; third, metric-weighted decision pipelines for distributed learning in untrusted environments.

3.2 GRAPH4: Security monitoring through attack graphs

Ubiquitous connectivity and reliance on digital infrastructure have amplified the impact of cyberattacks, while anomaly detection has emerged as a crucial approach to identify deviations from normal behavior beyond signature-based recognition [2]. Yet anomalies are not always attacks, and making timely, accurate decisions requires automated pipelines that weigh evidence and trigger proportionate responses. In networked CPS and ICPS, this calls for measurement signals that can be acted upon with minimal delay and overhead.

Software-Defined Networking (SDN) offers a natural control fabric for collecting indicators and orchestrating responses, thanks to its centralised control and global visibility [187]. However, naively computing rich metrics in the controller can introduce non-negligible overhead and energy cost, especially in large or time-critical environments [218, 198]. Offloading pacosts, especially in large or time-critical environments [021tracking, 218], but indiscriminate in-network computation risks burdening switches and increasing forwarding delay [83].

The proposed architecture, called GRAPH4, addresses this trade-off by selectively placing in-network monitoring where it matters most. Attack graphs are derived in the control plane from the active topology and configuration to identify nodes and paths that are vulnerable to specific attack classes. Only those portions of the network receive targeted monitoring rules in the programmable data plane; when local deviations are detected, concise alerts are raised to the controller for correlation and response. This focused strategy concentrates computation where it yields the highest value, reduces overall overhead, and supports prompt mitigation.

Programmable data planes enable in-network telemetry and first-line reactions with bounded overhead. In the following, these capabilities are combined with attack-graph reasoning to place monitoring exactly where risk concentrates, so that only the relevant portions of the network execute metric collection and local checks.

3.2.1 Background

P4

P4 [30] is an open-source programming language that controls data plane packet processing. As depicted in Figure 6, a P4-enabled switch differs from traditional ones in two aspects: first, the switch functions are not hardwired but specified by a P4 program; second, control and data plane communication takes place through a fixed-function device channel, but the data plane APIs are established by the P4 program. The Southbound Interface APIs, which expose the specific features and protocols supported by the data plane, are built using the specifications provided by P4Runtime [52] for abstracting the hardware interfaces. P4's primary goals are as follows:

- Reconfigurability: the packet parsing logic and processing rules can be dynamically installed and updated by the controller.
- Protocol independence: by giving rule names, key types, and typed match+action tables in the header fields, the controller can determine how to process header fields, freeing the switch from fixed actions taken on standard packet formats.
- Target independence: the P4 code is completely portable across all targets. The P4 compiler is tasked with translating program features that make use of target-specific capabilities.

P4 is designed around an abstract model that explains the traffic forwarding of the switch through match+action steps that are organized in series, parallel, or both. The parser, which extracts header information and serves as a programmable interpreter of supported protocols, first handles inbound packets. The match-action tables, which determine the egress port and queue for the packet, then receive the extracted header data. The packet may be sent, replicated, dropped, or cause flow control depending on the ingress processing. A P4 program defines the following elements to express the behavior of the data plane:

1. Header types: packet header definitions, i.e., the set of fields and their sizes.

2. Parsers: finite-state machines that map packets into headers and metadata.

3. Tables: data structures defining matching fields and actions applied to them.

4. Actions: code fragments that describe packet manipulation and can consider external data, supplied by the control plane at runtime.

5. Match–action units: elements that construct lookup keys from packet fields' metadata and use them to find the right action and execute it.

6. Control flows: imperative blocks that describe packet processing on a target using the data-dependent sequence of match–action unit invocations.

Thus, P4 enhances traditional SDNs (heavily reliant on the OpenFlow protocol [183]) by addressing two well-known shortcomings [90]: a significant communication overhead is generated between the data plane and the control plane, and significant processing capabilities are needed at the controller.

P4's unrivaled expressiveness provided a revolutionary new perspective on network programmability and monitoring. With P4 programmable switches, it is now possible for network operators to partially overcome such drawbacks. Programmable switches can execute a portion of network monitoring and security operations directly in their data plane pipeline, providing partially or fully processed information to the control plane. On the other hand, data plane programming has some intrinsic entry barriers, such as the need for specialized hardware and the effort required for network architects to become proficient in creating efficient and portable code [240]. The expressiveness limitations of OpenFlow-based SDNs do not allow for coding algorithms on the data plane, such as network entropy calculation, which is presented in the next section. In fact, OpenFlow only allows for a fixed set of configurations, while P4 can be used to code custom packet processing algorithms.

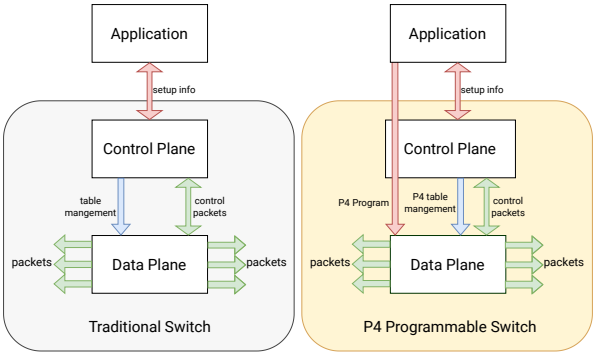


Figure 6: Comparing the architecture of a traditional switch and a P4 programmable switch. The forwarding behavior of the P4 programmable switch can be configured directly from the Application, setting the P4 program. In contrast, a legacy switch's application determines the behavior of the control plane, which then configures the data plane.

Network Traffic Entropy

Entropy was introduced as a measure of uncertainty by Shannon in 1948 [248]. Assuming that X is a dataset with a finite number n of independent symbols, represented by $x_1, x_2, ..., x_n$, with corresponding probabilities $p = p_1, p_2, ..., p_n$, the entropy of X is defined as follows:

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{3.1}$$

The entropy value ranges between 0 and $\log n$, reaching the upper bound when X has a uniform distribution. To make entropy values independent of the number of distinct symbols, entropy can be normalized to vary from 0 to 1 as follows:

$$H_N(X) = \frac{H(X)}{\log n} \tag{3.2}$$

From the above definitions, it is possible to define network traffic entropy as an indication of traffic distribution across the network [144]. Each network switch can evaluate the traffic entropy related to the network

flows that cross it in a given time interval T_{int} as:

$$H = -\sum_{i=1}^{n} \frac{f_i}{|S|_{tot}} \log_d \frac{f_i}{|S|_{tot}} \tag{3.3}$$

where f_i is the packet count of the incoming flow i, $|S|_{tot}$ is the total number of processed packets by the switch during T_{int}, n is the overall number of distinct flows, and d is the base of the logarithm. Network traffic entropy reaches its minimum value $H = 0$ when in the given time interval T_{int} all packets $|S|_{tot}$ belong to the same flow i, while it reaches its maximum value $H = \log_d n$ when each of the n flows transports only one packet.

3.2.2 Related Works

In literature, security metrics proposed for network analysis focus especially on attack graphs, which are graphical representations of potential attack paths and the various steps an attacker might take to compromise a target system or network. They provide a visual depiction of the relationships between different vulnerabilities, system components, and attack techniques that an adversary could use to achieve their objectives. Over time, various research works have exploited their potential. Wang et al. [287] proposed a general framework for designing network security metrics based on AGs. In [163], Lippmann et al. propose the Network Compromise Percentage Metric while evaluating the so-called defense-in-depth strategy using AGs. In [185], Mehta et al. compute a ranking of states in an AG based on the probability of attackers reaching each state during a random simulation; the PageRank algorithm is adapted for such a ranking; a key assumption made in this work is that attackers would progress along different paths in an AG in a random fashion. In this work [214] from Pamula et al., attack trees are replaced by attack trees with more advanced AGs and attack paths with attack scenarios. In [151], Leversage et al. proposed a mean time-to-compromise metric based on the predator state-space model. Homer et al. [121] address several important issues in calculating such metrics, including the dependencies between different attack sequences in an AG and cyclic structures in such

graphs. Poolsappasit et al. [221], instead, proposed Bayesian networks to quantify the chances of attacks and to develop a security mitigation and management plan as a metric.

More recent works start to consider vulnerabilities inside the nodes of the graph to make more accurate calculations: Wang et al. [288] instead of attempting to rank unknown vulnerabilities, propose a metric that counts how many such vulnerabilities would be required for compromising network assets: a larger count implies more security because the likelihood of having more unknown vulnerabilities available, applicable, and exploitable all at the same time will be significantly lower; Zhang et al. [319] presented a biodiversity-inspired metric based on the effective number of distinct resources, with the idea that the larger the diversity in components, the more secure it is the system because it is less probable that the same vulnerability is shared between different manufacturer; Ramos et al. [230] proposed metrics that consider the length and the number of attack paths resulting from the graph. An implementation of AGs very common in literature is MulVAL [208], a project that, since 2006, has proposed an end-to-end framework and reasoning system that produces AGs by conducting multihost, multistage vulnerability analysis on a network. A recent work by Stan et al. [259] that utilizes this tool on modern network architectures leverages MulVAL to model multiple attack techniques, including spoofing, man-in-the-middle, DDoS, and other types of attacks.

Metrics that pertain to this category require a high-level view of the network to produce and utilize an AG. Moreover, many of them are resource-intensive and cannot be computed on devices that must maintain a very low overhead, such as switches. A device, such as a switch, could implement only metrics of a measure type, but can still exploit the possibility of managing the data plane. The aforementioned studies all employ AGs as a means of obtaining a comprehensive overview of potential attack vectors within network environments, with the intent of deriving relevant metrics from such data. Distinguishing this research from prior work, the analytical scope is extended beyond metrics obtainable from AGs alone by integrating AG-derived indicators with lower-level data collected at

multiple vantage points. In addition, the investigation focuses specifically on networks employing DPP, leveraging their intrinsic capabilities; an emphasis that sets it apart from earlier efforts.

In contrast, considering low-level metrics deployable on network devices, the first contribution is the direct detection of DDoS attacks within the data plane of switches. However, the widely adopted data plane programming language, P4, lacks support for many arithmetic operations, limiting the straightforward implementation of advanced network monitoring functionalities required for DDoS detection. To address this limitation, Ding et al. [58] present two novel strategies for estimating flow cardinality and normalized network traffic entropy, which rely solely on P4-supported operations and ensure a low relative error. Building upon these contributions, the authors propose a DDoS detection strategy based on variations of normalized network traffic entropy. The results demonstrate comparable or higher detection accuracy compared to state-of-the-art solutions while being simpler and executed entirely in the data plane. Moreover, Gao et al. [83] propose an alternative solution, as mentioned in the introduction, which allows it to overcome the use of sketches and enables switches to alert the controller automatically upon detection of anomalies. They implement statistical checks in P4 by revisiting the definition and computation of statistical measures and collecting the techniques in a P4 library. Considering these works, the distinction inherent in this research lies in the opposite concept to what was previously discussed. Low-level metrics are consolidated with high-level metrics, and the resulting integrated analysis is applied coherently across distinct layers of security evaluation.

3.2.3 The GRAPH4 Architecture

Prior work has demonstrated that several security indicators can be computed directly in a P4 data plane to detect network attacks, such as traffic entropy or reachability-derived measures. The computation performed on forwarding devices, however, introduces overhead that may be unacceptable in environments with strict latency or traffic-shaping requirements. GRAPH4 addresses this tension by combining low-level measurements in

the data plane with high-level reasoning in the control plane. The controller employs Attack Graphs (AGs) to identify vulnerable regions of the topology and to place data-plane computations where they are most valuable, thereby concentrating resources around critical assets and minimizing unnecessary network-wide instrumentation.

Architecture Figure 7 illustrates the split between the control plane and the data plane. The controller maintains a model of the network that includes the topology, hosts, and their exposed services, as well as the attack vector under analysis. From these inputs, it generates an AG that enumerates feasible paths to vulnerable targets. The AG serves two purposes. First, it enables the computation of high-level metrics that summarise the security posture over the graph, such as the number of attack paths, the standard deviation or mode of path lengths [230], and the defence depth [33]. Second, it provides precise guidance on where to place low-level measurements and detection routines. In GRAPH4, the emphasis is on this second role: AGs are used to select the subset of switches that forward traffic to or from vulnerable hosts, and only these devices are instrumented.

Control plane The controller constructs the AG based on the current view of the network and the specified attack vector. Using the resulting graph, it identifies vulnerable hosts and the paths that reach them. This information is translated into placement decisions for data-plane code. Only switches that lie on vulnerable paths receive the programs that compute low-level indicators such as entropy or counters required for AG-related metrics. The outcome is a focused deployment that reduces overhead while preserving detection capability on the paths of interest.

Data plane On the forwarding devices, GRAPH4 follows the philosophy of in-network telemetry and line-rate analytics. The difference with network-wide approaches such as [83] is the scoping: monitoring is restricted to flows associated with the vulnerable hosts selected by the controller. Switches compute the required indicators and raise an alert when

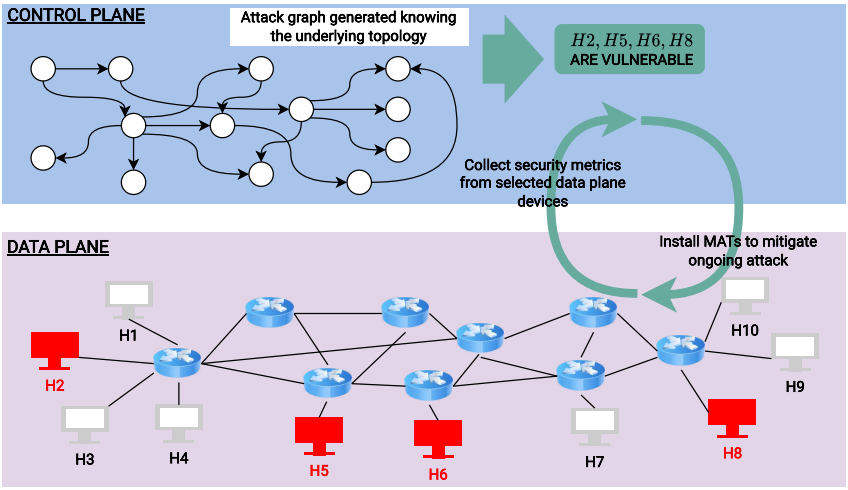


Figure 7: Interaction between control plane and data plane in GRAPH4. Hosts in red are identified by the AG as vulnerable; the controller instruments nearby switches to gather and evaluate metrics.

thresholds are exceeded or anomalous patterns are observed. Alerts are reported to the controller, which determines the appropriate response, such as adjusting rules, rate-limiting specific flows, or isolating segments of the topology.

Workflow Operation proceeds in two phases. During initialisation, the controller gathers topology and service information, generates the AG, identifies vulnerable hosts, selects the switches that handle their traffic, and installs the necessary match–action tables that implement the low-level detection logic. During run time, the instrumented switches monitor traffic associated with the protected assets and notify the controller when indicators suggest an ongoing attack. The controller then applies countermeasures and, if needed, updates both the AG and the placement to reflect the new state of the network.

GRAPH4 is agnostic to the specific indicator computed in the data plane and to the specific metric extracted from the AG. Any measure

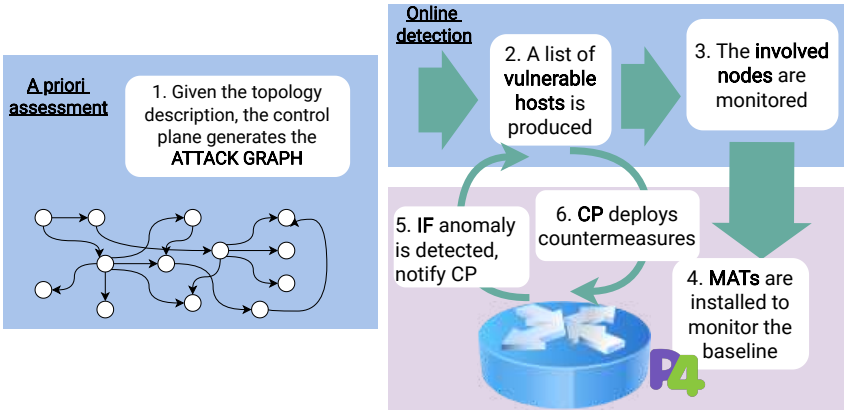


Figure 8: GRAPH4 workflow. The control plane (blue) generates the AG and places data-plane programs; the data plane (violet) monitors the selected traffic and reports anomalies for mitigation.

that benefits from local visibility can be implemented on selected switches, while model-based metrics over the AG provide the global view needed for placement and for high-level reporting. This cooperation between model-based and measurement-based metrics enhances resource allocation and reduces energy and computational usage, which is crucial for inline detection at scale.

3.2.4 Proof of Concept

This section presents a proof of concept (PoC) that demonstrates the feasibility and effectiveness of GRAPH4 in detecting distributed denial-of-service (DDoS) attacks while minimising monitoring overhead through targeted metric placement. The attack model consists of a volumetric flood directed at one or more hosts and traversing multiple network regions with the aim of exhausting bandwidth and isolating the targets. GRAPH4 combines high-level reasoning on the controller and low-level measurements in the data plane: the controller generates MulVAL AGs to locate vulnerable hosts and their corresponding paths, and selected switches compute a normalized traffic-entropy indicator (P4NEntropy)

[58] that triggers alerts on anomalies. When the entropy indicator falls below an adaptive threshold, the controller installs mitigation rules that block the malicious flow; the metric then returns to nominal values. In the emulation environment, the mean reaction time from detection to rule deployment was 2.5 s.

Low-level metric

P4NEntropy [58] estimates the normalised entropy of network traffic over configurable time windows (cf. Eq. 3.3), and is implemented in P4 despite the lack of native division, logarithm, floating-point operations, and loops. The P4DDoS application [58] was adapted for GRAPH4, allowing detection rules to be installed only for hosts designated as vulnerable by the AG. This restriction ensures that monitoring is focused on the traffic that matters in the current scenario. DDoS is signalled when the entropy value drops below an adaptive threshold. Figure 9 shows a representative run with a threshold of 0.5: once the threshold is crossed, the controller deploys match–action rules to block the flow, and the entropy returns to its typical value.

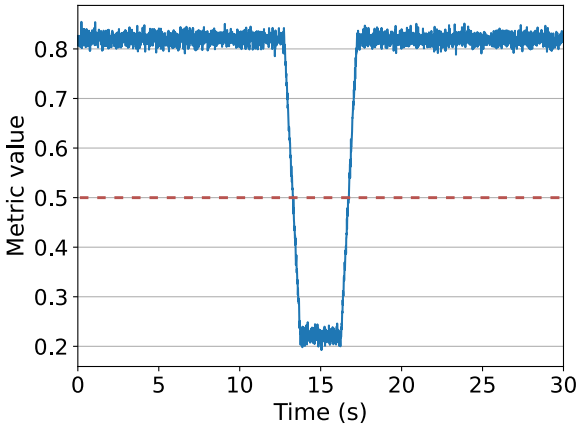


Figure 9: Normalised entropy (P4NEntropy) over time. When the metric drops below 0.5, a DDoS condition is detected and mitigation rules are installed; the metric then returns to nominal values.

AG cuts

The testbed is built in Mininet[1] with a single P4 switch (bmv2 behavioral model[2]) and a custom controller for the control plane [240]. The topology (Fig. 10) follows [83] with an additional subnet protected by a firewall; an external traffic source is attached to the P4 switch, which connects to three /24 subnets with six destination hosts each. Virtual link capacity is bounded by CPU resources, and the experiments ran on Ubuntu 20.04 LTS (14 GB RAM, 3 vCPU, KVM). To produce the AG, the extended MulVAL version [259] was used; the knowledge base is expressed in Datalog [46] via an input.P file declaring hosts, gateways, services, vulnerabilities, and the attack goal (Figs. 11–12). Hosts that appear along at least one feasible attack path are deemed vulnerable. Denoting by N_h the total number of hosts, the set splits into V vulnerable and S non-vulnerable hosts, with

$$N_h = V + S. \tag{3.4}$$

Switches compute the entropy indicator only for flows whose endpoints include a vulnerable host.

The effect of targeted placement is quantified by measuring the Packet Processing Time (PPT)[3] on the switch. Figure 21 reports PPT distributions for 10,000 packets under different fractions of monitored traffic. The distributions exhibit stable averages and limited sensitivity to the proportion of monitored packets; the single-packet PPT depends primarily on whether the packet triggers metric computation, rather than on the number of other packets monitored in the same interval. The total computation time on a switch, T_c, can therefore be modelled as proportional to the number of monitored flows (and thus to V) times the average PPT of monitored packets. This single-switch approximation is consistent with the analytical treatment in [83]. Figure 14 illustrates the relationship: without AG guidance, T_c grows with N_h (effectively $V = N_h$), whereas with GRAPH4 it is upper-bounded by V, yielding a reduction

[1] https://mininet.org/

[2] https://github.com/p4lang/behavioral-model

[3] PPT is computed as $t_{out} - t_{in}$ from Wireshark captures at ingress and egress interfaces.

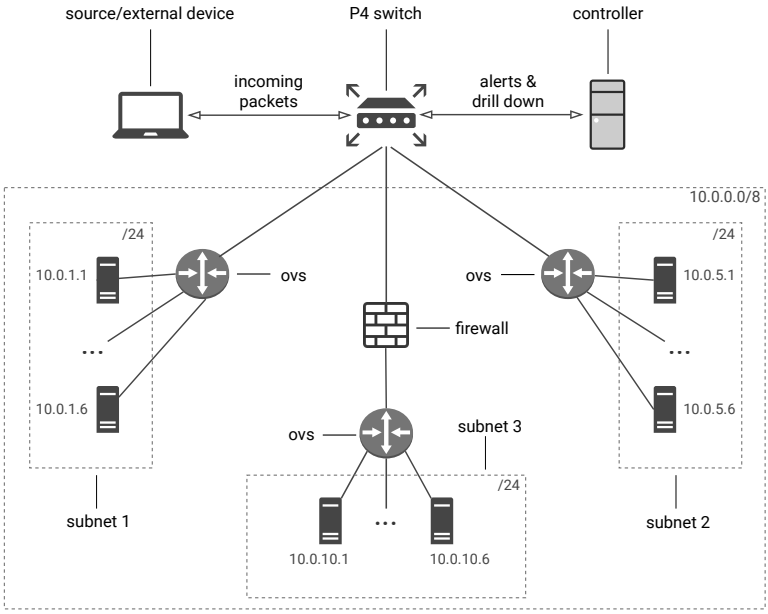


Figure 10: Emulated topology based on [83] with an additional protected subnet (10.0.10.0/24).

```
/* attacker info */
attackerLocated(internet).
malicious(attacker).

/* network topology */
isGateway(switchP4,subnetP4).
isGateway(ovs1,subnet1).
isGateway(ovs2,subnet2).
isGateway(ovs3,subnet3).

located(ovs1, subnetP4, ipSubnet).
located(ovs2, subnetP4, ipSubnet).
located(ovs3, subnetP4, ipSubnet).

located(host1, subnet1, ipSubnet).
/* repeat for every host of subnet 1, 2 and 3 */

hacl(internet, host1, tcp, 80).
/* repeat for every host */

/* active services */
networkService(host1, ssh, tcp, 80, _).
/* repeat for every host */

aclH(host1, _, _, host1, tcp, 80).
/* repeat for every host of subnet 1 and 2 */


/* vulnerability information */
vulHostDos(host1).
/* repeat for every VULNERABLE host */

attackGoal(dos(attacker,host1)).
```

Figure 11: Excerpt from input.P (MulVAL/Datalog) used to generate the AG. Repeated declarations are omitted for brevity.

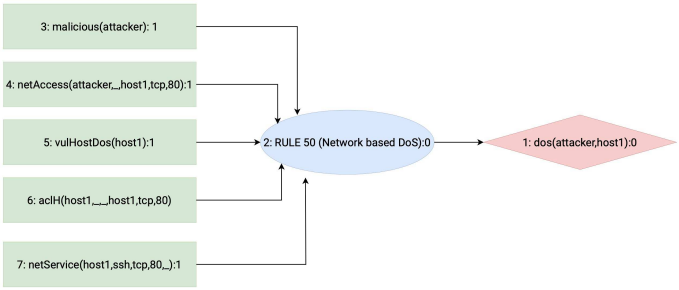


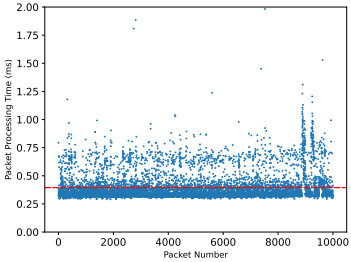
Figure 12: Attack Graph generated from the topology in Fig. 10 for one host in a vulnerable subnet. The full output contains analogous graphs for all relevant hosts.

Δ_i whenever only a subset of hosts is vulnerable. The linear trend is further supported by experiments with 50,000 packets per run and monitored fractions of 0%, 12.5%, 25%, 50%, and 100%, where the mean and variance of PPT scale with the monitored load (Fig. 15). Control-plane overhead remains negligible because the data plane emits compact alerts only when thresholds are crossed: a report with source and destination IPv4 addresses fits in 64 bits per event, unlike SDN mirroring approaches that forward bulk traffic to the controller.

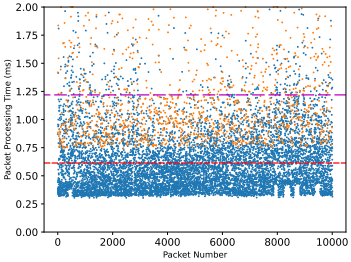
3.2.5 Limitations and outlook

GRAPH4 demonstrates that metric-driven monitoring can be selective and efficient when high-level reasoning on attack paths guides the placement of low-level, in-network measurements. The proof of concept shows real-time detection and mitigation of DDoS conditions with focused instrumentation, limited control-plane signalling, and a measurable reduction of switch load whenever only a subset of hosts is classified as vulnerable.

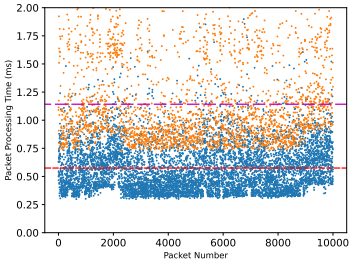
Two limitations deserve attention. The completeness of the Attack Graph depends on the quality of the inputs used by the controller, namely topology, exposed services, and associated vulnerabilities. Missing or inaccurate facts produce incomplete graphs and may hide feasible attack



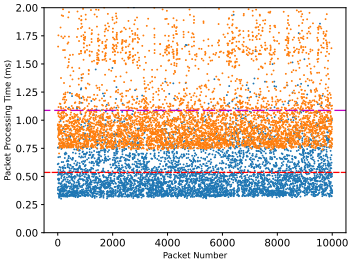
(a) 0% monitored



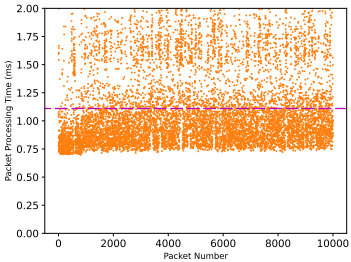
(b) 12.5%



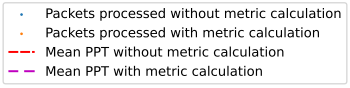
(c) 25%



(d) 50%



(e) 100%



(f) Legend

Figure 13: Packet Processing Time (PPT) on the switch for 10,000 packets under different fractions of monitored traffic. Blue: packets not used for metric computation; orange: packets used to compute the metric; red lines: respective averages.

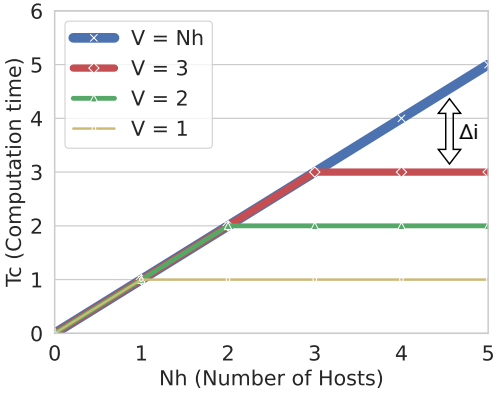


Figure 14: Computation time T_c on the switch. The blue curve corresponds to $V = N_h$ (no AG guidance). Other curves show cases with $V < N_h$, where the reduction Δ_i reflects the benefit of AG-guided placement.

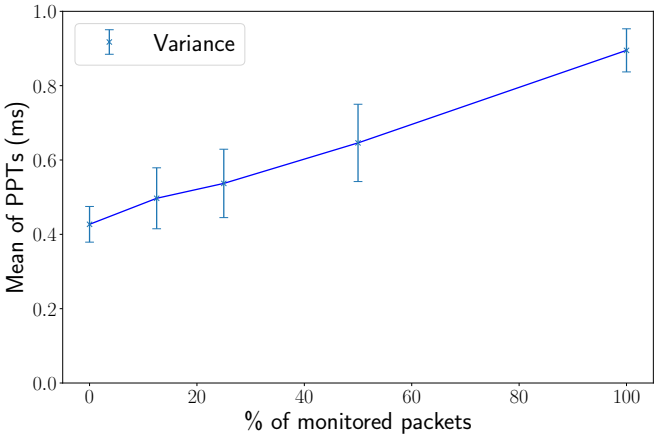


Figure 15: Mean and variance of single-switch PPTs across five scenarios (50,000 packets each) as the monitored fraction increases from 0% to 100%.

paths, creating a misleading sense of protection. In addition, the workflow assumes relative stability, whereas industrial environments can change frequently in topology and exposure. Keeping the graph aligned with reality and updating placement decisions with low latency is challenging; without timely adaptation, the model drifts and detection accuracy degrades.

Mitigation of these issues points to improved asset discovery, continuous configuration auditing, and rapid re-generation of the graph with corresponding re-deployment of data-plane probes. Further development includes quantifying end-to-end gains in multi-switch scenarios, exploring indicators beyond entropy while preserving explainability, and incorporating learning-based detectors with careful calibration and control.

3.3 PK-IOTA: Secure certificate distribution

3.3.1 Motivation and context

Industrial Control Systems (ICS) integrate cyber and physical components to operate critical infrastructures and production lines [127]. With Industry 4.0 and the Industrial Internet of Things (IIoT), machines, sensors, and actuators communicate across heterogeneous networks at scale. The OPC Unified Architecture (OPC UA) [200] has emerged as the de facto standard for interoperable industrial communications [139], offering built-in mechanisms for authenticity, integrity, and confidentiality; its security design has been positively assessed by the German Federal Office for Information Security [35].

Despite security-by-design, real deployments often fall short. Internet-wide measurements revealed that most reachable OPC UA endpoints are misconfigured, often lacking access control, disabled security features, deprecated cryptography, or certificate reuse [55]. A systematic assessment of 48 artifacts further highlighted gaps in certificate management and trust-list handling, enabling attacks such as forged data ingestion, eavesdropping, and parameter manipulation with direct physical consequences [70]. Two practical challenges emerge: managing certificates and trust lists at scale, and coping with inconsistent or partial security support

across multi-vendor estates.

The previous chapter (GRAPH4) established that validated security metrics can guide selective, real-time monitoring and mitigation. That perspective optimises where and how to measure and react. A complementary requirement is ensuring that only authenticated and authorized entities exchange data, with verifiable and timely certificate validation and revocation across heterogeneous and legacy installations: as shown in Chapter 2, the current State of the art in security metrics does not address this condition. Thus, an architecture called PK-IOTA will be presented, which addresses this need by automating OPC-UA credential handling and relocating critical validation logic to the network, thereby ensuring trustworthy enforcement complements metric-driven monitoring.

3.3.2 Challenges in OPC UA deployments

OPC UA [173] is a machine-to-machine communication standard designed to ensure secure, reliable, and platform-independent exchange of data between devices and systems in Industry 4.0 settings. The latest OPC UA specification 1.05 [200], which comprises 24 parts that focus on various features of the protocol, was released in 2022. It establishes two communication strategies based on the client-server model and the publish-subscribe model. The two main features of OPC UA are its information model-based architecture and integrated secure communication by design [35]. The information model architecture enables platform-independent communication between industrial devices of different manufacturers: device functions, sensor values, and other variables, as well as their relationships, are represented by OPC UA servers as a set of nodes in an address space, from which clients can dynamically request the execution of functions or access data of variables. Core Information Models are already defined as part of the OPC UA specification, but vendors can extend them to add information about their products [204].

To establish a communication channel, an OPC UA client and an OPC UA server must first perform a security handshake consisting of four steps, as shown in Figure 16. During the security handshake, the client and server authenticate themselves using their OPC UA Application Instance

Certificate, a type of X.509-compliant digital certificate. These certificates are mutually exchanged and verified between devices when initiating a secure communication session. To verify received certificates, each device maintains a Trust List. This list consists of certificates that are trusted by the device. A received certificate is considered valid if it is either in the trust list itself or part of a certificate chain that has an anchor in the trust list.

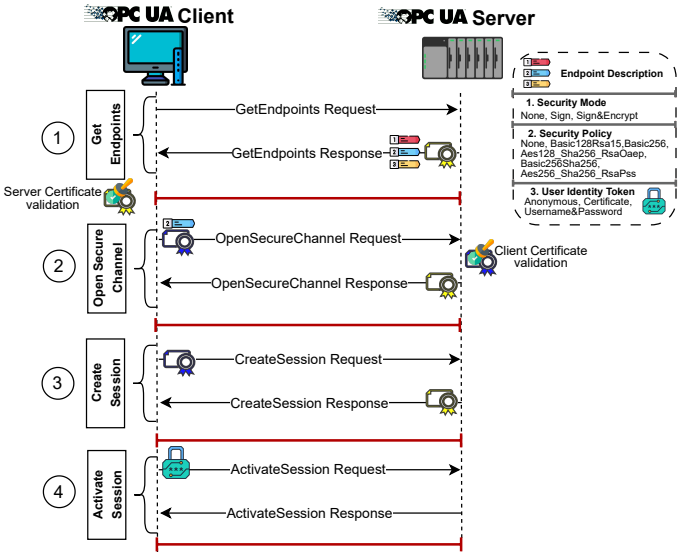


Figure 16: Connection establishment handshake in OPC UA, which comprises four steps: GetEndpoints, OpenSecureChannel, CreateSession, ActivateSession. In the GetEndpoints response, the OPC UA server provides a list of endpoints, each one specifying a Security Mode, Security Policy, and User Identity Token. The OPC UA client selects one endpoint to connect to.

In the first step, the client must select an Endpoint, i.e., a configuration option that specifies how a connection to the server can be established. Therefore, it sends a GetEndpoints request to the server in order to obtain the descriptions of the existing Session Endpoints. Each Session Endpoint is defined by a Security Mode, a Security Policy, and

the supported User Identity Token(s). The Security Mode defines how messages are exchanged between parties to achieve authentication, confidentiality, and integrity. Available Security Modes are None, Sign, and SignAndEncrypt. These modes offer unprotected communication (None), authenticated communication (Sign), and authenticated and confidential communication (SignAndEncrypt). Security Policies define the cryptographic primitives and their parameters to implement the various security modes. Finally, the UserIdentityToken defines the supported user authentication methods for an endpoint: Anonymous (no user authentication), Username&Password, and Certificate.

Upon receiving the GetEndpoints response, the client promptly selects a Session Endpoint and validates the server's Application Instance Certificate. If the certificate is deemed trustworthy, the client then proceeds to send an OpenSecureChannel request to the selected Session Endpoint as the second step. When the Security Mode is set to None, the OpenSecureChannel message remains unsecured. If Sign is chosen, the message is signed with the client's Application Instance Certificate's private key. For SignAndEncrypt, the message is additionally encrypted with the server's Application Instance Certificate's public key, as specified in the Security Policy, which outlines the algorithms for signing and encryption. Once the server receives the OpenSecureChannel request, it first validates the client's Application Instance Certificate. The certificate is provided in an unencrypted part of the message and can thereby be read by the server. If the client's certificate is considered trustworthy by the server, then the message has to be interpreted according to the Security Policy and the Security Mode, i.e. decrypted with the associated private key of the server's Application Instance Certificate, and the signature of the message is verified with the public key of the client's Application Instance Certificate. The server then sends an equally secured response to this request, whereupon the client performs equivalent validations on the server certificate. The OpenSecureChannel Request and Response messages contain a nonce from the client and the server, respectively, in each message. These nonces are used to compute the symmetric signing and encryption keys for the session [205]. The Secure Channel has a finite lifetime to resist long-term

attacks. After this lifetime has expired, a renewal of the channel must be initiated by repeating the steps described above. Nevertheless, this renewal process is transparent to the Session that is created on top of a Secure Channel.

The third step involves creating a Session on top of the previously established Secure Channel. In this phase, the client and server must verify possession of their respective Application Instance Certificates through a challenge-response mechanism. In fact, the CreateSession request sent by the client to the server also contains the client certificate and a nonce. First, the server verifies that the client certificate matches the one used for the Secure Channel, and then signs the client nonce with its private key to prove possession of its Application Instance Certificate. Subsequently, in the CreateSession response, the server provides its certificate, the signed client's nonce, a new nonce, and two values that uniquely identify the Session. The first value is the sessionId, which is used to identify the Session in the audit logs and in the server's Address Space. The second is the authenticationToken, which is used to associate an incoming request with a Session.

The fourth and final step is session activation via the ActivateSession service, necessary to specify the identity of the user associated with the session. This Service request shall be issued by the Client before any other Service request. Failure to do so shall cause the Server to close the Session. In the ActivateSession request, the client proves that it is the same application that called the CreateSession service by signing the server nonce and the server certificate obtained in the CreateSession response with the private key associated with the client certificate specified in the CreateSession request. Note that the CreateSession and ActivateSession requests and responses are signed and encrypted using the symmetric keys derived in the OpenSecureChannel phase according to the selected Security Mode and Security Policy.

While OPC UA's protocol design is inherently secure [35], the range of configuration options available can greatly affect its overall security. Official guidelines aim to address these concerns [206]. Firstly, communication security should always be enabled, ensuring that messages

are both signed and encrypted whenever possible. Additionally, anonymous authentication should be prohibited. Finally, only three of the six available security policies (Aes128_Sha256_RsaOaep, Basic256Sha256, Aes256_Sha256_RsaPss) should be employed, as one offers no security and two have been deprecated due to their reliance on SHA-1.

3.3.3 Problem Statement

The majority of the security goals of OPC UA are achieved by relying on the Secure Channel [186]: depending on the selected Security Mode, it guarantees the confidentiality and integrity of the data exchanged in a session by encrypting the transmitted messages and applying a digital signature to them. Furthermore, when establishing secure communication, OPC UA clients and servers mutually authenticate themselves by exchanging their Application Instance Certificates and generating a session key for subsequent secure communication.

Similar to many other network protocols, OPC UA uses X.509 [29] compliant certificates for authentication. In OPC UA, it is possible to use either certificates issued by a trusted Certification Authority or self-signed certificates. In the latter case, the private key associated with the public key of the new certificate is used to sign the certificate, which implies that the entity generating the certificate is the Certification Authority. To verify received certificates, each client and server maintains a set of trustworthy certificates in its so-called trust list. A server accepts connections from a client in case the client can authenticate itself with a certificate that the server can successfully verify based on its local trust list, and vice versa [137].

A common approach for managing these certificates involves manually installing them into the trust lists of applications and devices. During the initial connection between two devices, the connection is blocked because a trust relationship is absent. Nevertheless, certificates from these attempts are stored, allowing an administrator to manually configure the devices and establish trust for these certificates, enabling future connections. This method, however, is not feasible for large OPC UA networks [136], requires significant manual effort, and is thus prone to human error.

Fortunately, OPC UA defines methods for the automated management of certificates [201], which include functionalities designed for standard OPC UA applications (both servers and clients), as well as capabilities tailored for a specialized server responsible for certificate management, referred to as the Global Discovery Server (GDS). The available methods enable applications to obtain certificates and update their trust lists, either through a process performed by the application itself (Pull Model) or on the initiative of the GDS (Push Model).

However, recent works [70, 55] have revealed that significant challenges still exist in setting up secure OPC UA deployments in practice. [55] shows that 92% of OPC UA deployments reachable over the Internet are improperly configured. In particular, 24% of these servers completely disable communication security, while another 25% rely on outdated cryptographic methods, such as SHA-1. Additionally, 35% of the systems improperly implement otherwise secure configurations by reusing security-critical certificates across devices in multiple systems within different networks, making these systems susceptible to impersonation and eavesdropping. Finally, 44% of servers allow unauthenticated users to read, write, and execute functions on industrial devices. The authors affirm that secure protocols are no guarantee for secure deployments: they underscore the need to reduce configuration complexity in OPC UA deployments and demand secure defaults for all configuration options, eventually transitioning from security by design to security by default.

Diving deeper, [70] systematically investigated the security of 48 real-world OPC UA artifacts (22 products from vendors and 16 libraries, of which 11 provide OPC UA servers and 15 provide OPC UA clients), showing that in practice, many of them have missing support for security features of the protocol. Specifically, only 10 artifacts (20.8% out of 48) correctly implement the certificate management, while 7 artifacts do not support security features at all, and the remaining 31 (64.6% of the total) exhibit issues or errors in the trust list management. Moreover, the features the GDS offers to manage certificates are supported by only 5 artifacts. The authors demonstrate that inconsistent or nonexistent support for managing trust lists and integrating with the GDS can lead to

misconfigurations, rendering OPC UA deployments vulnerable to Rogue Server, Rogue Client, and Middleperson attacks. In the Rogue Server attack, the attacker feeds incorrect information to OPC UA clients while a Rogue Client eavesdrops and changes values, which can directly alter the physical process. Finally, a Middleperson attacker aims to establish himself as a man-in-the-middle (MitM) in the connection between the OPC UA client and server, intercepting and manipulating all communications between both.

Therefore, while the security-by-design of the OPC UA standard is certified by the German Federal Office for Information Security (BSI) [35], it is evident that practical deployments of OPC UA often fail to meet these security standards due to misconfigurations, lack of support for key security features, and the complexities associated with managing certificates and trust lists in real-world ICS environments. This security gap between protocol design and practical deployments highlights the need for automated configuration processes to ensure that OPC UA systems can consistently achieve the intended level of security in operational settings.

3.3.4 Design Goals & Challenges

To close the security gaps outlined in the previous Section, this work aims to facilitate the secure deployment and configuration of OPC UA systems in practical ICS settings. To achieve this, the following key challenges are directly tackled:

- The complexity of OPC UA certificate and trust list management in real-world deployments: the manual management of certificates and trust lists in large-scale OPC UA networks is impractical and highly error-prone. Although current OPC UA setups often rely on GDS for certificate management, many ICS environments still lack GDS support, and many OPC UA industrial devices lack compatibility with GDS features [70]. At the moment of the writing, existing commercial GDS solutions are also limited in number, and none are currently certified by the OPC Foundation[4].

[4]https://opcfoundation.org/products/?category=18

- Inconsistent support for OPC UA security features: OPC UA provides extensive security features, but these are inconsistently implemented across devices, with many lacking critical support for certificate management, trust list handling, or GDS integration. Such inconsistency hinders the security of OPC UA deployments, as administrators cannot rely on uniform functionality across devices. Standardizing security configurations becomes challenging when devices exhibit varied support for fundamental security features.

By addressing these specific challenges, a scalable architecture with the following design goals is deployed:

- Automated certificate management: by leveraging a Distributed Ledger Technology (DLT) as a decentralized PKI, eliminating manual, error-prone certificates by handling and ensuring that trust relationships between OPC UA devices are accurately maintained in real-time.

- Independence from GDS: while GDS provides valuable certificate management functions, not all OPC UA environments support it. Therefore, this approach removes reliance on GDS by distributing OPC UA certificates over a DLT and validating certificates in the network data plane via programmable switches.

- Independence from device-level security feature support: by integrating in-network certificate validation via Data Plane Programmable (DPP) switches, PK-IOTA performs real-time certificate verification that operates independently of device-specific security features such as trust list management. This design ensures a uniform security layer even if individual OPC UA devices have limited security features. The approach relies solely on a common set of features supported by most OPC UA devices (85.4% of the artifacts analyzed by [70]). Deployed industrial OPC UA devices are assumed to support the Sign&Encrypt security mode, a non-deprecated security policy, and either Certificate or Username&Password as user identity tokens.

- Security requirements: The system is designed to be robust against Rogue Server, Rogue Client, and Middleperson attacks by mandating certificate validation at the data plane, thereby blocking unauthorized devices before they access critical ICS components.

Data Plane Programmability (P4)

One of the key enablers of this approach is the adoption of P4 [31]. already introduced in Sec 3.3.4. By deploying P4-based programmable switches, OPC UA certificate validation is moved directly into the network path. Custom header parsing enables the inspection of OPC UA messages and the extraction of certificate data within the switch data plane, maintaining protocol independence and runtime control. Rather than relying on each endpoint device to maintain an up-to-date trust list, the network itself inspects connection requests and extracts certificates. Any OPC UA connection employing an untrusted or revoked certificate can be dropped immediately, effectively preventing unauthorized connections.

Decentralized certificate manager

This chapter introduces the necessity of a distributed certificate manager and analyzes possible approaches for its implementation. Beyond simple storage capabilities, such a manager must also communicate with all controllers in real-time, eliminating the need for continuous polling, which can be inefficient and prone to delays. This requires introducing a server-like structure that coordinates and manages certificates dynamically across various devices.

One possible solution to distributed certificate management could be a centralized system. This method already exists in the current OPC UA standard, which provides a centralized certificate authority. However, this approach introduces significant challenges. Centralized systems typically involve extensive human intervention, requiring administrators to handle certificate requests, revocations, and renewals [67]. This human element can lead to errors, delays, inefficiencies, and security incidents [217], making the system unsuitable for large-scale or highly dynamic industrial en-

vironments. Furthermore, centralized management presents a single point of failure (SPoF), which is particularly problematic in mission-critical industrial applications. For these reasons, an automated, decentralized approach to certificate management is highly desirable.

A promising alternative to the centralized approach is DLT, which inherently supports decentralized architectures. DLTs are decentralized systems that record data across a network of nodes to ensure consistency, transparency, and immutability. Each participant in the network maintains a copy of the ledger, and updates are validated through a consensus mechanism, ensuring all nodes agree on the state of the data. Blockchain, in particular, is a specific type of DLT that combines cryptography, consensus protocols, peer-to-peer networks, and smart contract technologies to create a decentralized and secure system for recording transactions. A blockchain operates as a chain of blocks, where each block contains a set of transactions and is cryptographically linked to the previous one, forming a continuous and tamper-proof ledger. The process of adding new blocks begins with transactions being proposed and disseminated across the network. Depending on the system, these transactions are grouped into a block by a participating node, commonly referred to as a miner or validator. A consensus mechanism takes action before the block is added to the chain to ensure agreement among nodes on the ledger's state and prevent attacks such as double-spending, which occurs when an attacker successfully spends the same digital tokens twice by creating conflicting transactions. Consensus can be implemented through various methods, such as Proof-of-Work (PoW), used in Bitcoin; Proof-of-Stake (PoS), used in Ethereum; or Delegated PoS, used in TRON. Once consensus is achieved, the cryptographic hash of the previous block is embedded in the new block, creating a secure link between them. This chaining ensures that altering a block would require modifying all subsequent blocks, which is computationally infeasible in most cases but still enables quick verification. The longest chain rule is applied to avoid forks, where the only valid chain considered is the one with the majority of blocks attached. These mechanisms, without the need for a central authority, maintain the integrity and consistency of the ledger while preventing unauthorized alterations.

These properties offer several advantages that make it a potential candidate for distributed certificate management in industrial contexts [254]. In fact, each node in the network holds a complete copy of the blockchain, meaning that even if one or several nodes fail, the certificate management process can continue uninterrupted. In addition, blockchain facilitates automatic synchronization across multiple controllers and clients for automated certificate management [142]. By utilizing smart contracts, which are self-executing contracts with the terms of the agreement directly written into code, blockchain can handle certificate lifecycle events such as expiration, renewal, or revocation without requiring manual intervention. This streamlines the process and reduces the likelihood of errors arising from manual handling. Moreover, as the number of devices or controllers increases, blockchain enables seamless integration without introducing additional management complexity, allowing the system to scale effectively. Additionally, using the blockchain solves some security issues naturally present in centralized CAs, such as the split-work attack, certificate revocation and validation problems, and trusted certificate/key store management problems [142].

However, blockchain faces significant challenges, including scalability and performance issues. Blockchain networks utilize consensus algorithms to reach a consensus on the validity of transactions and updates to the ledger. Standard methods include PoW. Many of these popular consensus mechanisms introduce significant inefficiencies, and using miners and validators that create one block at a time causes the entire infrastructure to struggle with throughput, taking up to ten minutes to craft a single block in Bitcoin[5]. Those problems make blockchain a complex choice for PKI despite its potential benefits [303].

For this reason, innovative solutions have been proposed to overcome the limitations of blockchain, primarily by developing different consensus protocols or validation mechanisms within the P2P network [302] and modifying the chain-like architecture. A promising project in this direction is IOTA, which will be introduced in the next chapter.

[5] https://bitinfocharts.com/

IOTA

"The Tangle" [222] introduced the mathematical foundations of the IOTA project. It focused on providing a distributed microtransaction infrastructure for the IoT domain, designed to be more energy-efficient and faster than classical blockchains. IOTA is not based upon a chain ledger but is based on a Direct Acyclic Graph (DAG) where each node represents a transaction and consensus is achieved through the confirmation process: each new transaction verifies two previous transactions, eliminating the need for PoW or PoS and making the system more scalable and efficient. This confirmation is based on checking that the two transactions are not conflicting and do not approve conflicting transactions to avoid double-spending. In classic blockchains, nodes validate transactions by checking digital signatures and ensuring sufficient balances before adding them to a block. In IOTA, transactions are validated through mutual confirmation, eliminating the need for block formation. This improves transaction speed by ensuring that at least partial peer confirmations occur quickly [310], thereby avoiding the bottleneck caused by many blockchain architectures and enabling asynchronous and parallel transactions. It should also be noted that the tangle may contain conflicting transactions; some will become orphaned, as subsequent nodes will not choose them for verification (due to the tip algorithm). Early research on IOTA identified delays in attaching transactions to the Tangle [102]. However, later simulations highlighted its superior performance over traditional blockchains in terms of transaction confirmation rates and computational requirements [32]. While Bitcoin averages 7 transactions per second (TPS) and Ethereum can handle around 15 TPS, IOTA demonstrates an almost linear relationship with the transaction arrival rate, making it possible to exceed 80 Confirmed TPS under certain conditions [72]. This is due to the almost linear relationship between transactions sent per second and throughput until it reaches the maximum load [60], keeping the latency constant and limiting CPU usage. Although IOTA has an innovative DAG-based architecture, it preserves almost all of the benefits of traditional blockchains, such as protection from tampering, immutability, the absence of a central

point of failure, and transparency [325], while also introducing enhanced scalability and efficiency: key improvements that were essential for addressing the limitations of traditional blockchain systems [54, 291]. These characteristics enable the designation of IOTA's technologies as a secure, efficient, and tamper-proof choice for implementing a distributed certificate manager.

3.3.5 Threat model

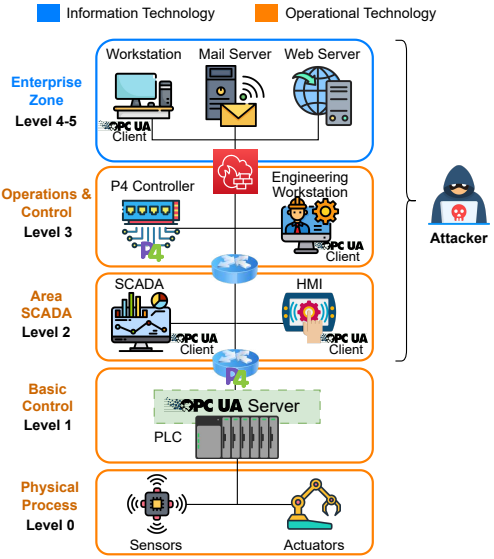


Figure 17: The Threat Model of this work is based on the ICS Purdue Reference Architecture. The attacker is assumed to have Dolev-Yao capabilities above level 1, but cannot tamper with the channel between the P4 controller and the P4 programmable switch.

The system and threat model behind this work are illustrated through the Purdue Enterprise Reference Architecture [297] for ICS systems security. As depicted in Figure 17, the Purdue Architecture organizes the ICS network into six layers: levels 4 and 5 form the Information Technology (IT) network, while the lower levels constitute the Operational

Technology (OT) network. The latter handles the control, monitoring, and automation of physical processes. At level 0, sensors and actuators are deployed to interact with the physical process: they are referred as Industrial IoT (IIoT) devices. They are directly connected to level 1, which comprises various Programmable Logic Controllers (PLCs). The PLCs implement system control logic by observing sensor readings and consequently updating actuator signals. They expose an OPC UA server and employ the OPC UA protocol for communicating with the upper layers. This aspect has become so ubiquitous in the Industry 4.0 ecosystem that virtually every major PLC vendor today includes an embedded OPC UA in their products [15, 128, 129].

Levels 2 (SCADA, HMI) and 3 encompass devices that implement supervisory control, data acquisition, and monitoring to manage plant operations. To provide these functionalities, these devices act as OPC UA clients by requesting sensors and actuators' data collected by the OPC UA server at level 1. As described in Section 3.3.2, the OPC UA communication is supposed to be properly configured as described by the official OPC Foundation guidelines [206], i.e. (1) by enabling Security Mode SignAndEncrypt with certificates either signed by a trusted Certification Authority or self-signed certificates, (2) disabling anonymous authentication, and (3) employing one of the three suggested Security Policies.

In the system model, network communication between levels 1 and 2 is assumed to be enabled through a P4 programmable switch. The P4 controller is deployed at level 3 of the Purdue Enterprise Reference Architecture and is able to communicate with the Enterprise Zone and with the public IOTA blockchain through the demilitarized zone (DMZ), which manages the connection between the IT and the OT networks while keeping them isolated from each other [130].

With respect to the attacker model, the ICS network is assumed to be partially controlled by a Dolev–Yao intruder [59]. A typical Dolev–Yao intruder can access all public network messages and modify, inject, delete, or delay them. The Dolev-Yao attacker has, therefore, almost infinite capabilities. However, the intruder is constrained by the perfect cryp-

tography assumption: he can only decrypt a ciphertext or forge a signature if they possesses the corresponding keys. In summary, cryptographic attacks (e.g., brute force on Private Keys or dictionary attacks on passwords) cannot be carried out by the attacker. Additionally, consistent with prior work in ICS security [1, 127, 220, 71], the adversary is assumed to operate only at or above Level 1 of the Purdue Enterprise Reference Architecture. The Dolev–Yao attacker is further constrained by the assumption of a secure channel between the P4 controller and the P4 programmable switch [311]; consequently, the P4 Runtime southbound APIs cannot be tampered with.

The attacker's goal is to perform one of the following three attacks:

1. Rogue Server: a new OPC UA device has been added to the network and needs to establish secure OPC UA communication with other devices. The attacker on the network aims to deceive this new OPC UA client by injecting malicious data. The attacker sets up an OPC UA server offering secure endpoints, making new clients believe they are communicating with the legitimate OPC UA server.

2. Rogue Client: the attacker attempts to connect to the OPC UA server to eavesdrop or alter the information exchanged between the server and clients. They set up a client that, despite lacking authorization from the network administrator, tries to establish a connection with the server.

3. Middle person attack: the attacker acts as an intermediary between the OPC UA client and server, intercepting and modifying all communications between them. This attack requires achieving both Rogue Client and Rogue Server objectives.

The attacker model is consistent with those in the prior work of OPC UA security [70, 227, 65, 64, 139]

3.3.6 PK-IOTA Architecture

This section introduces PK-IOTA, an automated architecture designed to manage and distribute OPC UA certificates, addressing the challenges

discussed in Section 3.3.4. First, a comprehensive overview of the elements encompassed within the architecture is given, along with the underlying assumptions. Then, the workflow is outlined, illustrating how the PK-IOTA entities interact during runtime.

The architecture of PK-IOTA, represented in Figure 18, is composed of four key and interconnected layers: ① Device Layer, ② Data Plane Layer, ③ IOTA Clients Layer, and ④ IOTA Layer. Its foundation is the Device Layer, comprising OPC UA clients and servers deployed in accordance with the Purdue Enterprise Reference Architecture. Building on this, the Data Plane Layer integrates P4 programmable switches to enforce real-time security policies and perform in-network certificate validation, ensuring that communication is solely allowed to trusted OPC UA devices. Positioned above the Data Plane Layer, the IOTA Clients Layer acts as a middleware connecting the blockchain with the devices in both Layer ① and ②, facilitating seamless management of OPC UA certificate transactions. At the top of the architecture, the IOTA Layer provides a secure, immutable ledger that underpins the entire system, handling the issuance and revocation of OPC UA certificates.

Device Layer

The Device Layer consists of OPC UA clients and servers already deployed within the ICS network, as described in Section 3.3.5. Each device is configured by the network administrator to use the Sign&Encrypt security mode, avoid deprecated security policies, and is equipped with a self-signed certificate or one issued by a trusted Certification Authority.

The assumption is that OPC UA devices can perform all certificate validation steps outlined in the OPC UA standard specification [203], excluding the "Trust List Check" for the reasons outlined in Section 3.3.3. This step involves verifying whether an OPC UA certificate is trusted by determining if it is included in the device's trust list. In the PK-IOTA architecture, the certificate validation is performed in-network at the Data Plane Layer, offloading it from end OPC UA devices.

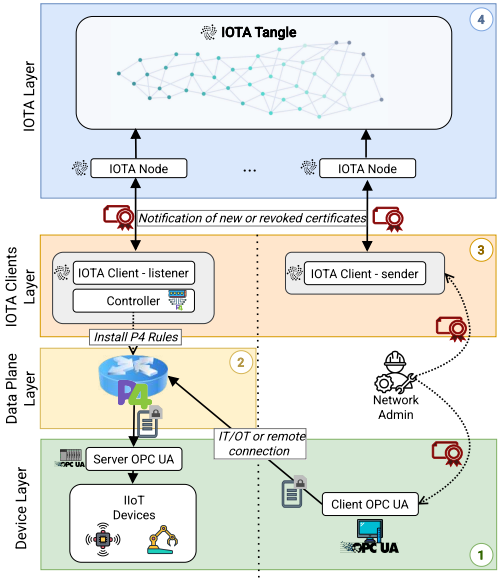


Figure 18: Overview and interactions between the 4 layers of the PK-IOTA architecture.

Data Plane Layer

The Data Plane Layer consists of P4 programmable switches inspecting network packets and enforcing trust on every connection establishment handshake between an OPC UA client and server. In particular, whenever an OPC UA Open Secure Channel request is sent by an OPC UA client, the programmable switch parses the OPC UA payload, extracting the sender's certificate and the certificate thumbprint. Subsequently, it checks the certificate's trust status: if the certificate is untrusted or revoked, the programmable switch drops the packet and the OPC UA connection is rejected directly at the data plane, preventing untrusted endpoints from establishing OPC UA Secure Channels in the first place. This process is then repeated with the following OPC UA Open Secure Channel response sent by the OPC UA server.

Algorithm 1 illustrates the PK-IOTA P4 Parsing Pipeline, which begins by handling the standard TCP/IP packet headers (Ethernet, IPv4, and TCP). Once the pipeline identifies that the payload is an OPC UA message, the parser transitions to a specialized state that extracts the OPC UA header and determines the OPC UA message type. If the OPC UA message type is OPN (indicating an Open Secure Channel), the switch parses the corresponding security header and proceeds to parse the sender's certificate and its thumbprint, as shown in Algorithm 2. The extractCert() procedure retrieves the certificate length from the relevant OPC UA message fields and then performs the necessary endianness adjustments. Specifically, although the OPC UA specification encodes the certificate length as a big-endian integer, it must be converted to little-endian to be processed as bytes in the P4 language. Since the maximum size of a variable in P4 is 256 bytes, and the certificate may exceed this limit, the parser processes the certificate iteratively. The final parsing step involves extracting the certificate thumbprint.

As soon as the OPC UA Open Secure Channel request or response is completely parsed, the Data Plane Layer enforces in-network, real-time trust verification of OPC UA certificates within the Ingress processing pipeline. The P4 switch checks the certificate thumbprint against a dedi-

Algorithm 1 PK-IOTA P4 Parsing Pipeline.

```
Input: packet p
Output: hdr.ipv4, hdr.tcp, tcp_metadata, hdr.opcua, opcua_cert,
    opcua_cert_thumbprint
 1: control block Parser
 2:     p.extract(hdr.ethernet);
 3:     if p is IPv4 then
 4:         p.extract(hdr.ipv4);
 5:         if p is TCP then
 6:             p.extract(hdr.tcp);
 7:             tcp.full_length = (hdr.ipv4.totalLen - IPV4_LEN) * 8;
 8:             tcp.header_length = ((bit<16>)hdr.tcp.dataOffset) << 5;
 9:             tcp.payload_length = tcp.full_length - tcp.header_length;
10:             if tcp_metadata.payload_length > 0  and p is OPCUA then
11:                 p.extract(hdr.tcp_options);
12:                 p.extract(hdr.opcua);
13:                 opcua_cert = extractCert(p);
14:                 p.extract(opcua_cert_thumbprint);
15:             end if
16:         end if
17:     end if
18: end control block
```

cated match-action table (thumbprint_table). This table is populated by the P4 controller, which installs rules mapping valid thumbprints to an "allow" action. If a matching thumbprint rule is found, the packet is classified as trusted, permitting the OPC UA Secure Channel establishment to continue. Conversely, if no corresponding rule exists, the data plane immediately drops the packet, thus preventing any connections with unknown or unauthorized OPC UA certificates. This mechanism effectively offloads trust-list management from individual OPC UA devices, enforcing real-time, centralized policy control at the data plane level.

Iota Clients Layer

To complement the real-time certificate enforcement performed by the Data Plane Layer, the IOTA Clients Layer provides an interface for OPC UA network administrators to manage and distribute OPC UA trusted

Algorithm 2 OPC UA Certificate extraction inside the PK-IOTA P4 Parsing Pipeline.

```
Input: packet p
Output: opcua_cert
 1: procedure extractCert()
 2:     bit<2048>[100] opcua_cert;
 3:     bit<32> cert_bytes;
 4:     bit<32> remaining;
 5:     state parse_certLength
 6:         p.extract(hdr.opcua_certLength);
 7:         bit<8> hex1 = hdr.opcua_certLength[31:24];
 8:         bit<8> hex2 = hdr.opcua_certLength[23:16];
 9:         bit<8> hex3 = hdr.opcua_certLength[15:8];
10:         bit<8> hex4 = hdr.opcua_certLength[7:0];
11:         cert_bytes = (bit<32>)(hex4 ++ hex3 ++ hex2 ++ hex1);
12:         remaining = cert_bytes;
13:         transition select(cert_bytes)
14:             0 : drop;
15:             _ : check_cert_length;
16:
17:     state check_cert_length
18:         transition select(cert_bytes > 255)
19:             false : parse_certificate_ending_part_only;
20:             true : parse_certificate;
21:
22:     state parse_certificate
23:         packet.extract(opcua_cert.next);
24:         remaining = remaining - 256;
25:         transition select(remaining > 255)
26:             false : parse_certificate_ending_part;
27:             true : parse_certificate;
28:
29:     state parse_certificate_ending_part
30:         packet.extract(opcua_cert, (bit<32>)(remaining * 8));
31:
32:     state parse_certificate_ending_part_only
33:         packet.extract(opcua_cert, cert_bytes);
34:
35:     return opcua_cert
36: end procedure
```

certificates via the IOTA Tangle. In this architecture, the P4 controller itself acts as a listener to relevant IOTA transactions originating from the administrator wallet. Whenever these transactions carry an OPC UA certificate for issuance or revocation, the controller updates the thumbprint_table entries in the P4-programmable switch accordingly, by either installing a match-action rule to allow OPC UA connections for a newly trusted certificate or removing a rule to prevent future connections for a revoked one. This mechanism ensures that trust relationships are tamper-proof and maintained in near-real time throughout the ICS network, offloading individual OPC UA devices from the complexities of certificate distribution.

From the Stardust release onward, IOTA enabled the implementation of smart contracts by allowing interoperability between two layers: Layer 1 (L1), based on IOTA's Tangle, and Layer 2 (L2), which works as a sidechain infrastructure and supports Ethereum virtual machines (EVMs) [63]. This dual-layer architecture enables different approaches to managing certificates, with each layer providing distinct advantages in terms of speed, computational requirements, and functional capabilities. PK-IOTA architecture is proposed with two different interactions with the IOTA layer: MQTT interactions with the Tangle (L1) and Smart contract (L2); the two solutions are distinct, and it can be chosen which of the two to use, as they reach the same objectives but with different requirements and effects, both in terms of performance and functionalities.

L1 implementation

In the L1 implementation, IOTA clients utilize the iota-sdk MQTT Python library to facilitate the exchange of transactions between devices. This framework allows the sender to read certificates sequentially and transmit them as the payload of a Tangle transaction. Receivers, meanwhile, listen for transactions tagged with specific labels, such as "certificate," using the MQTT protocol. The certificate data can be encoded within the transaction's payload in various formats, together with a boolean flag indicating whether the transaction involves the issuance of a new certificate or its revocation. To ensure the integrity of the process and filter out

unauthorized submissions, receivers only consider transactions originating from a designated administrator: clients verify the sender's signature by checking that the associated public key is among the pre-authorized keys (i.e., one of the administrator's keys). Each client implements both the sender and listener functionalities to enable bidirectional communication. This implementation leverages the inherent scalability and efficiency of the Tangle, allowing for the assessment of baseline performance in terms of computational overhead.

L2 implementation

In the L2 implementation, clients use Node.js along with the web3 library to interact with SCs deployed on the L2 sidechain. These SCs provide various functionalities to automate certificate management. They maintain a list of all valid certificates, automatically removing entries when certificates expire or are revoked. The addition and revocation of certificates are managed by specific functions: addCertificate(cert, expireDate) and revokeCertificate(cert). These functions update the local list of certificates and their statuses. Moreover, when changes occur in the list, the SC emits an event to notify all IOTA clients, ensuring real-time updates. The SC maintains a tamper-proof and valid list of certificates and facilitates certificate management and automation, such as during client initialization. It offers a function, getAllCertificates(), that retrieves all valid certificates and sends them to clients. As a result, clients only need to use the Node.js frontend application to interact with the SC and receive event notifications. Although this approach introduces a higher computational load due to the additional complexity of smart contract execution, it offers greater functionality and automation.

Iota Layer

The Iota Layer contains the IOTA Tangle, which is employed to provide tamper-proof and immutable records of certificate information, including revocations. Clients can interact with the Tangle either by running a local IOTA node or by connecting to publicly available nodes, also those provided by IOTA as a service. IOTA is equipped with load balancing to

optimize performance and ensure reliable access to its publicly available nodes.

Operating a local IOTA node offers direct, real-time visibility into the Tangle and, for transactions containing OPC UA certificates, enhances control and reliability. However, this setup requires substantial computing resources, which may be challenging for devices with limited processing power or storage capacity. Therefore, clients must carefully weigh the benefits of direct monitoring against the resource demands when choosing between a local node and a public one.

Local nodes utilize the Hornet daemon, managing L1 network assets and related tools. Clients can connect to both public testnets and mainnets in the L1 network. For SC capabilities, the Wasp daemon is used alongside Hornet to handle L2 assets. However, due to current security policies set by the IOTA community, running a local L2 node connected to official IOTA public L2 sidechain is not possible. Consequently, clients requiring SC functionality have two options: connect only to public L2 nodes provided by IOTA via their load balancing service or deploy a private L2 sidechain running local nodes. The choice depends on the specific trade-offs between resource availability, the desired control over the blockchain environment, and eventual fees.

Workflow

Figure 19 illustrates how new OPC UA devices are added as trusted entities in the OPC UA network and how OPC UA certificates are revoked. The workflow unfolds through coordinated interactions among the network administrator, the IOTA network, the P4 controller, and the P4 switch, thus ensuring that certificate updates are securely propagated throughout the 4 layers of the PK-IOTA architecture.

Insertion of a Certificate:
Initially, the network administrator generates or obtains a valid OPC UA certificate for the new client or server. During the process of provisioning the certificate to the OPC UA device, the network administrator automatically interacts with the IOTA network using an IOTA client to

register the new certificate as trusted. This certificate is published on the IOTA network through the IOTA Clients Layer, either to the Tangle on Layer1 via an MQTT-based transaction or to a smart contract on Layer2. Once the certificate transaction is successfully processed on the IOTA network, the local IOTA client on the P4 controller is notified. In response, the controller extracts the certificate's thumbprint and installs the corresponding entry in the thumbprint_table of the P4 switch via the TableEntry P4Runtime API [52]. These rules are enforced by the P4 switch, which controls the traffic between OPC UA clients and the server, allowing the new device to establish OPC UA secure channels with other trusted entities. Any future connection request involving the new certificate is accepted immediately, provided its thumbprint matches an entry in the switch's table. If an update is missed (e.g., transient disconnection), the controller reconciles state on resume by (L2) calling getAllCertificates() or (L1) replaying recent transactions from the administrator's address.

Revocation of a Certificate:
Revocation follows a similar process. The administrator, upon identifying a potentially compromised or expired certificate, sends a revocation transaction through the IOTA Clients Layer. On Layer1, the Tangle processes and broadcasts this update, while on Layer2, a smart contract removes the certificate from its valid list and emits an event indicating its revocation. In both cases, the P4 controller detects the revocation message through its local IOTA client and removes the certificate's thumbprint from the thumbprint_table. As a result, any subsequent attempt to initiate an OPC UA connection with that certificate is dropped at the data plane, effectively cutting off compromised devices from the network.

By joining in-network certificate validation with the IOTA-based certificate management, it is ensured that trust decisions remain consistent throughout the ICS network. Administrators can thus extend or revoke trust as needed, while end devices remain focused on core OPC UA functionalities rather than managing local certificate lists.

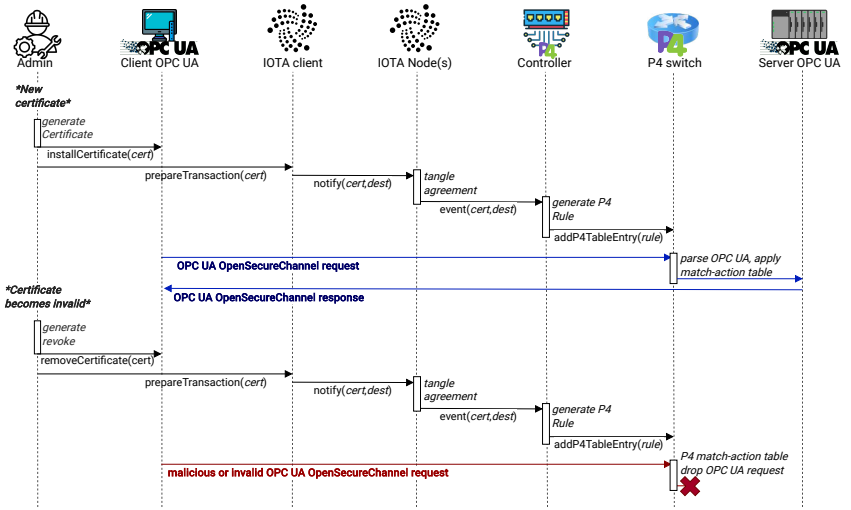


Figure 19: Workflow of the PK-IOTA certificate management process for adding and revoking certificates.

3.3.7 Evaluation

This section evaluates the overall efficacy of PK-IOTA to determine its applicability in real-world Industry 4.0 scenarios with real-time constraints on OPC UA communications. The assessment empirically addresses the following evaluation questions:

Q1 How much in-network overhead does PK-IOTA introduce on the data plane for validating OPC UA certificates during the OPC UA security handshake?

Q2 How efficiently can PK-IOTA propagate and manage OPC UA certificates through the IOTA Tangle, and how do the Layer 1 and Layer 2 approaches compare in terms of latency, resource consumption, and functionality?

Q3 What is the total resource consumption of the various components of the PK-IOTA architecture, and how does this impact runtime performance under realistic operational conditions?

This Section begins by outlining the experimental setup used to conduct the tests needed to answer these questions.

Experimental Setup

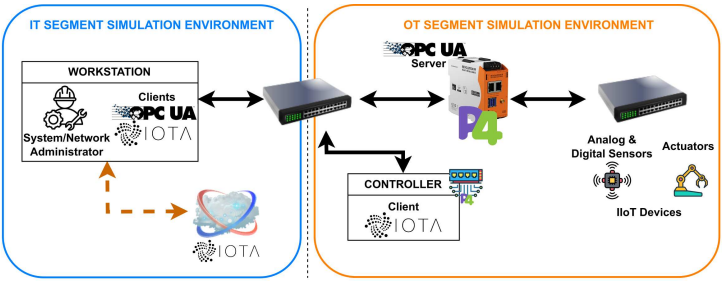


Figure 20: Experimental setup employed to conduct the evaluation of the PK-IOTA architecture.

To evaluate the proposed PK-IOTA architecture, a physical testbed was set up as depicted in Figure 20. This testbed adheres to the Purdue Enterprise Reference Architecture, maintaining the conventional separation between the Information Technology and Operational Technology domains. By incorporating devices and systems commonly encountered in industrial environments, the testbed provides a realistic representation of an Industry 4.0 scenario, enabling rigorous assessment of PK-IOTA's applicability and performance in practical settings.

The IT domain is implemented using a single high-performance workstation, representing the Enterprise Zone of the testbed. This workstation, a Dell PC equipped with an Intel i7 processor and 64 GB of RAM, hosts the OPC UA client (implemented with opcua-asyncio[6]) and the IOTA client for managing transactions. It represents the HMI and main workstation for the system administrator.

The interconnection between the IT and OT domains is realized through an Aruba Switch. The OT segment is structured into three layers, omitting the SCADA Level 2 of the Purdue Reference Architecture due to

[6]https://github.com/FreeOpcUa/opcua-asyncio

the small scale of the testbed. The first layer corresponds to Level 0 in the Purdue model and includes sensors and actuators[7]. Additionally, an embedded IoT device capable of using different industrial protocols[8] is employed. The second layer, corresponding to Level 1, is composed of a Revolution Pi (RevPi) device[9], an open-source industrial-grade computer used to emulate various components of an industrial network, such as PLCs or Edge Nodes. The RevPi was chosen for its versatility in simulating industrial protocols like OPC UA and Modbus, as well as for its compatibility with P4 bmv2 virtual switches. In addition, the RevPi hosts an OPC UA server that exposes the values of sensors and actuators operating at Level 0. The third layer, corresponding to Level 3 of the Purdue Architecture, is represented by an edge node equipped with an Intel i7 processor and 32 GB of RAM. This edge node hosts the IOTA client, which observes certificate transactions, and a custom P4 controller presented in previous works [12, 6, 236]. The edge node is connected to the Aruba Switch, enabling its interaction with the IOTA blockchain.

In-Network certificate validation Overhead (Q1)

To quantify the overhead introduced by in-network certificate validation within PK-IOTA, a series of experiments were conducted to measure packet processing and packet dequeuing times. These tests analyzed the impact of P4 data plane in-network certificate validation on the OPC UA connection establishment process. Specifically, Open Secure Channel request and response packets (i.e., with OPN message type) were tagged within the parser pipeline with metadata. Then, in the egress pipeline, the packet processing and dequeuing were measured times exclusively for tagged packets. The results of these computations were stored in registers, enabling a precise packet-level analysis.

Figure 21a presents the packet processing times for 1000 OPC UA connection attempts, covering 2000 Open Secure Channel requests and responses. When comparing standard OPC UA traffic to that with in-

[7] https://www.dexterindustries.com/grovepi/
[8] https://sklep.inveo.com.pl/en/monitoring/49-nano-temp.html
[9] https://revolutionpi.com/en/revolution-pi-series

network certificate validation, an average increase of 38%, approximately 2600μs per packet can be observed. This extra processing is attributed to the additional steps required to parse and validate OPC UA certificates in the data plane as described in Section 3.3.6.

Figure 21b displays the packet dequeuing times for the same experimental setup. The introduction of in-network certificate validation adds only about 1.04μs per packet, highlighting its minimal impact on overall network throughput.

Finally, Figure 21c compares the end-to-end handshake times across 1000 OPC UA connection attempts, with and without certificate validation. The total time for performing the OPC UA security handshake increased by 14%, approximately 14.37 ms on average, due to the in-network validation process. Despite this overhead, the handshake process remains within acceptable limits for real-time industrial applications.

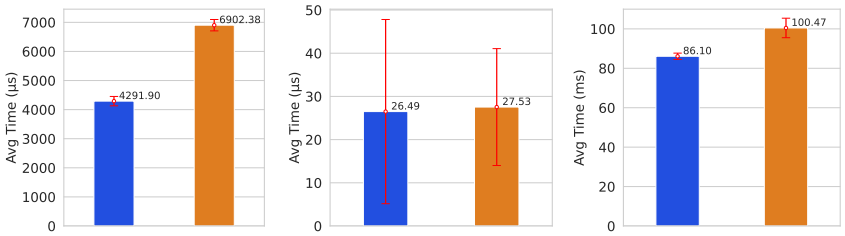
Overall, the results confirm that the inclusion of in-network certificate validation introduces a measurable but minimal overhead. The implementation ensures that the security of OPC UA communications is enhanced without significantly compromising the performance or real-time requirements of industrial control systems.

OPC UA certificates propagation delay over the IOTA Tangle (Q2)

PK-IOTA aims to provide a scalable, globally distributed certificate distribution system. For this reason, the experiments aimed to evaluate the feasibility and performance of IOTA as a certificate manager, focusing on the delays, resource consumption, and functionalities in transmitting certificates.

PK-IOTA is deployed on both layers to compare the performance and feasibility of these two approaches. By evaluating both models, the aim is to identify the trade-offs between transmission delays, computational efficiency, and the enhanced capabilities provided by smart contract automation.

Distinct Hornet and Wasp IOTA nodes were deployed over VMs located in different physical positions. In particular, VMs are placed respectively with long distances (United States - Australia and Europe -



(a) (b) (c)

Figure 21: Evaluation on a RevPi-hosted P4 data plane comparing standard OPC UA connections (blue) versus in-network certificate validation (orange): (a) packet processing times collected over 1,000 OPC UA connection attempts (2,000 data points from the OpenSecureChannel request/response); (b) packet dequeuing times under the same setup and dataset (1,000 attempts, 2,000 OpenSecureChannel points); (c) end-to-end OPC UA handshake time, reported as the mean over 1,000 client–server connection attempts, with error bars indicating the standard deviation.

Australia) and short distances (North Europe - Central Europe and even local) to evaluate the impact of the use of this technology over networks of different sizes, which is a requirement of the distributed IIoT use-case scenario. Moreover, the tests are performed with certificates in different formats to also assess the difference in delays caused by payload size.

The L1 tests, based on tangle transactions, are performed using Python scripts and the iota-sdk mqtt library[10]. Transactions are sent with JSON-RPC calls to the local Hornet node of the VMs, which operates over the IOTA Testnet.

300 tests were performed for each distance and each certificate format (.txt, .pem, and .der). The delay is considered between the local time of the sender's call to the function send() of iota-sdk and the local time at the reception of the transaction from the listener, with a previous synchronization of the two clocks.

The L2 tests, based on SC interactions, i.e. blockchain, are performed with Nodejs projects and the web3 library. The SC is written in Solidity[11]

[10]https://github.com/iotaledger/iota-sdk
[11]https://soliditylang.org/

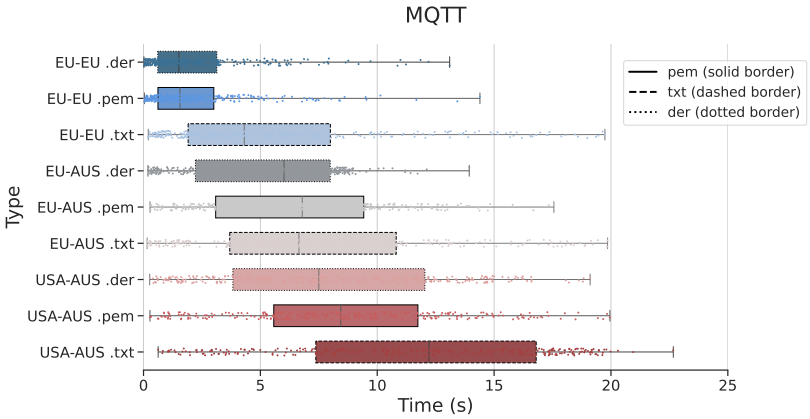


Figure 22: The boxplot illustrates the time taken to send and receive transactions in the IOTA L1 layer, using various certificate formats as payloads across different physical distances between nodes. The different certificate formats are distinguished by varying box border styles, while the color shades indicate the different physical distances.

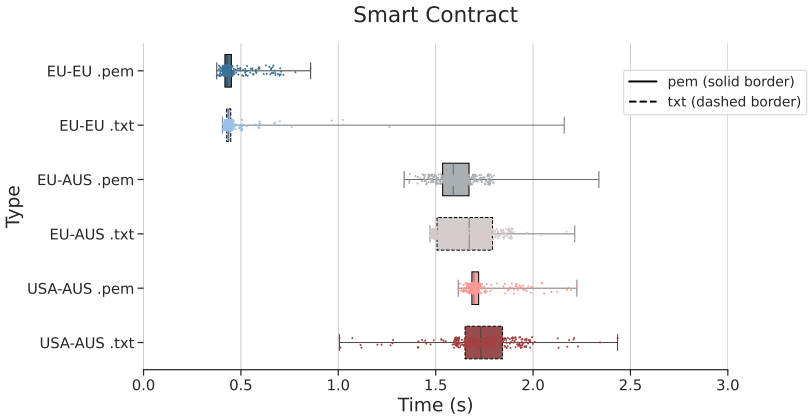


Figure 23: The boxplot illustrates the time taken to call the SC sendCertificates() function and receive the event notification in the IOTA L2, using various certificate formats as payloads across different physical distances between nodes. The different certificate formats are distinguished by varying box border styles, while the color shades indicate the different physical distances.

and deployed over a private chain, where all the Wasp nodes of the VMs from the tests take part. The local test, instead, is performed over the IOTA EVM testnet to evaluate timing over a public EVM testnet. The SC has some basic functions, such as storing, retrieving, and revoking certificates. The actions performed by the admin are mimed by calling the sendCertificates() function to insert the certificate of a new OPC-UA client in the storage manager; this is a restricted function, so it can be called only by authorized parties. Adding or revoking certificates will emit events that the receiver will be listening to. The delays are measured from the moment the sender calls the sendCertificate() function and the reception of the event by the listener. Tests are performed for the insertion case, as the revoke mechanism has an identical structure and, therefore, would give the same results in terms of delay. For each distance and certificate format (.txt and .pem), 300 tests were performed.

Results L1 tests As we can observe in the mqtt test's results shown in Figure 22, the average transaction time is proportional to the distance between network nodes. This behavior is likely influenced not only by network delay but also by the consensus mechanism of IOTA, given the significant differences observed across cases. Furthermore, in smaller networks, the number of outliers is notably lower, reinforcing that the efficiency of the transaction confirmation mechanism depends on node proximity [72, 313]. The process by which a transaction is added in IOTA primarily relies on the closest nodes, making this outcome expected, and delays caused by more distant transactions are likely also due to transaction reattachments within the DAG [54]. However, results show that even in cases where reattachment occurs, the delay remains within one minute, keeping the architecture responsive and within the system's operational requirements.

Results L2 tests The results on Layer 2, shown in Figure 23, differ significantly from those observed on Layer 1. First, the average time for sending a certificate, executed as a smart contract function, is considerably shorter than on L1, ranging between 1.5 and 2 seconds for long-distance cases and around 0.5 seconds for shorter distances, with no evident differences related to payload size. The low variance is likely due to the smaller

private network of the sidechain and the distinct L2 consensus mechanism, where the execution of the smart contract occurs on a blockchain that indirectly relies on a DAG [63]. Since updating the chain state on the L1 DAG is decoupled from the smart contract's execution, it does not affect the smart contract's response time, allowing for quicker processing and additional functionalities, such as the getAllCertificates() function, which retrieves all valid certificates at a given moment.

Overall Resource Consumption (Q3)

Table 4: Comparison of memory usage, CPU usage, and power consumption for each component.

Component	Deployed in	Memory (MB)		CPU (%)		Power Cons. (W)	
		Mean	Max	Mean	Max	Mean	Max
IOTA sender	Windows Workstation	148.17	154.00	0.61	0.63	19.18	19.37
IOTA receiver	Edge Node	132.22	170.02	0.23	0.67	18.53	19.40
IOTA node	Azure Cloud VM	444.27	455.31	3.38	3.80	25.33	26.21
OPC UA Client	Windows Workstation	53.97	54.00	1.54	2.28	18.73	20.45
OPC UA Server	RevPi Connect 4	110.85	112.12	10.31	11.30	2.15*	2.15*
P4 bmv2	RevPi Connect 4	233.45	262.76	29.09	35.75	2.15*	2.15*
P4 Controller	Edge Node	288.05	288.05	6.63	6.97	45.77	57.80

* Power on RevPi estimated via Ohm's law $P = V \cdot I$: voltage from vcgencmd measure_volts, current $I = 2.5\,\text{A}$ supplied by the DIN Rail PSU MDR-60-24.

Table 4 summarizes the memory, CPU, and power consumption of each PK-IOTA component, revealing how resource demands vary according to each component's role within the architecture. The IOTA node exhibits the highest memory footprint at 444.27 MB on average, reflecting the overhead of maintaining a direct connection to the Tangle and processing distributed certificate transactions. By contrast, the IOTA sender and receiver processes require significantly less memory (148.17 MB and 132.22 MB on average, respectively) since they only handle lightweight, client-side interactions with the ledger.

The RevPi, which functions as both an OPC UA server and a host for the bmv2 P4 switch, averages 233.45 MB of memory consumption, underscoring the computational demands of simulating industrial devices and running in-network certificate checks. This dual role also contributes

to its higher CPU utilization, which averages 29.09%.

In terms of power consumption, the edge node that hosts the P4 controller and IOTA client shows the highest average draw at 45.77 W. This usage reflects its dual responsibilities of continuously listening for certificate transactions and dynamically modifying match-action table rules via the P4Runtime API. Meanwhile, the RevPi's power usage, estimated through Ohm's law, remains relatively modest at 2.15 W, even though it manages industrial traffic emulation and P4 switching concurrently.

Overall, these findings confirm that while resource usage differs across components, the PK-IOTA architecture operates within feasible limits for typical industrial applications. The results highlight that both DLT-based certificate management and in-network certificate enforcement can be supported in real time without imposing excessive strain on the underlying hardware.

3.3.8 Discussion & Limitations

Security Analysis

The security of the OPC UA protocol has been extensively analyzed and formally proven by [227, 65, 64]. Building on these formal guarantees, PK-IOTA architecture augments OPC UA with a decentralized certificate manager and in-network certificate validation to ensure robust protection against Rogue Server, Rogue Client, and Middleperson attacks. To formally prove this, based on the threat model described in Section 3.3.5, the DLT is modeled as an authentic channel. By doing so, the assumption is that messages published on, or retrieved from, the IOTA ledger cannot be tampered with or falsified by the attacker. While an adversary can read these messages, they cannot alter the sender's identity, which is represented by the network administrator in this use-case scenario, or the content of any transaction. This property ensures that a certificate added to the IOTA ledger for issuance or revocation arrives unmodified at all legitimate parties. Additionally, the communication link between the P4 controller and the P4 programmable switch is modeled as a secure channel that is both confidential and authentic. An adversary cannot

learn or modify messages exchanged over this link, which prevents on-path manipulation of P4 rules or the insertion of forged entries into the thumbprint_table. The next section illustrates how the Pk–IOTA architecture thwarts each attack.

Security against Rogue Server Attacks

Let C_s be the certificate of an OPC UA server S, and let T denote the trusted certificate set stored on the IOTA ledger. In PK-IOTA, the P4 data plane inspects each OpenSecureChannel request and checks whether $C_s \in T$. If $C_s \notin T$, the request is dropped immediately, preventing the connection from being established.

Under the defined threat model, the Dolev–Yao attacker cannot modify the IOTA ledger or falsify transactions, because the ledger is treated as an authentic channel: the attacker can read messages but cannot alter them or spoof their origin. Similarly, the adversary cannot tamper with the secure channel between the P4 controller and the programmable switch, so it cannot install malicious rules or bypass the ledger-based trust checks. The only remaining avenues for a Rogue Server attack would be to (1) steal the ledger wallet credentials of the network administrator and add a fake certificate to T, or (2) compromise an existing OPC UA server directly. Both scenarios exceed the adversarial capabilities defined under the assumed Dolev–Yao attacker model. Additionally, the first scenario will involve social engineering techniques and is outside the scope of this work. The second scenario, instead, violates the assumption that the attacker does not have access to devices below Level 2 of the Purdue architecture, which is a typical location for OPC UA servers in industrial control systems. Therefore, given these channel assumptions and the ICS network segmentation, the attacker cannot succeed in creating a Rogue Server.

Security against Rogue Client Attacks

Let C_c be the certificate of an OPC UA client C, and let T denote the trusted certificate set on the IOTA ledger. In PK-IOTA, the P4 data plane

inspects every OpenSecureChannel request originating from a client. If $C_c \notin T$, the packet is dropped immediately, preventing any connection attempt from proceeding.

Within the defined threat model, a Dolev–Yao attacker cannot alter the ledger, since it is modeled as an authentic channel: the attacker can read but cannot forge or modify transactions. Likewise, the secure link between the P4 controller and the switch is beyond the attacker's reach, so installing malicious entries in the thumbprint_table is not possible. The only viable option for impersonating a legitimate client would be to (1) steal the ledger account belonging to the network administrator and add the attacker's certificate to T, or (2) compromise an existing OPC UA client. Both scenarios lie outside the defined threat model, which assumes a Dolev–Yao attacker.

Security against Middleperson Attacks

For a Middleperson attack to succeed, the attacker must impersonate both the server S (with certificate C_s) and the client C (with certificate C_c). However, as shown in the analyses of Rogue Server and Rogue Client attacks, neither impersonation is feasible within the PK-IOTA architecture. Consequently, the conditions necessary for a Middleperson attack cannot be fulfilled.

Scalability

The scalability of the PK-IOTA architecture requires consideration of both the P4 programmable data plane and the IOTA blockchain. In the data plane, OPC UA certificates are parsed in 2048-bit (256-byte) chunks by means of a P4 header stack. For the experiments, the header stack size is set to 100, allowing for a maximum certificate length of $256 \cdot 100 = 25600$ bytes. Since the header stack size is defined by a 32-bit unsigned integer in the P4 language specification, this limit can be significantly increased. This is a parameter whose value should be taken into consideration at deployment time to balance resource usage and performance.

The thumbprint_table in the P4 pipeline is configured to hold up to

1024 match-action table entries. Each entry corresponds to the certificate thumbprint of an OPC UA device. This capacity is expected to suffice for most industrial networks, yet it can be expanded based on the capabilities of the P4 programmable switch. The RevPi employed, for instance, has 32 GB of available memory, and related studies show that modern programmable switches typically provide abundant memory for match-action table entries [48].

With respect to the IOTA blockchain, instead, this solution behaves differently depending on whether it operates on the L1 or the L2 of the IOTA network.

On L1, scalability is one of Tangle's core strengths. Its feeless nature ensures that the cost of issuing or revoking certificates does not increase with the number of transactions. Additionally, the IOTA Tangle is designed to improve its performance as the network grows, meaning that as more certificates are managed and more nodes participate, the system's throughput scales effectively [313]. Therefore, even with a significant increase in the number of certificates, the solution remains sustainable in terms of performance and cost. However, it is worth noting that operations on L1 may occasionally encounter network-wide confirmation delays during periods of extremely high activity, as observed during the performance tests.

On L2, scalability depends on the architecture of the private sidechain or the chosen public L2 solution, as the chain settings are configurable at deployment time, allowing for tailored adjustments. When using a public L2 sidechain, scalability is influenced by that platform's fees and network congestion, which can become a limitation if certificate operations increase dramatically. However, in this architecture and specific PKI use case, the number of certificates is not expected to be very high, mitigating potential scalability concerns.

Costs of the IOTA Infrastructure

When discussing the theoretical monetary costs of this PKI implementation for OPC UA clients and servers on the IOTA blockchain, it is essential to distinguish between L1 and L2 operations. As mentioned earlier,

transactions on the IOTA Tangle at L1 are feeless, meaning registering or revoking certificates does not incur direct transaction costs. Moreover, sending transactions on L1 does not require additional infrastructure costs unless one opts to operate a local IOTA Hornet node for greater control or independence, though this is not strictly necessary.

However, if the implementation relies on L2, a private sidechain would need to be established to maintain feeless operations. The costs would primarily stem from maintaining the L2 infrastructure, including hosting and operational expenses. Alternatively, relying on a public blockchain on L2 is possible, which typically involves transaction fees. Thus, the monetary costs depend on the choice between sustaining a private L2 infrastructure or paying fees for a public L2 network. Nevertheless, from the performance tests conducted, the computational resource costs required to sustain the private IOTA infrastructure have proven negligible and are entirely manageable by the workstations operating as OPC UA clients and servers.

3.3.9 Related Works

OPC UA certificate management is a widely recognized challenge in industrial communication security, leading to numerous research efforts aimed at simplifying or automating the associated processes [139, 137, 138, 186]. For instance, the work by Atutxa et al. [17] demonstrates the potential of in-network validation of Datagram Transport Layer Security (DTLS) certificates to enhance both the efficiency and security of IIoT communications. This approach closely aligns with PK-IOTA's objectives, which combine blockchain technology with P4-based in-network processing to ensure robust certificate management and validation.

Beyond the scope of OPC UA, many researchers have explored blockchain-based approaches for implementing PKIs, such as Certcoin [78] and its privacy-aware extension [18]. They are deployed over different public blockchains: Bitcoin, in CertCoin [78], Ethereum in LightCert4IoT [84], Hyperledger Fabric in DPKI [216]; however, they follow the same core concept: leveraging the blockchain to distribute the certification management over different nodes but still guaranteeing the authenticity and security of

the infrastructure. Toorani et al. [272] implement a dynamic PKI that removes reliance on centralized Certificate Authorities by using blockchain and consensus mechanisms for secure public key registration, revocation, and verification, whereas [142] exploits the blockchain to achieve a PKI with Certificate Transparency.

Most of the works employ classic blockchains as an append-only storage [293, 155, 177], thus not considering the advantages that DAG-based blockchains could bring. Only a few works leverage a DAG-based ledger for managing certificates [267, 291], but they lack integration with smart contracts and are based on initial versions of IOTA, which require a central coordinator. However, they confirmed the utility and advantages of DAG-based PKI, including no transaction costs, low energy consumption, scalability, and parallelization. By building on newer IOTA versions, PK-IOTA expands these concepts to include both L1 and L2 functionalities, aiming to provide greater flexibility, improved performance, and richer OPC UA certificate management features.

3.3.10 Future works

Pk–IOTA is an architecture that integrates programmable data planes with a DAG–based distributed ledger to address certificate management in OPC UA deployments for Industry 4.0. By combining P4–based validation in the network with IOTA for decentralised publication, verification, and revocation of certificates, the framework automates trust–establishment and enforces secure communication across heterogeneous devices. Under the stated threat model, a Dolev–Yao adversary cannot successfully mount OPC UA Rogue Server, Rogue Client, or Middleperson attacks.

Evaluation on a physical testbed indicates that in–network certificate validation introduces a modest overhead on the OPC UA security handshake, with an average increase of approximately 14%, without disrupting time–sensitive operations typical of ICS environments. The use of IOTA provides an immutable and tamper-evident substrate for certificate issuance and revocation with low computational overhead, suitable even for resource-constrained settings and resilient to single points of failure that affect centralised PKI designs. Comparative analysis between

Layer 1 and Layer 2 on IOTA highlights a flexible deployment space: fee–less operations and simplicity on Layer 1 versus enhanced functionality on Layer 2, where smart contracts enable automation of certificate lifecycle tasks. Tests show that both configurations remain sustainable as the number of certificates grows, with lightweight processing that avoids becoming a bottleneck for OPC UA client and server operations.

However, the approach has limitations. P4–capable hardware is required to enforce validation in the data plane, and the IOTA substrate entails operating compatible infrastructure, especially in private or hybrid deployments. Layer 1 can leverage public nodes, whereas Layer 2 needs a dedicated side chain with associated operational costs. These trade–offs can be mitigated through careful architectural choices and benefit from the scalability properties of the IOTA DAG.

Future work encompasses three directions. First, broadening applicability across programmable data planes through the Tofino software architecture [210]. Second, extending support to OPC UA Part 21: Device Onboarding [202] to streamline secure provisioning. Third, with IOTA Rebased, investigating smart contracts directly on Layer 1 to remove the need for a separate Layer 2, simplifying deployments while retaining tamper–proof and scalable properties and enhancing automation of issuance, revocation, and lifecycle monitoring.

3.3.11 From Identity Assurance to Trust in Distributed Learning

The results obtained so far cover two complementary pillars for securing industrial CPS communications: metric–driven monitoring and reaction (GRAPH4), and identity and policy enforcement with automated certificate handling (Pk–IOTA). Yet, as modern ICPS increasingly rely on collaborative contributions from a wide variety of edge devices and organizations, identity and transport security alone do not guarantee reliable operation. In open and crowd–sourced cyber–physical ecosystems, participation is intended to be permissionless: nodes may join and leave at will, contribute sporadically, and operate across heterogeneous administrative domains [242]. In such settings, managing identity through traditional

public key infrastructures and long–lived X.509 certificates is not aligned with both operational reality and the design goal of openness, as it introduces enrollment overhead, central gatekeepers, brittle revocation workflows, and persistent identifiers that conflict with privacy and churn. This shift in assumptions necessitates mechanisms that can maintain openness while establishing accountability and reliability based on observable behavior, rather than relying on static credentials.

First, certificate–centric identity presupposes an enrollment authority and a provisioning workflow that issues, distributes, rotates, and revokes credentials. This introduces gatekeeping, administrative overhead, and potential single points of failure.

Second, openness and privacy expectations often conflict with strong, persistent identities. Many contributors prefer to remain pseudonymous or rotate their keys regularly to minimize linkability across interactions. Binding each participant to a stable certificate simplifies access control but undermines unlinkability, creates durable identifiers that can be tracked across sessions, and can deter participation from stakeholders who do not wish to disclose organizational or personal attributes. Where data protection and minimal disclosure principles apply, strong identity becomes a liability rather than an enabler.

Third, heterogeneity across vendors, software stacks, and device capabilities leads to uneven support for certificate storage, secure elements, and on–device cryptographic hygiene. Weak key protection, certificate reuse, and expired or misconfigured chains are common failure modes. Enforcing uniform PKI practices in a federated, multi–stakeholder landscape requires coordination that is costly and fragile. The resulting configuration drift erodes the very guarantees that certificates are meant to provide.

Fourth, open participation amplifies adversarial strategies such as identity multiplication. Certificates do not, by themselves, prevent Sybil–like behaviors where an entity presents many distinct identities to gain disproportionate influence. Even with stringent issuance policies, any scheme anchored in static credentials is vulnerable to key compromise and clandestine redistribution, especially at the edges where physical security is

limited.

Finally, the control surface in open collaboration shifts from authenticating "who" to evaluating "what" is being contributed. When the value and risk of participation are determined by behavior over time, static proofs of identity provide limited assurance. What is needed is accountability that does not rely on a priori vetting and that can cope with ephemeral, rotating, or pseudonymous identifiers while still enabling exclusion of harmful actors and attenuation of low–quality contributions.

For these reasons, certificate–based identity, while effective in closed and tightly administered environments, is not well-suited as the primary mechanism for open, crowd–sourced contexts. Alternative models must privilege openness and privacy, tolerate churn and heterogeneity, and focus assurance on observable behavior and continuously updated evidence rather than on static enrollment credentials. In the next section, an architecture to address this problem will be presented, with a specific use-case scenario: Federated Learning.

3.4 DAGTrustFL: Trust management for distributed AI systems

3.4.1 Federated learning in IoT and its vulnerabilities

Federated Learning (FL) is a distributed training paradigm in which multiple participants collaboratively learn a shared model by exchanging updates (e.g., gradients or model deltas) rather than raw data. A coordinator aggregates these updates into a global model and redistributes it to participants, enabling iterative improvement while keeping local datasets in place [184]. This approach reduces the exposure of sensitive information and curtails the risks inherent in centralized data collection, which has made FL attractive for privacy–aware and bandwidth–constrained settings [157].

In industrial and urban IoT scenarios, such as smart manufacturing lines, traffic sensing, energy grids, or environmental monitoring, data are naturally generated at the edge and are often heterogeneous, bursty, and

sensitive. FL aligns with these constraints by allowing devices to contribute to a common model without exporting raw measurements, thereby lowering communication costs and enhancing privacy. Smart–city deployments, in particular, can benefit from FL to build models for forecasting, anomaly detection, and resource optimization from geographically dispersed sensors while respecting local data residency and privacy requirements [215].

IoT ecosystems are frequently open, dynamic, and multi–stakeholder: devices may join or leave, connectivity is intermittent, and capabilities vary widely. This participation model eases adoption and scales contributions, but it also weakens assumptions that underlie traditional identity and access control. In practice, FL must tolerate non–IID data distributions, irregular reporting, energy constraints, and limited cryptographic hygiene across heterogeneous hardware and software stacks [157, 215].

These possibilities allow a new paradigm to emerge within FL: public environments. In this scenario, any IoT device can participate in the training process, joining and leaving the network at will. Unlike more controlled setups, there is no central authority to vet the devices or enforce strict participation rules. This openness fosters inclusivity and lowers barriers to entry, allowing a broader range of devices to contribute to model improvement. For instance, an IoT device might connect to the network, train on the provided pre-trained model until it reaches the desired accuracy, and then disconnect after updating the global model. This dynamic environment offers notable economic advantages for participants. Access to pre-trained models can require costly licensing fees. In public FL environments, IoT devices receive a pre-trained model and actively enhance their performance by contributing with their locally gathered insights. As they improve the model, they also derive greater value from it, creating a collaborative relationship where costs are significantly reduced.

The promise of public FL environments lies in their ability to make cutting-edge machine learning more accessible and cost-effective for IoT ecosystems, particularly smart cities. Yet, this openness also introduces significant challenges, particularly concerning the trustworthiness of the devices and the integrity of the global model. Since FL does not provide

visibility into each participant's local training process, malicious devices may attempt to poison the model, enabling evasion attacks. In smart city scenarios, ensuring the integrity of critical data in these untrustworthy environments is essential. This balance between opportunity and risk forms the foundation for exploring advanced mechanisms to secure and scale FL in such dynamic, decentralized systems, and trust between peers becomes fundamental. Standardizing trust management in FL can ensure consistency, security, and interoperability across diverse systems. A unified approach would streamline the deployment of FL models, improve scalability, reduce fragmentation, and facilitate broader adoption while maintaining security.

In cyber–physical contexts, model errors and delayed detection can propagate into control decisions and physical actuation. The consequence space therefore includes safety, availability, and quality–of–service impacts, not only accuracy loss. Two classes of safeguards emerge as necessary complements to transport and identity security: trust assessment over contributions (e.g., consistency, similarity, and reputation over time) to modulate acceptance, weighting, or quarantine of updates; and auditable coordination across stakeholders to record decisions, support rollback, and enable forensic analysis in case of suspected poisoning or collusion [157, 215].

Trust Management in IoT Environments

In FL, practical applications in collaborative IoT environments rely on distributed data and privacy preservation. However, these environments pose significant challenges, as they involve diverse and resource-constrained IoT devices that contribute to a shared model without inherent trust among participants.

Trust Challenges in IoT-Based FL

In smart cities, IoT devices such as traffic sensors and environmental monitors generate valuable data for applications like traffic prediction and resource allocation. However, the integrity of local model updates is critical. Malicious devices could introduce manipulated data, leading to

incorrect model outputs that impact public safety. A poisoning attack on a traffic prediction system, for example, could cause congestion, while flawed environmental data could hinder climate monitoring efforts.

Similarly, in smart healthcare, FL enables the training of AI models across networks of Internet of Medical Things (IoMT) devices, hospitals, and research centers while preserving patient privacy. However, the accuracy of local model updates directly impacts medical outcomes. A model poisoning attack could lead to misdiagnoses, improper treatment plans, or even life-threatening errors.

Unlike traditional centralized environments, where data sources can be strictly verified, FL in collaborative IoT lacks inherent trust. This necessitates trust mechanisms capable of evaluating model updates, detecting anomalous behavior, and mitigating threats dynamically.

Defining Trust in IoT Systems

Trust in IoT refers to confidence in a device's reliability, integrity, and behavior. Unlike human trust, which is often based on intuition, trust in IoT relies on measurable, dynamic criteria that evolve based on interactions and performance. Several key properties define trust in these environments:

- Asymmetry: Trust is not necessarily mutual. For instance, Device A may trust Device B, but the reverse is not guaranteed.
- Transferability: Trust can be inferred indirectly. If Device A trusts Device B and Device B trusts Device C, Device A may develop trust in Device C.
- Subjectivity: Each device or system has its own trust requirements and evaluation criteria.
- Dynamism: Trust scores evolve over time, adapting to new interactions, decaying, or strengthening based on evidence.

Effectively managing these trust properties in IoT-based FL requires sophisticated frameworks known as Trust Management Models (TMMs),

which dynamically evaluate, propagate, and update trust across the network. Thus, TMMs provide structured methods to assess and maintain trust in decentralized IoT networks [281]. They consist of five key components:

- Trust Composition: The attributes or samples used to evaluate trust, such as Quality of Service (QoS) data: data accuracy, responsiveness, and social factors like cooperation and frequency of communication.

- Trust Indicators: The single application-specific metrics, such as sensor reliability in IoT or the authenticity of partial models in FL, that make part of the composition.

- Trust Computation: Methods to calculate trust, ranging from weighted aggregations to advanced models leveraging machine learning or entropy-based analysis.

- Trust Propagation: Disseminating trust evaluations throughout the network.

- Trust Updating: Adapting trust scores dynamically using decay functions, sliding windows, or aging factors to ensure trust reflects recent behavior.

Approaches to TMMs

Various TMMs have been proposed to manage trust in FL, differing in how trust is computed, propagated, and evaluated, but no standard is available or proposed. Some approaches rely on centralized trust computation, where the FL server assesses trust based on model updates, ensuring robustness but creating a Single Point of Failure (SPOF). Instead, trust evaluation can be distributed across edge nodes using various techniques, even applying Deep Reinforcement Learning [179]. Other models establish inter-server trust [179], enabling the expansion of the trust environment, a case particularly attractive in the Smart Cities scenario. Trust computation and trust indicators vary, with some methods clustering FL

agents by behavioral similarity, while others dynamically adjust aggregation weights based on model accuracy [292]. In this landscape, Trusted Execution Environments (TEEs) have been explored as a potential solution, as they provide secure enclaves for executing sensitive operations, ensuring data integrity and confidentiality even in compromised systems. However, their applicability in IoT-driven FL is limited due to hardware constraints and the heterogeneity of IoT devices, which often lack the resources needed for efficient TEE deployment [191].

Issues of TMM

Managing trust in FL and IoT is challenging due to network heterogeneity, resource constraints, and evolving security threats. Sybil attacks exploit false identities, while bad-mouthing and self-promoting attacks manipulate trust data. Trust inconsistency arises in decentralized systems because each device independently calculates trust based on data it gathers from its own perspective, often obtained from indirect trust evaluations of different network nodes. The local trust computations can vary significantly due to differences in the quality, availability, and timeliness of the data each device receives. As a result, devices may reach conflicting conclusions about the trustworthiness of the same entity. This inconsistency creates vulnerabilities, such as enabling malicious actors to exploit gaps or discrepancies in trust assessments. Mechanisms that enable the consistent sharing of trust data among all devices, ensure access to a tamper-proof historical record of trust evaluations, and provide guarantees of non-repudiation and data integrity can mitigate these issues and establish a unified and secure foundation for trust propagation in decentralized environments [167].

Blockchains in FL trust

Blockchain technology has become a critical component in TMMs, particularly for managing trust in decentralized environments such as FL and general IoT networks [166], for several reasons:

- One of the most significant benefits blockchain brings to TMM is decentralization. By distributing trust-related information across

nodes with equal rights and obligations, blockchain eliminates the risk of a SPOF and promotes fairness in the system.

- Another fundamental feature is tamper-proofing. Blockchain ensures that data recorded within its ledger cannot be altered, facilitating auditable and reliable trust evaluations.

- Smart contracts further augment TMM by automating trust-related processes. These self-executing programs run on the blockchain, ensuring the atomicity of trust evaluation rules and consistent application without human intervention.

- Consistency is another critical advantage offered by blockchain, as it maintains a unified record of trust data across all nodes that keeps integrity.

Blockchain contributes to TMM through its layered integration within IoT architectures. At the sensing layer, it can record all device data, creating an end-to-end chain of trust from data collection to processing. At the network layer, blockchain can act as an auditable log of data transmission, verifying the integrity of data paths and constraining intermediary behavior. At the application layer, blockchain can enhance user confidence by enabling secure access control, identity management, privacy preservation, and trust-based auditing [166]. Moreover, blockchain and TMM exhibit mutual benefits. Trust Management can facilitate blockchain's mechanisms by evaluating the trustworthiness of participants, influencing roles in consensus mechanisms, assisting in asset management, providing trust to cross-blockchains, and creating incentive mechanisms based on reputation credits. Blockchain, in turn, provides the robust infrastructure required to implement decentralized, transparent, and tamper-proof trust evaluation. In [28], blockchain accelerates transmission rates and ensures reliability in Medical IoT-based FL. Based on blockchain-stored trust and reputation scores, federated aggregators and group leaders are elected. They employ a trust-empowered consensus mechanism that enhances the efficiency of the agreement.

Challenges and limitations of Blockchain-based TMMs

Despite their advantages, blockchain-TMMs face several limitations, as both technologies are complex and require considerable computational resources. One key challenge is scalability. From an auditability perspective, it is preferable to store extensive trust-related data. However, while blockchains offer tamper-proof records, the high volume of trust data that can be stored and managed within blockchains can be problematic [300]. The sequential layout of a classical blockchain, one block after the other, limits the blockchain throughput, creating a bottleneck when trust data reach significant density, as shown in Fig. 24. Techniques such as sharding, off-chain processing, and reducing storage redundancy are often employed to strike a balance between scalability and auditability. Energy efficiency is another critical issue. Classical consensus mechanisms, such as Proof of Work, are extremely energy-intensive, making them unsuitable for resource-constrained IoT and IoMT devices. Lightweight alternatives are urgently needed to ensure the practical deployment of these systems. There are currently more energy-efficient solutions, like PoS, where validators are chosen to create new blocks and confirm transactions based on the amount of cryptocurrency they stake as collateral. However, most PoS protocols are vulnerable to certain security threats [74] and limit access to the consensus algorithm to a minority set of nodes that can perform staking and, potentially, engage in censorship. Context-awareness in trust composition is often overlooked, even if it is essential for TMMs to function effectively in heterogeneous IoT environments. FL operates in diverse scenarios, and trust evaluations must adapt to varying contexts, such as model characteristics, data reliability, and honest behaviors in trust recommendations. Just a few of the reviewed blockchain-TMMs by Liu et al. [166] satisfy context awareness. Moreover, blockchain-TMMs are designed to enhance security within distributed systems, but can also introduce vulnerabilities. Bad-mouthing attacks, for example, occur when malicious entities intentionally provide false feedback about honest participants, and Sybil attacks happen when a single adversary creates multiple fake identities to gain an unfair reputation. Addressing these limitations requires innovative DLT designs to overcome scalability

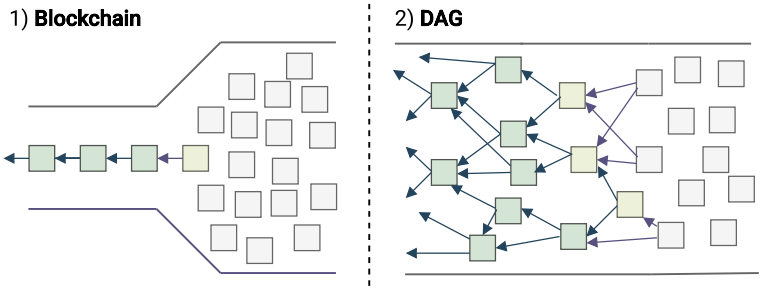


Figure 24: The bottleneck of blockchain vs DAG.

and energy-efficiency limitations, as well as the integration of specific and context-aware trust indicators for efficient and effective trust tailored to FL and IoT needs.

As already presented in Section 3.3.4 with IOTA, a DAG-based ledger organizes its structure as a directed acyclic graph, where each node represents a new single transaction. Unlike blockchains, which sequentially add blocks one at a time, DAGs allow transactions to be processed asynchronously, as shown in Fig. 24. This means multiple transactions can be added simultaneously, drastically improving transaction throughput. In DAGs, each transaction is linked to one or more previous transactions, serving as a record of new activity and as confirmation of earlier data. Each transaction has a weight based on computational effort, and its Cumulative Weight (CW) is the sum of its own weight and those of all transactions approving it. As more approvals accumulate, the CW increases. Once it surpasses a set threshold, the transaction is considered confirmed. Blockchains can be seen as a specific case of DAGs, where the cardinality of each link between nodes is just one. However, DAG and blockchain follow different principles: while blockchain prevents forking by delaying block creation via consensus mechanisms, DAG allows simultaneous transaction insertions, leading to multiple forks. However, DAG limits excessive forking to avoid attacks like double-spending. Similar to Blockchain's longest chain rule, DAG employs the heaviest DAG rule,

where tip selection favors transactions with the highest CW [160].

Leverage DAG and Context-Aware metrics

DAG-based ledgers address critical issues of scalability and synchronicity inherent in traditional blockchains, making them well-suited to take their place in managing trust in FL environments [5]. The DAG structure provides an efficient DLT for recording trust-related data and enables automatic and context-aware trust computation across IoT and edge devices. However, integrating these technologies could be challenging as the complexity of the overall architecture leads to an augmented attack surface, which can also damage efficiency and limit scalability advantages. Despite the potential of DAG-based TMMs to address challenges in decentralized environments, the number of solutions in the literature remains limited and incomplete, exhibiting notable gaps [5]. In particular, the literature shows no dedicated architectures for managing trust in FL, which limits the use of general QoS indicators in trust evaluation. We state that these indicators alone are insufficient for detecting malicious behavior in FL environments. Moreover, these architectures often use DAGs to record all network interactions, leading to high storage demands and inefficiencies, which can particularly hinder scalability. On top of that, ensuring equitable participation remains an issue, as nodes with higher trust might not receive proportionate rewards or influence in consensus, potentially disincentivizing their active involvement [306]. For these reasons, it is desirable to develop an alternative framework that enables seamless integration between TMMs and DAG-based ledgers. Designing such a framework, however, first requires a thorough analysis of the potential threats that may emerge in these systems, which are presented in the following section.

3.4.2 Threat Model

The following system model is assumed. An open, crowd-sourced FL deployment runs over IoT devices, coordinated by an honest aggregator and augmented with a DAG-based trust layer. In each round, a subset

of clients contributes local updates that are committed as transactions on the DAG and subsequently aggregated. Devices may join and leave freely; identities are not pre-vetted. The network operates asynchronously, and collusion among participants is possible.

Adversary and capabilities

An adversary can compromise a fraction of participants and coordinate their behavior across rounds. Let N be the total population and K the sample size per round. We denote by $M\%$ the fraction of malicious clients whose goal is to degrade the global model or bias aggregation via poisoned updates, and by $C\%$ the fraction exhibiting corrupted behavior inside the DAG (e.g., strategic attachments or collusion that amplify malicious influence). The adversary controls local training data and update generation for compromised clients, schedules message timing and DAG attachments to influence tip selection and confirmations, and issues misleading signals in any peer-derived trust path (bad-mouthing, ballot-stuffing, self-promotion) [154]. We assume the aggregator is not compromised and cryptographic primitives are not broken.

Objectives

The attacker aims to:

- Reduce model utility or stall convergence via poisoning.
- Increase the inclusion probability of malicious updates in the effective aggregation set.
- Manipulate trust assessments to down-rank honest peers and up-rank colluding ones.
- Distort the ledger structure (e.g., parasite-chain attempts) so that malicious transactions gain confirmations and survive selection.

Trust manipulation and Poisoning in scope

We focus on untargeted poisoning where the goal is to depress overall accuracy. Following prior work, we model label flipping as the primary mechanism: adversarial clients alter their own traning dataset D_k by remapping samples from class c_{src} to c_{tgt} before local training [271, 73, 133, 270]. Model/gradient perturbations consistent with this objective are

permitted. Targeted backdoors are acknowledged but not evaluated. Regarding trust, we consider classic reputation attacks: bad-mouthing (submitting adverse assessments to depress victims), ballot-stuffing (mutual promotion within a colluding set), and self-promotion (inflating one's own reliability) [154]. In the DAG setting, these behaviors materialize as corrupted behavior: compromised clients coordinate tip selection and timing to steer confirmation flows and cumulative weight, effectively implementing end-of-round parasite-chain attempts (Fig. 25). The goal is twofold: increase the probability that poisoned transactions are admitted into the effective aggregation set, and inflate the apparent trust of malicious peers through the additional confirmations accrued by the parasite chain.

Assumptions and constraints

The aggregator executes the prescribed protocol (honest-but-curious is not leveraged here); signatures and hash links prevent undetected tampering, but the adversary can choose when to publish and which tips to reference. Sybil capability is considered at the level of identity churn (e.g., newcomer attack), but unlimited identity fabrication is out of scope. Denial-of-service, privacy attacks (inference/membership), and physical compromise of the aggregator are excluded.

This setting aligns with FL poisoning and reputation-manipulation models in the literature (e.g., [258, 158]) while extending them with explicit DAG-structural adversarial behavior (corrupted attachments, parasite chains) that targets the confirmation dynamics rather than only the update content.

3.4.3 Related Works

To overcome the threats presented, trust management in IoT and edge computing has been explored through various paradigms, each differing in how trust is computed, propagated, and evaluated. In this section, we survey the literature across five key dimensions: non-DLT-based trust mechanisms in FL, TEE and committee-based models, blockchain-based and smart contracts approaches, DAG-based trust architectures, and poisoning-aware trust metrics for FL. This review highlights the lim-

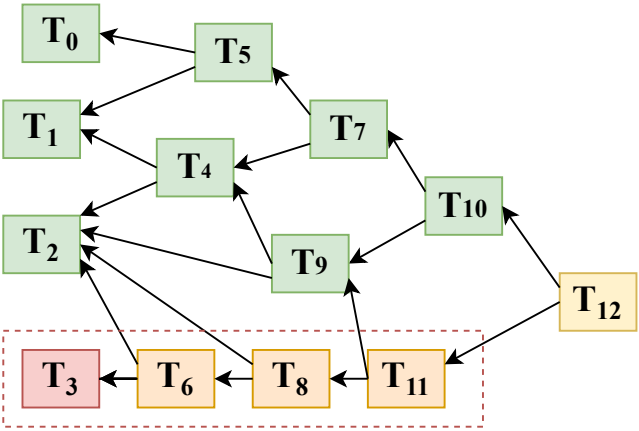


Figure 25: The parasite chain attack: T_{12} is a malicious node showing corrupted behavior, and referencing a parasite chain containing T_2, that is a poisoned model update.

itations of existing approaches and motivates the need for DAG-based trust mechanisms tailored specifically to open and heterogeneous FL environments in the IoT.

Several studies have addressed trust in FL without leveraging DLTs. For example, centralized models such as [43] assess trust on the server side based on client update behaviors, offering robustness against Byzantine clients but introducing a single point of failure. Other methods, like [237], use deep reinforcement learning to optimize task scheduling based on resource availability and trust, though they neglect poisoning detection and peer-to-peer reputational dynamics. Multi-server and clustering approaches [179, 91] promote decentralized trust evaluations based on distance metrics, while others [10] rely on Quality of Service (QoS) and behavioral analytics. Despite their contributions, these methods lack consistent and decentralized trust coordination, often resulting in conflicts due to the absence of shared, immutable knowledge [167].

TEE-based solutions [191, 324] and committee-based consensus mod-

Table 5: Contributions and limitations for different approaches in trust management for IoT and FL.

Key dimension	Related Works	Main Contribution	Limitations	FL
Non-DLT Trust Models	[43, 237, 179, 91, 10]	Centralized and clustering-based trust metrics (accuracy, QoS, update behavior)	Lack decentralization; limited or no poisoning detection; scalability and inconsistencies issues	✓
TEE and Committee-Based	[191, 324, 47]	Secure aggregation using TEEs or rotating committee voting	High computational cost; no tamper-proof trust record; limited scalability and applicability in FL	✓
Blockchain-Based Trust	[266, 307, 316, 323, 104, 285]	On-chain trust and incentive systems with smart contracts and custom consensus (PoR, PoTC)	Sequential block creation limits scalability and efficiency.	✓
DAG-Based TMM	[5, 306, 156, 61]	Scalable trust via DAGs in IoT/edge with parallel transaction validation	Not applied to FL; poisoning robustness not considered, and implemented only as a repository for the FL context.	✗
Poisoning-Aware Trust	[264, 172, 290, 328, 168]	Use Shapley values, update similarity, adversarial scoring, weighted aggregation	High computational cost; not combined with DAG-based TMMs; only used to compare global model and local model	✓

els [47] have also been proposed to enhance trustworthiness. TEEs offer hardware-level isolation but are often impractical for resource-constrained IoT devices, limiting their scalability. Committee-based approaches rely on rotating subsets of participants to validate updates, yet they do not benefit from persistent, tamper-resistant storage and lack mechanisms for global trust transparency, which are key requirements in open, decentralized FL.

Blockchain-based FL frameworks have gained significant attention due to their properties of immutability and auditability [266, 307, 316, 323]. These systems commonly employ smart contracts to automate aggregation, reward distribution, and reputation evaluation. Consensus mechanisms such as Proof-of-Reputation [104] and Proof-of-Trust Collaboration [320] aim to enhance trust resilience. However, traditional blockchain architectures suffer from limited throughput and high latency due to their sequential block validation and energy-intensive consensus protocols [164]. Furthermore, many of these models rely on simplistic trust metrics (e.g., participation frequency, model accuracy, data quality, or contribution size), often neglecting direct use of model poisoning detection as a trust metric against model poisoning or malicious contributions

[264, 172]. Smart contracts have also been employed to automate trust verification and incentivization [233, 103, 176]. While they introduce useful automation, these systems typically rely on static or coarse-grained trust metrics and fail to integrate dynamic adversarial robustness evaluations.

DAG-based DLTs such as IOTA [223], NANO [149], and Hedera [20] address many of blockchain's scalability challenges by enabling asynchronous and parallel transaction validation. Their applicability to trust management in IoT and edge computing has been demonstrated [5, 306, 156, 61]. In [180], the authors used IOTA as a ledger to access models published in IPFS, enabling a fully decentralized FL architecture without a central entity, but without considering the inclusion of a TMM. However, DAG has been exploited to provide TMM features: cumulative weights (CW) in DAGs have been interpreted as implicit trust signals [306], and reputation-driven tip selection strategies have been introduced [156]. Nonetheless, these systems have not yet been applied to FL scenarios, nor have they incorporated poisoning-aware or adversarially robust trust evaluations, which are critical in highly dynamic and untrusted learning settings.

Finally, recent works have sought to introduce robustness into FL trust evaluation [264, 172, 290, 328], utilizing techniques such as Shapley values [168], reputation-weighted model aggregation [292], and adversarial-aware scoring [242, 171]. These poisoning detection metrics have been applied to identify poisoning between the global model and a local model update. Although promising, these approaches are often computationally demanding and have not been integrated into ledger-based systems capable of decentralized coordination.

Table 5 presents a summary of the findings from the State of the Art. While significant efforts have been made to build trust in FL through centralized, committee-based, and blockchain-based strategies, they fall short in terms of scalability, decentralization, or robustness against adversarial behaviors, such as model poisoning. DAG-based DLTs emerge as a compelling alternative, offering high throughput, lightweight consensus, and asynchronous validation mechanisms. However, their integration into FL trust management remains unexplored. This work aims to bridge that

gap by proposing DAGTrustFL, a DAG-based trust architecture explicitly designed for open and resource-constrained FL environments in the IoT, enabling secure, scalable, and poisoning-resilient collaboration across heterogeneous devices.

3.4.4 Towards a trust framework for FL

The DAG structure provides an efficient DLT for recording trust-related data, enabling automatic and context-aware trust computation across IoT and edge devices. However, integrating these technologies can be challenging, as the complexity of the overall architecture leads to an increased attack surface, which can also compromise efficiency and limit scalability advantages. Despite the potential of DAG-based TMMs to address challenges in decentralized environments, the number of solutions in the literature remains limited and incomplete, exhibiting notable gaps [5]. In particular, the literature shows no dedicated architectures for managing trust in FL, which limits the use of general QoS indicators in trust evaluation. We state that these indicators alone are insufficient for detecting malicious behavior in FL environments. Moreover, these architectures often utilize DAGs to record all network interactions, resulting in high storage demands and inefficiencies that can particularly hinder scalability. Furthermore, ensuring equitable participation remains an issue, as nodes with higher trust may not receive proportionate rewards or influence in the consensus, potentially disincentivizing their active involvement [306]. For this reason, the proposed framework coherently integrates TMMs with a DAG ledger through two complementary components. The first adapts the tip-selection algorithm so that the DAG itself functions as a trust-management mechanism: attachments to tips are steered by contribution reliability indicators, shaping the flow of updates and enforcing, at the FL network layer, verification and attenuation of harmful behavior. The second introduces a smart-contract layer that moves coordination away from a single central aggregator toward a distributed, auditable logic, removing the SPoF and reducing any trust assumptions in the coordinator. The selection, weighting, and inclusion of updates into the global model are determined by on-chain, reproducible rules. Figure 30 presents the overall

architecture with a numerical example illustrating the interaction between these two components; technical details follow in the next sections.

Adapting the tip selection algorithm

Each peer evaluates trust independently, as trust is inherently subjective and often relies on indirect assessments when direct data is unavailable. IoT and edge devices actively manage trust in our framework by publishing their updated performance parameters on the DAG at the end of each FL round. Specifically, after each round of local training, each device generates a transaction containing its updated model parameters and attaches it to two previous transactions using an adapted tip selection algorithm. This algorithm incorporates trust evaluations to prevent connections with potentially malicious nodes. Unlike traditional approaches that rely solely on QoS data, our system integrates a context-aware trust indicator: the proximity of parameter updates [322]. This ensures that devices form connections only with nodes whose updates are consistent and not anomalous. In Fig. 30, the DAG shows an example of this mechanism, where honest peers do not connect their tips with nodes containing divergent values of parameter updates. However, Fig. 30 presents scalar values for simplicity, whereas, in practice, the proximity of model updates is assessed through matrix comparisons of updated parameters. For example, in the case of Convolutional Neural Networks, the matrices contain the weight of the network. Despite this simplification, the numerical example effectively represents the underlying mechanism of proximity-based update selection. The CW of a transaction, derived from the approvals it receives, reflects the confidence in the legitimacy of the associated node. A decay function dynamically adjusts CW over time, ensuring trust computations remain adaptive. If a node's adjusted CW falls below a predefined threshold, its trust score decreases, signaling potential malicious activity. Imagine the case in which a peer starts to publish model updates that are considered divergent; the aggregator, monitoring the adjusted CW, can exclude the malicious node from future FL rounds. This design enables the aggregator to extract trust metrics directly from the DAG. It also prevents bad-mouthing attacks: even if malicious nodes try to avoid connecting with legitimate nodes, they cannot alter the trust values of

those nodes stored in the DAG, making it easier for the aggregator to identify and flag suspicious activity. Each peer is incentivized to maintain a high trust score and avoid connections to nodes with anomaly values in the DAG, as this increases its chances of being selected for future FL training.

Smart Contract aggregation

The TMM proposed in our framework enables a central aggregator to perform model aggregation based on trust data from the DAG, ensuring that only contributions from trusted nodes are considered. However, this process can also be fully distributed through Smart Contracts, as shown in Fig. 26, providing automation, security, and transparency in trust evaluation. In this paradigm, the Smart Contract acts as the aggregator. At the end of each FL round, training nodes submit their model updates to the Smart Contract, which autonomously runs poisoning detection algorithms to identify adversarial contributions. Those algorithms can be based on weight matrix similarities, the detection of a substantial decline in accuracy after the aggregation of an update, and the evolution of the global model updated parameters [322]. The Smart Contract then computes trust values for each node by aggregating the anomaly detection result with direct and indirect trust evaluations from the DAG as a weighted sum, as illustrated in Fig. 27. The computation of trust scores can incorporate multiple indicators to ensure a nuanced evaluation of trust. These indicators include QoS factors such as participation frequency, activity level, and duration of engagement, as well as explicit mechanisms for detecting illicit behavior. Once computed, the trust scores are published on the DAG, ensuring an immutable and transparent record of each participant's trustworthiness. Depending on the results obtained, the respective model update will be included or excluded from the model aggregation. By automatically excluding updates from untrustworthy peers, the Smart Contract utilizes trust evaluations to enforce a selective aggregation process, ensuring that only non-poisoned updates contribute to the global model, mitigating the risk of model corruption and enhancing the reliability of the final aggregation.

This thesis focuses on the development of an architecture centered

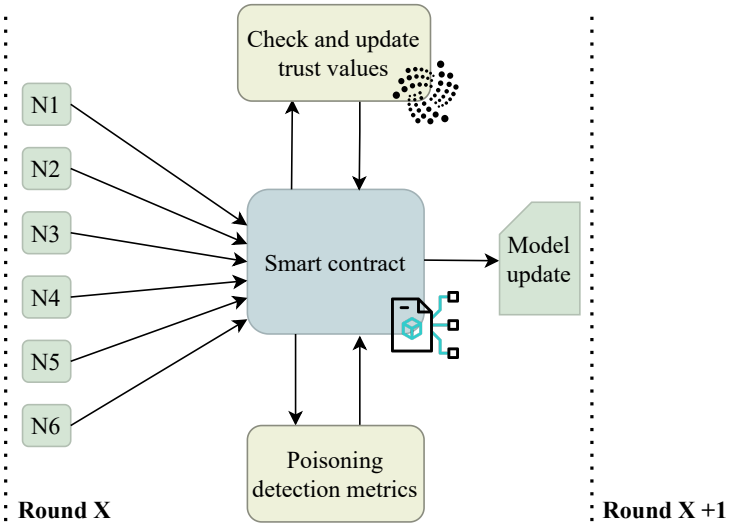


Figure 26: Model update on a round of FL in our framework. This example shows six nodes sending updates to the Smart Contract to produce the model aggregation based on trust evaluations.

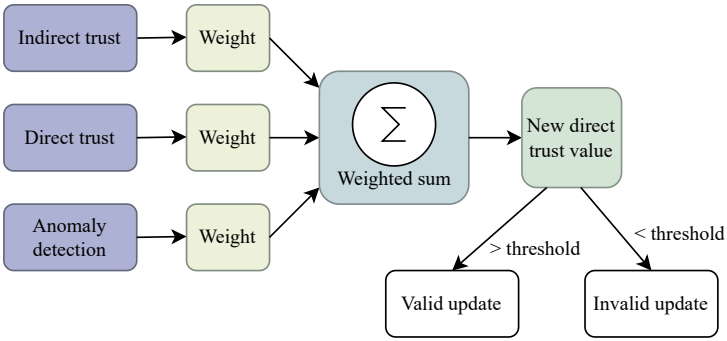


Figure 27: How the trust computation is done inside the Smart Contract.

on the first design path, specifically adapting the tip-selection algorithm, while deferring Smart Contract aggregation to future work. The rationale is twofold. First, tip selection is the pivotal mechanism that determines which updates the DAG advances, making it the natural locus for embedding trust signals and, in effect, turning the ledger into a trust management mechanism. Second, decentralised aggregation via smart contracts primarily removes reliance on a single central aggregator, an avenue already explored extensively in the literature [132, 250]. Although it is valuable for eliminating assumptions about coordinator honesty and exploring its integration with a DAG-ledger technology, it first needs the establishment of the DAG itself as the engine of trust.

3.4.5 DAGTrustFL System Model

To facilitate the reader's understanding of the DAGTrustFL model, we refer to the components illustrated in Fig. 29 and Fig. 30, labeled with letters ranging from A to H.

Transaction Structure

Each peer p_i publishes a transaction ψ_i to the DAG at the end of a local training round (component A). Building from [306], [156] and [258], this transaction encodes:

$$\psi_i = \langle \Delta w_i, t_i, \gamma_i, \beta_i \rangle \tag{3.5}$$

where:

- $\Delta w_i \in \mathbb{R}^d$ is the local model update vector;
- $t_i \in \mathbb{R}$ is the timestamp of the round;
- $\gamma_i = \{\psi_{j_1}, \psi_{j_2}\} \subseteq \mathcal{G}$ are the parent transactions approved by ψ_i;
- β_i is the approval component;

The approval component β_i is defined as:

$$\beta_i = \left\langle \text{DevID}_i, \{\text{PoW}_i^{\text{nonce}}\}, \text{PoW}_i^{\text{target}}, \text{Sig}_i \right\rangle \tag{3.6}$$

where:

- DevID_i is the identifier of the node issuing the transaction;
- $\text{PoW}_i^{\text{nonce}} \in \mathbb{N}$ is the proof-of-work (PoW) nonce used to validate the issuance of ψ_i;
- $\text{PoW}_i^{\text{target}} \in \mathbb{R}^+$ is the difficulty threshold that must be satisfied by the nonce.
- Sig_i is a digital signature ensuring authenticity and integrity.

This lightweight PoW mechanism is introduced to mitigate spam and Sybil attacks by requiring minimal computational effort from participating devices [54]. Each node must find a nonce such that the hash of the transaction metadata, including its parent references, satisfies a predefined PoW target. The target is intentionally set to a low difficulty level, ensuring that the process remains computationally lightweight and suitable for resource-constrained IoT or edge devices.

Transaction Verification Procedure

To preserve the integrity of the DAG and prevent the injection of malicious or redundant model updates, each node must verify any newly received transaction ψ_i before accepting it (component B. The verification procedure in our framework consists of three main steps:

1. Uniqueness Check: The verifier ensures that the issuing peer p_i has not already submitted another update in the current FL round. This prevents update duplication and ensures fair participation.
2. Validation of Approval Structure γ_i must reference valid parent tips $\{\psi_{j_1}, \psi_{j_2}\} \subseteq \mathcal{G}$, and the included PoW nonce and Signature in β_i are verified.
3. Aggregation to the local DAG: If the transaction passes the above checks, the transaction is added to the local DAG structure.

Cumulative Weight and Endorsement

Whereas in existing DAG-based DLTs, it is essential to strictly order transactions in the DAG, mainly due to financial requirements, and in our approach, this sequence ordering is not necessary, as also denoted by [156]. However, we need to reach consensus on confirmed transactions. Each transaction ψ_i accumulates trust over time based on direct or indirect approvals from other transactions.

Let $\mathcal{G}_t = (\Psi, E)$ denote the DAG at round t, where Ψ is the set of transactions and E the set of approval edges. The future cone τ_i of a transaction $\psi_i \in \Psi$ is defined as the subset of $\mathcal{G}_t$ that contains all the nodes that directly or indirectly approve ψ_i:

$$\tau_i = \{\psi_i\} \cup \bigcup_{\psi_j \to \psi_i} \tau_j \tag{3.7}$$

$$\text{where} \quad \psi_j \to \psi_i \text{ represents a direct approval in } \mathcal{G}_t$$

The cumulative weight $CW_i \in \mathbb{N}$ (component C) of a transaction $\psi_i \in \Psi$ is defined as:

$$CW_i = \sum_{\psi_j \in \tau_i} w_j \tag{3.8}$$

where w_j is the static weight of the transaction ψ_j, and τ_i is the future cone of ψ_i. When a transaction achieves a CW that surpasses a threshold Θ, we consider the transaction confirmed. An adjacency matrix can help in the calculation of CW [61]. In our case, we consider a weight $w = 1$ for all the transactions, leaving the possibility to exploit this feature to improve the model in future works.

Distance metric

To detect poisoning attacks, the distance metric function $D(\cdot, \cdot)$ can be defined using various methodologies. In literature, cosine similarity [133, 268, 165] and Euclidean distance [42, 294, 159] are common and quite effective when identifying poisoning by comparing the global model and

a local model update. However, in our system, the poisoning detection is distributed in the network; thus, we need to compare local models with each other. We focused on cosine similarity.

Cosine similarity is evaluated specifically on the last layers' gradient to highlight the divergence between gradients from honest participants and those from adversaries, due to their opposing objectives. Formally, the cosine similarity between two gradients ∇_i and ∇_j is expressed as:

$$cs(\nabla_i, \nabla_j) = \cos\phi = \frac{\nabla_i \cdot \nabla_j}{\|\nabla_i\| \cdot \|\nabla_j\|}. \tag{3.9}$$

A normalized cosine similarity that ranges from 0 (opposite direction) to 1 (identical direction) can be defined as:

$$cs_{\text{norm}}(\nabla_i, \nabla_j) = \frac{1 + cs(\nabla_i, \nabla_j)}{2} \tag{3.10}$$

where $cs(\nabla_i, \nabla_j)$ is the standard cosine similarity. This metric assesses the angular alignment between gradients. Thus, the cosine dissimilarity between two gradients ∇_i and ∇_j is defined as one minus their cosine similarity:

$$cd(\nabla_i, \nabla_j) = 1 - cs(\nabla_i, \nabla_j) \tag{3.11}$$

This measure ranges from 0, when the vectors are perfectly aligned, to 2, when they are diametrically opposed. Therefore, the normalized cosine dissimilarity is defines as:

$$cd_{\text{norm}}(\nabla_i, \nabla_j) = 1 - cs_{\text{norm}}(\nabla_i, \nabla_j) \tag{3.12}$$

The Euclidean distance [42] between two model updates Δw_i and Δw_j is given by:

$$Dis(\Delta w_i, \Delta w_j) = \sqrt{\sum_k \left(\Delta w_i(k) - \Delta w_j(k)\right)^2} \tag{3.13}$$

where $\text{Dis}(\cdot, \cdot) \in [0, \infty]$ is directly proportional to the distance between Δw_i and Δw_j.

A normalized Euclidean distance between two models Δw_i and Δw_j can be defined as:

$$Dis_{\text{norm}}(M_1, M_2) = \frac{Dis(M_1, M_2)}{Dis(M_1, 0) + Dis(M_2, 0)} \tag{3.14}$$

where $Dis(\Delta w_i, \Delta w_j)$ is the standard Euclidean distance, and 0 is the zero vector. $Dis_{\text{norm}}(\cdot, \cdot) \in [0, 1]$, with value 0 that indicates identical models and values closer to 1 indicate greater dissimilarity.

Empirical studies have demonstrated that the Euclidean Distances computed between benign models are typically much smaller than those involving poisoned models [42]. However, cosine dissimilarity tends to be more resilient than Euclidean distance. Attackers may attempt to scale their gradient updates to bypass detection from Euclidean distance methods, but they must preserve the directionality of their gradients to fulfill their goals and not be detected by cosine dissimilarity [133].

Moreover, [304] observes that the weights of the last layer in a model update are particularly sensitive to the local data distribution, making them more informative for detecting poisoning attacks such as label flipping. Leveraging this insight, the authors suggest that the comparison of only the last layer's weights, rather than the entire model, enables a clearer separation between benign and malicious updates. In the case of Convolutional Neural Networks (CNNs), this corresponds to the Fully Connected (FC) layer, which most effectively highlights discrepancies introduced by label flipping attacks, as the malicious updates tend to appear as outliers with respect to those from honest clients.

Percentile-Based Outlier Detection

Mean-based thresholds outlier-detection techniques in federated learning are particularly vulnerable to adversarial manipulation and extreme values [146]. While setting a fixed threshold on cosine similarity or Euclidean distance might initially distinguish benign from malicious updates, this gap diminishes as training progresses and model parameters consolidate. Consequently, a dynamic thresholding strategy becomes essential.

To address this, DAGTrustFL adopts a percentile-based approach that dynamically adapts thresholds using robust statistical measures. In each federated learning round, similarity metrics are calculated across all participant updates. We employ the Interquartile Range (IQR), calculated as the difference between the first and the third quartiles: $\text{IQR} = Q_{75} - Q_{25}$ The IQR is used to derive a dynamic threshold:

$\theta_{outlier}(t) = Q_{50} - IQR - \delta$

where Q_{50} represents the median quartile and $\delta = 0.05$ serves as a conservative buffer to reduce false positives. This method offers robustness to significantly divergent values, adaptivity to the evolving distribution of updates, and statistical soundness [275].

Redefined Tip Selection Strategy

To integrate both semantic similarity and structural validation, the tip selection process in DAGTrustFL is adapted to consider both the model proximity and the Cumulative Weight (CW) of existing tips. This dual condition ensures that each new transaction attaches to the most trustworthy and contextually aligned updates while maintaining consistency in DAG growth. To avoid the connection to malicious nodes inside the graph, the peer selects two parent nodes to connect that are not too old, to incentivize the growth of the graph in a linear way, and that have an update that should not be poisoned. To identify the poisoning, recent works compare the global update to the local ones to detect discrepancies. In our case, instead, we compare the local model of peer p_i with the nodes of the other peers that have a tip inside the graph: i.e., participated in this FL round and have uploaded an update (component D).

Let $L(t_i) \subset \mathcal{G}$ be the set of current tips in the DAG at time t_i, and Δw_i the model update of the new transaction ψ_i. The selection of the parent transactions $\gamma_i = \{\psi_{j_1}, \psi_{j_2}\}$ is guided by the following logic:

1. Model similarity: Prefer tips whose model updates are semantically close to Δw_i and are not considered outliers;

2. Cumulative weight: Consider only unconfirmed tips with $CW_j < \Theta$ as primary candidates;

3. Fallback: If no candidate satisfies the similarity constraint, allow attachment to confirmed nodes.

The selection mechanism is formally defined as:

$$\gamma_i = \underset{\substack{\psi_j \in L(t_i) \\ CW_j < \Theta}}{\arg\min} (D(\Delta w_i, \Delta w_j) \cdot \varsigma \cdot CW_j) \quad \text{where} \quad D(\Delta w_i, \Delta w_j) > \theta_{outlier}(t) \tag{3.15}$$

where:

- $D(\cdot, \cdot)$ is the distance metric, which can be implemented by the normalized Euclidean Distance or the normalized cosine dissimilarity;
- Θ is the Cumulative Weight threshold distinguishing confirmed from unconfirmed tips;
- CW_i is the Cumulative Weight of the transaction
- $\theta_{outlier}(t)$ is the dynamic percentile-based outlier threshold
- $\varsigma > 0$ is a scalar weight for penalty factor to avoid connection to old transaction (i.e., with higher CW)

If there are no such $\gamma_i = \{\psi_{j_1}, \psi_{j_2}\} \in L(t_i)$ items that meet the similarity constraint, we can search for eligible parents to connect by looking at the entire graph $\mathcal{G}$. Thus, this fallback rule is invoked:

$$\gamma_i = \underset{\psi_j \in \mathcal{G}}{\arg\min} (D(\Delta w_i, \Delta w_j) \cdot \varsigma \cdot CW_j) \quad \text{where} \quad D(\Delta w_i, \Delta w_j) > \theta_{outlier}(t) \tag{3.16}$$

For the selected tips $\gamma_i = \{\psi_{j_1}, \psi_{j_2}\}$, the algorithm examines a part of the corresponding past cone to prevent the approval of potentially malicious transactions done by corrupted behaviors, as discussed in the Assumptions and Threat Model section. Let $\mathcal{G}_t = (\Psi, E)$ denote the DAG at round t, where Ψ is the set of transactions and E the set of approval edges. The past cone ρ_i of a transaction $\psi_i \in \Psi$ is defined as the subset of $\mathcal{G}_t$ that contains all nodes directly or indirectly approved by ψ_i:

$$\rho_i = \{\psi_i\} \cup \bigcup_{\psi_i \to \psi_j} \rho_j \tag{3.17}$$

where $\psi_i \to \psi_j$ represents a direct approval in $\mathcal{G}_t$

A graphic example of past and future cones is shown in Figure 28. Determining the appropriate depth χ of the past cone to inspect is crucial for balancing the computational cost of the tip selection algorithm with the effectiveness of attack detection within the DAG. This depth depends on the number of peers participating in each federated learning round, denoted as K, since each round is expected to contribute K transactions to the DAG. Given that the heaviest subgraph is selected at the end of each round, the system can tolerate up to, but not including, 50% malicious peers. In the worst-case scenario, where an attacker manages to introduce a malicious update referenced exclusively by corrupted nodes, preventing the inclusion of this malicious path requires examining a past cone of depth $\chi = \frac{K}{2}$. Increasing χ beyond this threshold does not yield additional security benefits.

For each node within the inspected past cone, the distance metric is computed between the update associated with the node ψ_k and that of each of its parent nodes $\{\psi_{k_1}, \psi_{k_2}\}$. If, for any parent-child pair, the condition $D(\Delta w_k, \Delta w_{k_x}) < \Gamma(t) - \epsilon$ is not satisfied, the corresponding path is excluded from consideration, and the origin transaction ψ_{j_x} is discarded from the parent selection, as the node did not show a trustworthy behavior and did not follow the logic of the tip selection algorithm. In such cases, an alternative tip is selected from the previously identified candidates, and the procedure is reiterated.

Heaviest Subgraph Selection for Secure Aggregation

To ensure consistency and filter out malicious or semantically divergent branches in the DAG, our model uses a key mechanism that identifies the heaviest subgraph rooted at one of the tips at the end of each FL round. This selected subgraph is computed by the Aggregator (component E) and serves as the trusted context for determining which transactions are considered confirmed and eligible for aggregation.

Let $\mathcal{G}_t = (\Psi, E)$ denote the DAG at round t, where Ψ is the set of transactions and E the set of approval edges. Let $\mathcal{L}(t) \subset \Psi$ be the current set of tips.

For each tip $\psi_l \in \mathcal{L}(t)$, we compute the past cone ρ_i, and for each set,

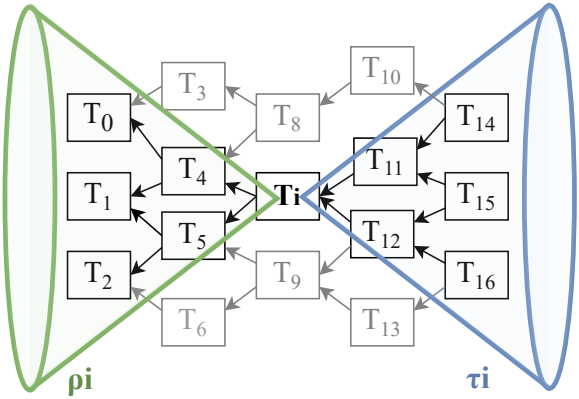


Figure 28: The past cone ρ_i and the future cone τ_i of a transaction T_i in the DAG.

we compute the weight of those subgraphs. The weight of the subgraph is computed as:

$$W(\rho_i) = \sum_{\psi_j \in \rho_i} CW_j \tag{3.18}$$

where CW_j is the cumulative weight assigned to the transaction ψ_j, reflecting participation and reputation.

The heaviest subgraph $\mathcal{G}^* \subseteq \mathcal{G}_t$ is then defined as:

$$\mathcal{G}^* = \arg\max_{\psi_l \in \mathcal{L}(t)} W(\rho_l) \tag{3.19}$$

Then, the Aggregator computes the corrupted behavior check of the tip selection algorithm on the elected tip to check that the last node did not attach a parasite chain to the graph. If anomalous behavior is detected, the node is discarded, and the process is reiterated. If no anomalous behavior is detected, from now on, only transactions in $\mathcal{G}^*$ are considered for downstream operations such as tip confirmation, trust computation, and global model aggregation. Transactions outside $\mathcal{G}^*$ are discarded

as unendorsed or inconsistent, thus enhancing the robustness of the FL pipeline by filtering out isolated or adversarial forks.

This key strategy resembles the "longest chain" rule used in blockchain but is adapted to the DAG topology by selecting the path with the highest cumulative trust weight, in a similar way to the Tangle [223]. It enforces convergence around a single, trustworthy subgraph that discards malicious transactions and enables the generation of Milestones at the end of every FL round (component F). Therefore, a new milestone is generated, a special transaction that confirms all the previous transactions and that contains, as a model update, the new global model generated by the central aggregator, and that will be considered as the first tip to connect in the following FL round.

Trust Metrics from the Aggregator

The model we have described so far allows for the creation of a graph that, in each round, filters out outliers and updates anomalies, leveraging trust votes that each peer demonstrates by connecting to specific nodes in the graph. However, to enable a high-level perspective over the trustworthiness among peers in the network, the aggregator makes its own trust evaluation based on direct and indirect trust (component G). Direct trust is performed by checking the behavior of nodes inside the graph; indirect trust comes from the edges of the graph that resemble trust manifestation. Similar to what has already been done by [156], we define a trust score μ_i that the aggregator assigns to each peer p_i at the end of each FL round, as a weighted sum of two components: structural consistency σ_i and update contribution η_i.

Structural Consistency (σ_i)

To reward structural consistency inside the graph and promote the correct use of the tip selection algorithm, we enforce a hard threshold on the similarity values in the inspected past cone ρ_i. If any node in the parent subgraph has a similarity below the threshold τ, the consistency is set to zero, as the behavior of the node inside the graph is not legitimate and

does not follow the logic of the tip selection algorithm:

$$\sigma_i(t) = \begin{cases} 0 & \text{if } \exists \psi_j \in \rho_i : D(\Delta w_i, \Delta w_j) > \Gamma(t) - \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{3.20}$$

where $\mathrm{D}(\cdot, \cdot) \in [0, 1]$ is a normalized distance metric function, implemented by normalized cosine dissimilarity or normalized Euclidean distance. The selection of the depth χ for the inspected past cone ρ_i reflects a balance between computational efficiency and security, and should always be less than both $\frac{K}{2}$ and the depth employed in the tip selection algorithm. In practice, a depth of 1 may suffice to flag malicious behavior, since the detection of corrupted activity is expected to be addressed in a distributed manner by the peers. Nevertheless, adopting a depth greater than 1 can further penalize corrupted actions in the computation of the composite trust score.

Updates Contribution (η_i)

To penalize peers that had discarded transactions, as they probably behaved maliciously, and to increase the reputation of active nodes with respect to inactive ones, we define η_i as:

$$\eta_i(t) = \begin{cases} 1 & \text{if } \exists \psi_i \in \mathcal{G}^* \\ \begin{cases} 0 & \text{if } \exists \psi_i \in \mathcal{G} \\ \eta_i(t-1) & \text{otherwise} \end{cases} & \text{otherwise} \end{cases} \tag{3.21}$$

where $\mathcal{G}^*$ is the heaviest subgraph of the current DAG, $\mathcal{G}$ is the current DAG, and ψ_i refers to transactions done by the peer p_i for which the trust component is being calculated. This way, if p_i has a transaction outside the heaviest graph, it will be penalized with a 0 value for this trust component; if it did not participate in this round of FL, the last score will be kept; and if the transaction is part of the heaviest graph, the score will be maximum.

Composite Trust Score

The overall trust score $\mu_i \in [0, 1]$ for the peer p_i is given by the composition of update contributions and structural consistency:

$$\mu_i = \upsilon_1 \cdot \sigma_i + \upsilon_2 \cdot \eta_i \quad \text{with} \quad \upsilon_1 + \upsilon_2 = 1 \tag{3.22}$$

Trust scores are periodically refreshed to reflect recent behavior. Let $\mu_i^{(t)}$ be the trust score at round t:

$$\mu_i^{(t)} = \lambda \cdot \mu_i^{(t-1)} + (1 - \lambda) \cdot \mu_i^{\text{new}} \tag{3.23}$$

where $\lambda \in [0, 1]$ controls the influence of historical versus current observations, as the magnitude of a decay function, and serves to maintain dynamism in the trust system.

Trust-Weighted Federated Aggregation

In DAGTrustFL, the composite trust score μ_i is directly incorporated into the aggregation phase of the FL process. Certain updates in the DAG could be poisoned even if we chose the heaviest sub-graph and tried to avoid corrupted behaviors, for example, if the depth χ of the examined past cone during the tip selection algorithm is not enough. At the end of each round, the global model update is computed as a trust-weighted average of the confirmed local model updates in the DAG (component H).

Let K denote the number of peers selected for aggregation, and Δw_i the model update from peer p_i that pertains to transactions in the heaviest sub-graph of the DAG, denoted as $\mathcal{G}^*$. Elaborating upon the commonly used aggregation strategy of FedAvg [184], our trust-weighted global aggregation is defined as:

$$\Delta w_{\text{global}} = \frac{1}{M} \sum_{i=1}^{K} \mu_i \cdot \Delta w_i, \quad \text{where} \quad M = \sum_{i=1}^{K} \mu_i \tag{3.24}$$

This ensures that model updates from more trustworthy peers (i.e., with higher μ_i) have a greater influence on the global model. It generalizes the standard FedAvg rule by replacing uniform weighting with a dynamic, reputation-based approach that does not require a threshold to be set. Thus, updates from peers with low cumulative trust will have limited or no impact on the aggregated model. M serves to normalize the influence across peers.

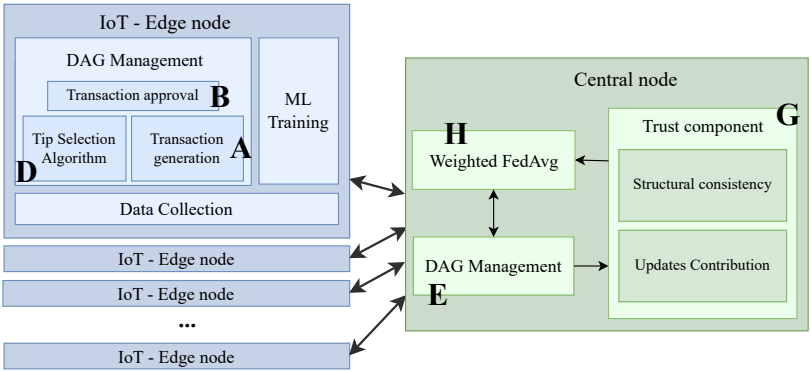


Figure 29: The architecture of DAGTrustFL: interaction between components.

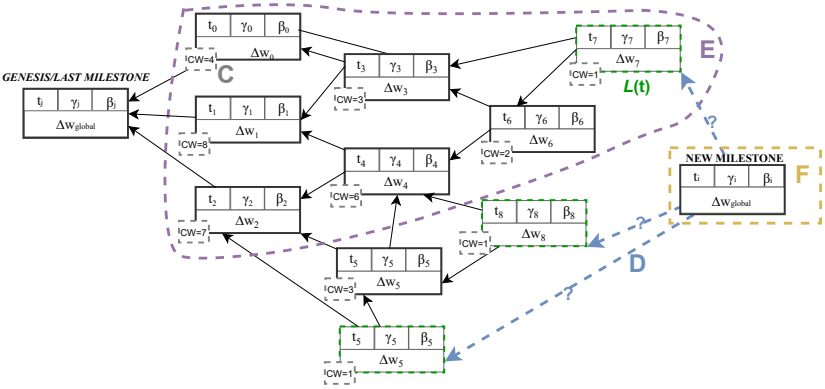


Figure 30: A single FL round inside the DAG, with references to the various components of the model. Each transaction contains the timestamp t_i, the approved parents $\gamma_i = \{\psi_{j_1}, \psi_{j_2}\}$, the approval component β_i, and the model update Δw_i.

3.4.6 Experimental evaluation

Goals

We empirically test whether DAGTrustFL trust layer filters malicious contributions as adversarial pressure increases, preserves model utility compared to standard FL, and adds acceptable overhead. We answer these via robustness/selectivity, utility, and cost/scalability measurements.

Setup

Environment. One Dockerized services run on the same host containing an FL orchestrator (Env-1) and a DAG simulator (Env-2). The source code of the experimental evaluation is available in our GitHub project[12]. Python 3.12 with PyTorch is used throughout. The host has a CPU 13th Gen Intel Core i9-14900K x 32 and as a GPU an NVIDIA Titan RTX, 24GB.

Data/Model. We used Fashion-MNIST [299] with the standard split and the reference CNN implementation from [270].

Federated Learning Environment (Env-1). We extend the poisoning-ready FL codebase [270] to: compute cosine/Euclidean similarities; export a per-round trust vector via get_trust_vector(); and perform trust-weighted FedAvg (Eq. (20)). Unless stated, total FL rounds are $R = 100$, local epochs are $E = 1$, batch size is $B = 10$, and learning rate is $\mathrm{lr} = 10^{-3}$, following the configuration suggested by [270]. While smaller batches increase per-round training time due to the higher number of iterations, they also inject stochastic noise into the gradient, which has been shown to improve generalization and stabilize training in non-IID federated settings [44].

DAG Environment (Env-2). Implemented with the networkx Python library. Each transaction encodes $\langle \Delta w_i, t_i, \gamma_i, \beta_i \rangle$ as in Sec. IV-A; The revised tip-selection is implemented with percentile/IQR thresholding, heaviest-subgraph selection, milestone creation, and trust scoring $(\sigma_i, \eta_i, \mu_i)$. The DAG grows across rounds; since trust is persisted in the trust_vector, we prune stale graph state every 25 rounds to bound memory and traversal cost.

[12] https://github.com/UniboSecurityResearch/DAGTrustFL

Threat Model. We vary the ratio of malicious ($M\%$) and corrupted ($C\%$) clients among the N clients. We based the attacks on the label-flipping attack, as specified in the threat model (Sec. III). We also test collusion, i.e., corrupted behavior inside the DAG, and parasite-chain attempt at the round end.

Baseline and Ablations

Baseline: FedAvg (uniform) [184] with Poisoning (with no countermeasures) [270].

Ablations:

1. Static distance threshold: replace IQR with fixed θ;
2. No central trust evaluation: normal FedAvg over G^* (no trust weights);
3. No history: disable decay $\lambda = 0$;
4. No corrupted-behavior check: disable past-cone validation ($\chi = 0$).

Performance metrics

We report false negative rate (FNR), false positive rate (FPR), and Detection Accuracy (DACC):

$$\text{FNR} = \frac{\#\{\text{evil tx in } G^*\}}{\#\{\text{evil tx submitted}\}} \tag{3.25}$$

$$\text{FPR} = \frac{\#\{\text{benign tx not in } G^*\}}{\#\{\text{benign tx submitted}\}} \tag{3.26}$$

$$\text{DACC} = \frac{\#\{\text{evil tx not in } G^*\} + \#\{\text{bening tx in } G^*\}}{\#\{\text{total tx submitted}\}} \tag{3.27}$$

Utility is Test Accuracy (MACC) at round R. Cost includes DAG time/round (tip selection + G^* + trust computations + milestone). Hyperparameters and defaults appear in Table 6.

Hypotheses

H1 (Ablations). Removing central trust or components (λ, χ, ς) degrades DACC, increases FNR/FPR; static thresholds underperform IQR.

H2 (Robustness/Selectivity). FPR grows slowly with $(M+C)\%$; FNR remains low ($< 15\%$) up to moderate $M\%$.

H3 (Utility). Under poisoning, MACC and convergence are better than

Table 6: Hyperparameters and default values used in the experimental evaluation.

Parameter	Sweep values	Default
Adversarial ratio $(C + M)\%$	$\{0, 10, 20, 30, 40\}\%$	–
CW penalty factor (ς)	$\{0.00, 0.02, 0.05\}$	0.02
Past-cone depth (χ)	$\{0, 1, \lfloor K/4 \rfloor, \lfloor K/3 \rfloor, \lfloor K/2 \rfloor\}$	$\lfloor K/4 \rfloor$
CW threshold (Θ)	$\{2, 3, 4, 5, 6, 7, 8, C, C + 1\}$	4
Decay factor (λ)	$\{0.0, 0.1, 0.3, 0.5, 0.7, 0.9\}$	0.1
IQR buffer δ	–	0.05
Rounds (R)	–	100
Total clients (N)	$\{10, 20, 40, 60, 80, 100, 200\}$	10
Clients per round (K)	$\{N - 1, N/2\}$	$N - 1$
Local epochs (E)	–	1
Batch size (B)	–	10
Trust weights (υ_1, υ_2)	–	$(0.6, 0.4)$

baseline; trust-weighted FedAvg mitigates residual inclusions; malicious peers earn lower trust.
H4 (Cost/Scalability). DAG time/round is small vs. local training; scaling with participants is near-linear to sub-quadratic.

3.4.7 Results and Findings

H1 — Ablations

Past-cone depth (χ). Small χ (e.g., 1 or $\lfloor K/4 \rfloor$) provides the best FNR/FPR trade-off; χ=0 fails in the presence of corrupted links; large χ increases FPR due to over-screening (Fig. 35a). Decay (λ). History smooths trust, but in our runs λ=0.1 performs marginally best (Fig. 35b). No central trust. Removing trust-weighted FedAvg degrades robustness, especially as $M\%$ grows (Fig. 35f). IQR vs static θ. The adaptive percentile range tracks the evolving similarity distribution and isolates poisoned updates more reliably than any fixed θ (Fig. 35e). CW threshold (Θ) and penalty (ς). Θ>4 avoids premature confirmations; ς around 0.25 biases toward recent tips and improves metrics (Figs. 35d, 35c).

Finding: H1 confirmed.

H2 — Robustness and Selectivity

Across $(M+C)\% \in \{0, 10, 20, 30, 40\}$ DAGTrustFL keeps FNR/FPR low at mild–moderate threat and degrades gracefully at high threat. With C=0 and M=$\{10, 20\}$ we obtain DACC $\geq$ 97.9% and $\geq$ 99.1% respectively; FNR remain under 15% in any of the tested scenarios. With corruption present, DACC stays competitive across settings (Table 7).
Finding: H2 confirmed.

H3 — Utility

Final MACC is the same without poisoning and remains consistently better than baseline at increasing $(M+C)\%$ (Fig. 31; Table 7); trust values separate honest and malicious peers over all scenarios (Figs 32).
Finding: H3 confirmed.

Table 7: DAGTrustFL metrics ordered by total threat percentage $(M + C)\%$.

M%	C%	(M+C)%	MACC%	FPR%	FNR%	DACC%
0	0	0	91.69	0.11	0.00	99.89
10	0	10	91.23	1.61	7.05	97.86
10	10	20	90.81	0.14	8.23	96.30
20	0	20	90.99	0.02	1.11	99.15
10	20	30	91.06	4.64	11.11	91.36
20	10	30	90.98	6.47	1.75	94.95
30	0	30	90.99	17.71	13.85	76.66
10	30	40	90.84	2.44	3.37	95.29
20	20	40	90.59	5.01	7.73	94.05
30	10	40	89.87	33.45	6.17	77.67
40	0	40	90.14	39.09	3.90	75.08

H4 — Cost and Scalability

Per-round time Mean ML time/round $\approx$ 17.86s; DAG time/round $\approx$ 2.05s (device-DAG time + aggregator-DAT time). Thus, ledger overhead

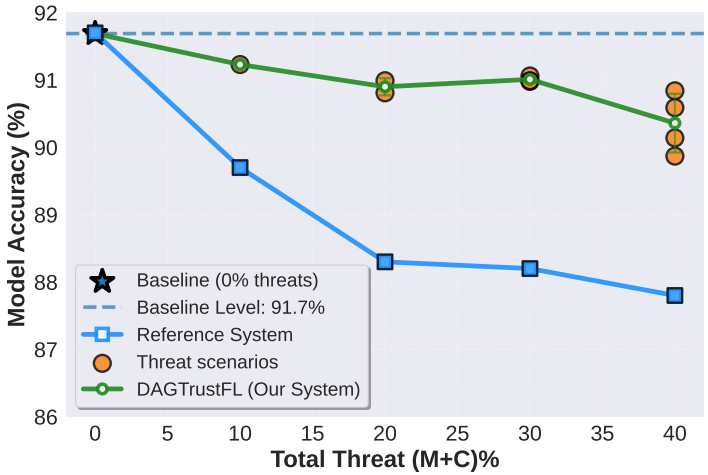


Figure 31: Model accuracy varying (M+C)%: our system compared to reference [270].

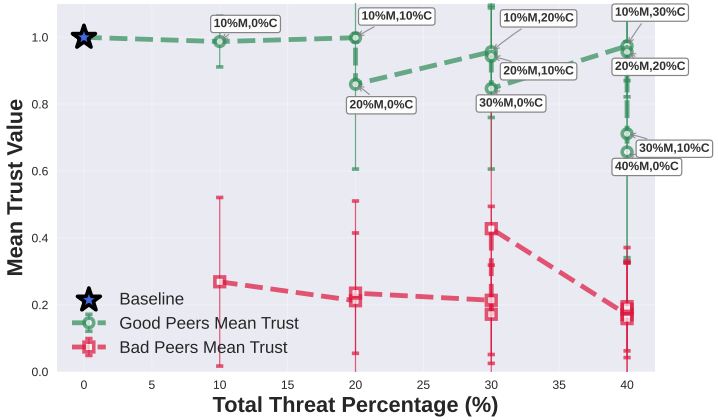


Figure 32: Mean trust values of honest and malicious peers by varying (M+C)%.

is a fraction of training time, and the majority of DAG time is spent by the Aggregator elaborating the trust vector, leaving a low load to the nodes of the FL system. Scalability We assess ledger-layer scalability by increasing the number of participating clients (and proportionally the per-round sample K) while keeping payload size and hyperparameters fixed, and we record the DAG time per round (tip selection, G^* selection, and trust scoring), and the device-side attachment latency for each transaction. In DAG-based ledgers, validation and referencing proceed asynchronously and in parallel, so throughput tracks the arrival rate of transactions, and the per-transaction latency remains approximately stable over the operational range [41, 257]. Consequently, as K grows, the total DAG time per round should increase roughly linearly (more transactions to process) while the per-transaction latency stays nearly constant. By contrast, blockchains with sequential block production (PoW/PoS) exhibit a capacity ceiling: let T_b be the average block interval and L the block capacity, the maximal sustainable throughput is $\mathrm{TPS}_{\max} \approx L/T_b$; once the transaction arrival rate $\lambda(N)$ exceeds this ceiling, a queue builds up and waiting time increases with the excess load [305]. We denote by N_b the smallest population such that $\lambda(N_b) = \mathrm{TPS}_{\max}$. This value depends on the PoW, or PoS methods; for example, it results in around 7 for Bitcoin and 30 for Ethereum [305].

Empirically, our measurements match these expectations. The device-side attachment latency is about 5 ms per transaction (R=100, N=10), i.e., negligible compared to local training. Moreover, 94.3% of local training completions fall within the same one-second window across devices, so updates arrive in short bursts; even under bursty arrivals, the DAG maintains approximately constant per-transaction latency. In an equivalent blockchain, assuming a PoW time similar to the device-side attachment latency in the DAG, the same burst pushes $\lambda(N) = \mathrm{TPS}_{\max}$ beyond N_b, resulting in queueing delay and increasing confirmation times. The comparative view in Fig. 34 captures this effect: blockchain throughput flattens at $\mathrm{TPS}_{\max}$ and latency rises once $K > N_b$, whereas the DAG curve continues to scale with K in our tested regime. Other than the DAG itself, the trust-weighted tip selection algorithm demonstrates scal-

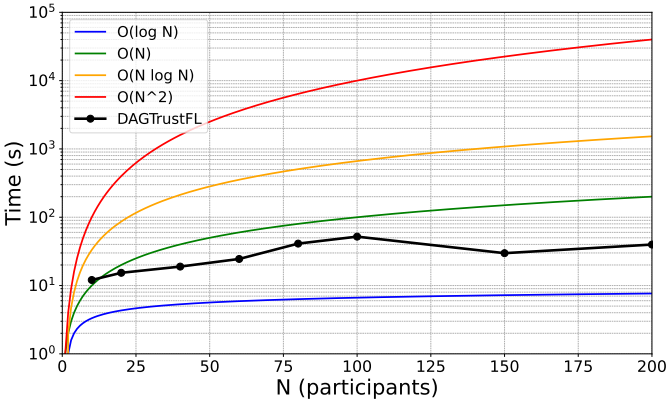


Figure 33: Mean DAG time per round (s) by varying the number of FL participants, and comparisons with scaling references: $O(\log(N)), O(N), O(N\log(N)), O(N^2)$.

able performance: Figure 33 reveals that the total DAG time per round grows between $O(\log N)$ and $O(N)$. This trend highlights that, beyond the intrinsic scalability of the DAG structure, the computation of the trust vector is itself efficient, contributing to the overall scalability of the system rather than limiting it.
Finding: H4 supported.

3.4.8 Limitations and Future Works

Our evaluation reveals some limitations that bound the generality of our claims. First, the poisoning filter ultimately relies on distance-based separability of client updates (cosine/Euclidean). Under higher adversarial pressure, these metrics can become less discriminative, and robustness degrades accordingly; hence, the security of the whole pipeline hinges on how well the chosen similarity captures attack artifacts across rounds and layers. Second, our threat model focuses on label-flipping and corrupted attachment behavior inside the DAG under an honest, uncompromised aggregator; we did not test backdoors, gradient-sign or model-scaling attacks, adaptive evasion that targets the percentile range itself,

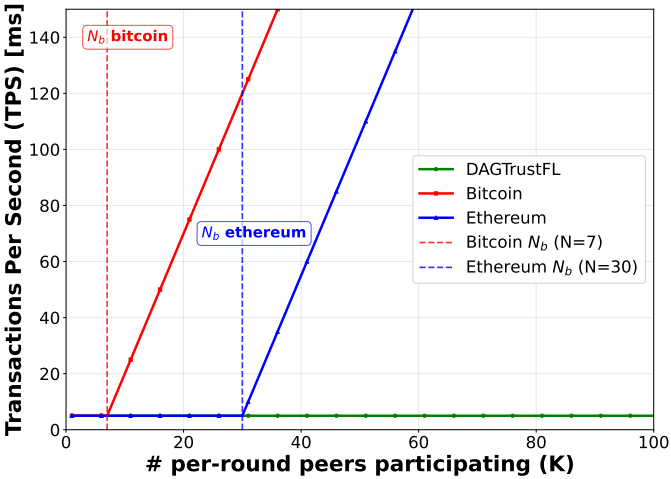
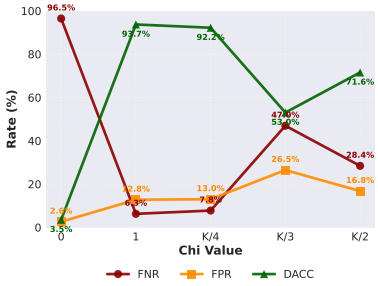
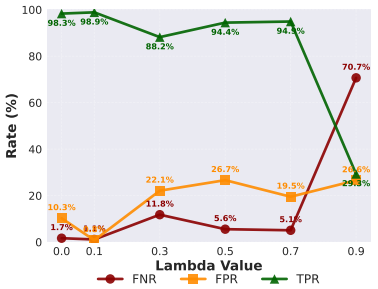


Figure 34: The TPS behavior while varying the number of FL participants in a single round (K) with different ledgers: DAG compared to blockchains (Bitcoin and Ethereum).

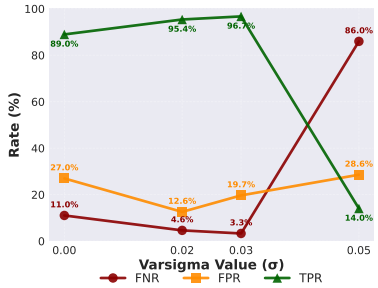
or strongly non-IID regimes with severe skew. Importantly, if the aggregator were compromised, it could bias trust computation and admission policies. Third, comparability with blockchain-based trust layers is limited by reporting practices in prior work: most papers provide only end-model accuracy on heterogeneous datasets/protocols and rarely report FPR/FNR/DACC or system cost, which prevents metric-aligned head-to-head comparisons; our discussion therefore remains qualitative at the DLT-family level. Fourth, the empirical scope is narrow: results are obtained on Fashion-MNIST with a lightweight CNN and moderate N, K; the DAG simulator is co-located with the FL orchestrator, so overheads exclude WAN propagation, asynchronous participation, churn, and bandwidth constraints typical of cross-device deployments. Fifth, several knobs expose robustness–cost trade-offs (IQR buffer δ, CW threshold Θ, past-cone depth χ, decay λ, penalty ς): larger χ improves screening of corrupted links but increases false positives and computation; CW-related parameters modulate confirmation latency and selectivity; our



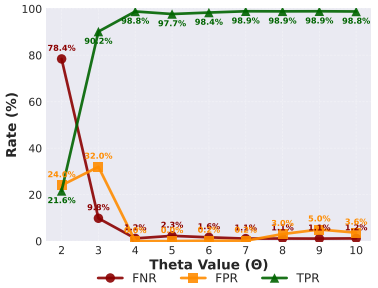
(a) FPR, FNR and DACC by varying χ value.



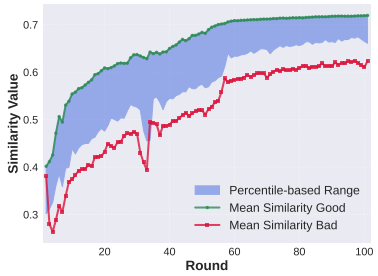
(b) FPR, FNR and DACC by varying λ value.



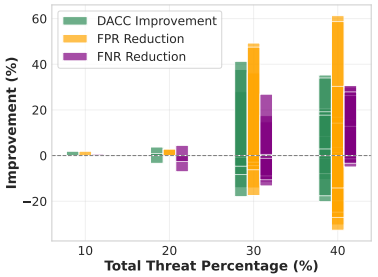
(c) FPR, FNR and DACC by varying ς value.



(d) FPR, FNR and DACC by varying θ value.



(e) Behavior of percentile-based outlier detection.



(f) Improvement in FPR, FNR, and DACC of the model with and without trust vector.

Figure 35: Comparison of experimental results across different parameter settings, to answer $H1$.

sweeps identify workable regimes but do not provide principled autotuning. Finally, we adopt low PoW difficulty for edge devices; while this eases analysis and deployment, it may underutilize the DAG's ability to encode graded endorsement strength and leaves the exact difficulty–security–latency operating point underexplored.

Looking ahead, we plan to evaluate richer similarity signals beyond cosine/Euclidean (e.g., model inversion, topological data analysis, and Benford's Law analysis [192]); broaden the adversarial suite to backdoors, adaptive attacks, and sybil strategies under both IID and strongly non-IID data; prototype a (Smart Contract)-governed or committee-based aggregation on-ledger that replaces the single SPoF aggregator; evaluate the model over an emulated WAN with asynchronous clients, dropouts, and bandwidth limits to measure end-to-end latency, validation cost, and convergence under churn; introduce adaptive pruning that preserves security-relevant subgraphs while keeping memory bounded.

3.4.9 Conclusion

This chapter has developed a cohesive security path for ICPS: GRAPH4 shows how validated security metrics can be embedded into programmable networks to sense and react with minimal overhead; Pk–IOTA addresses identity and policy enforcement by automating certificate issuance, validation, and revocation for OPC UA through in–network checks and a DAG–based ledger; DAGTrustFL extends assurance to collaborative intelligence by introducing a DAG–backed trust management mechanism that weights contributions and detects poisoning during distributed learning, leveraging explicitly defined trust metrics (e.g., update consistency, contribution quality, reputation trajectories) to quantify behavior and drive decision–making. Together, these results link measurement (metric–driven monitoring), enforcement (trustworthy identity and policy at line rate), and governance (trust over model contributions) into a single architectural trajectory for CPS.

Empirical evidence supports the practicality of this stack. GRAPH4 reduces the monitoring footprint by instrumenting only attack–relevant paths derived from attack graphs, preserving forwarding performance

while maintaining detection capability. Pk–IOTA introduces a modest overhead on the OPC UA security handshake (about 14% on the testbed) and leverages IOTA to provide immutable, tamper–evident certificate lifecycle management; Layer 1 offers simplicity and feeless operation, whereas Layer 2 adds automation via smart contracts with limited additional cost. DAGTrustFL demonstrates on Fashion–MNIST that trust–weighted aggregation improves selectivity under poisoning while preserving utility near the clean baseline; ledger overhead remains modest, the TMM cost scales linearly with participants, and transaction time remains essentially stable as the network grows, in contrast to blockchain baselines that suffer throughput degradation. These results collectively indicate that DAG–based trust and programmable enforcement can harden ICPS with limited operational cost.

Limitations remain. Attack graphs rely on complete and timely vulnerability knowledge; incompleteness can result in reduced coverage. P4–capable hardware is required to push checks into the data plane. Operating the ledger substrate entails infrastructure choices (public nodes on Layer 1 or a sidechain on Layer 2) with corresponding operational trade–offs. Finally, trust management in open, heterogeneous environments must strike a balance between robustness, efficiency, and privacy, and calls for the continued refinement of indicators and aggregation logic.

Toward standardisation, an architectural foundation emerges that balances efficiency, security, and interoperability through formally specified trust metrics. A shared metric schema—encompassing definitions, units, collection procedures, and aggregation rules—provides the backbone for comparable and reproducible assessments across deployments. Scalability is supported by DAG–based storage and retrieval of metric–derived trust signals; security and privacy improve as trust decisions rely on cross–checked indicators while raw data remain local; auditability follows from the immutability and verifiability of the ledger; adaptability is enabled by time–evolving metric scores; and automation is achievable via smart contracts for policy enactment and lifecycle control. Advancing from a promising approach to a widely adopted practice will require governance and access rules that use metric thresholds and scores to admit,

weight, or exclude participants; careful engineering of computational cost and ledger growth; reference implementations and benchmarks to foster adoption; and a common metric representation to integrate diverse FL and CPS infrastructures.

Future directions are clear: broader support across programmable data planes, secure onboarding for industrial devices (OPC UA Part 21), consolidation of smart–contract capabilities directly on Layer 1 with IOTA Rebased, and richer, context–aware trust signals that span network, middleware, and learning layers. With these steps, the path from measuring to enforcing and governing security in ICPS becomes increasingly concrete and deployable.

Chapter 4

The Human Factor

4.1 Expanding to Human-Cyber-Physical Systems (HCPS)

Industrial security in the previous chapters has been addressed from the vantage point of networks, software, and devices. Yet they do not operate in a vacuum, and real production systems are socio–technical: people design, supervise, intervene, and increasingly co–act with autonomous assets. Ignoring human roles leaves blind spots in both assurance and performance. Industry 5.0 makes this explicit by placing human centricity alongside efficiency and resilience, arguing for technologies that complement human capabilities rather than replace them. Within this vision, the Human–Cyber–Physical System (HCPS) becomes a foundational construct: a composite system where humans, cyberspace, and physical assets interact across manufacturing levels and phases to optimise lifecycle outcomes and stakeholder well–being. HCPS elevates machine and cyber intelligence by integrating human perception, cognition, and tacit knowledge, while simultaneously augmenting human capabilities through interfaces, analytics, and assistive devices [170].

HCPS is not merely CPS with operators "at the edge." It is organised around three recurring interaction patterns that span the factory: human–in–the–loop at the unit level (operators sensing, deciding, and

acting with machines), human–on–the–loop at the system level (supervisors steering cyber services, digital twins, and decision support), and human–in–the–society at the system–of–systems level (stakeholders collaborating through platforms and services across the value chain). These paradigms are captured by an architecture that weaves cognition–to–technology integration and human–to–human interaction across connection, conversion, cyber, cognition, and configuration layers. The result is a tri–space integration in which data, models, and decisions circulate between humans, cyber services, and physical processes in a disciplined manner [170].

Key enablers make HCPS practical in modern plants: operator ability augmentation (e.g., wearables, VR/AR/MR, exoskeletons) to enhance perception and reduce workload; human–robot interaction methods that ensure safety and fluid task sharing; digital twins that synchronise human and machine states for analysis, prediction, and control; and tri–space data fusion and crowdsourcing mechanisms that turn heterogeneous human/cyber/physical signals into actionable knowledge and value–added services throughout the lifecycle. These technologies shift the control surface from purely technical assets to coordinated human–machine ecosystems, motivating security models that treat people as first–class components rather than externalities.

Treating security as a property of Human–Cyber–Physical Systems (HCPS) entails analyzing infrastructure and human behavior in tandem, and optimizing both as a unified sociotechnical system. This perspective is not optional; it follows from consistent evidence that human error is implicated in a large fraction of losses and cannot be neutralized by technology alone [45]. A sociotechnical lens clarifies why HCPS security is hard. First, unlike traditional industrial safety, adversaries in cyber contexts actively try to induce mistakes through deception [260]: a convincingly spoofed vendor email that lures an operator into clicking a link, a fake maintenance ticket that elicits credentials, or an insider who learns local escalation routines and bypasses them. These are not outliers but routine pressures on everyday work, as operators are simultaneously part of the attack surface and part of the defense. Second, enterprise net-

works must remain open enough to be useful, which limits the reach of hard technical barriers and lets open channels for attackers. Third, complexity and tight coupling make small slips cascade [7]: a rushed firewall change during a shift handover, followed by an overlooked alert, can disable redundancy paths and escalate to a plant-wide outage. Even forensic reconstruction becomes difficult when multi-protocol, partially encrypted traffic obscures root causes. Finally, the systems are opaque and fast: a single miskeyed command or misapplied script can do millisecond-scale damage before anyone can intervene.

Human fallibility also follows recognizable patterns [45]. Execution slips and lapses, such as doing the wrong thing by accident or omission, tend to be noticed more often than planning mistakes, like applying the wrong rule or acting on bad assumptions, which are detected less than half the time across industrial settings. In IT-enabled environments, those undetected planning errors are the ones that often hurt most: accepting a "standard" operating procedure copied from another line that does not fit the actual topology; disabling a "noisy" security control to restore throughput under time pressure; trusting a mental model of the network that no longer matches the deployed reality.

The common response, that is, more awareness training, is rarely sufficient on its own. Large organizations report extensive programs, yet breach rates do not reliably improve and can even worsen when training remains abstract, one-off, or misaligned with incentives and daily context [76]. The lesson from industrial safety is to move from episodic instruction to organizational practice; in cybersecurity, this means cultivating a security culture that rewards early reporting of small anomalies, strengthening sensemaking for ambiguous incidents where no playbook fits, and building "anti-fragile" learning loops that turn controlled failures in safe environments into durable habits. Translating this culture into day-to-day capability calls for personalized training rather than generic courses [196]: role- and system-specific pathways that reflect each team's exposure, technology stack, and recent incident patterns; micro-drills embedded in real workflows; and adaptive refreshers paced by measurable behaviors. Gamified feedback, timely acknowledgment, and structured after-action reviews

help internalize these practices without adding brittle procedure for its own sake, while competency metrics close the loop by steering what to practice next and when to retire obsolete content [76].

Grounded in this HCPS perspective, the next step is to examine how to exploit personalization to deliver more effective and efficient cybersecurity training. For this reason, understanding how human attributes shape security outcomes in practice is crucial. A large body of work has explored links between psychological traits and susceptibility to social engineering and cyber risk [77], and has even suggested tailoring security training to psychological profiles [81]. In organizational settings, however, collecting reliable psychological data is difficult: surveys are intrusive and costly, responses may be inattentive or biased [16, 199], and the literature offers no stable consensus on which traits consistently predict vulnerability [234]. A pragmatic alternative is to leverage attributes that are simpler to obtain, such as sociodemographic factors like education, role, domain background, and tenure, as objective and readily obtainable signals that still capture meaningful behavioral differences. Even here, caution is required: higher awareness does not automatically translate into secure behavior, as evidenced by the persistent gap between stated knowledge and actual practices, such as password hygiene [124].

To clarify these tensions and inform actionable, targeted interventions, the following section presents a systematic literature review that separates awareness, attitude, behavior, and training. The review addresses the research question: to what extent do sociodemographics influence users' awareness, attitude, behavior, and training regarding cybersecurity and social engineering attacks? By mapping findings, contradictions, and theoretical lenses across studies, the analysis aims to surface where sociodemographic signals can reliably guide customizable training and where complementary approaches are needed. This prepares the ground for a scalable, human–centered security program aligned with the HCPS and Industry 5.0 principles.

4.2 The influence of sociodemographic factors

4.2.1 Background

Several studies have examined the effectiveness of training programs in increasing cybersecurity awareness, comparing tailored approaches to generic training. Research has shown that a one-size-fits-all solution is not a good choice, as each individual possesses distinct sociodemographic characteristics that influence their perception of security threats [199]. Among the various approaches explored in the literature, some studies have focused on gamification to enhance learning, differentiating it from traditional training techniques. Gamified training programs leverage interactive and engaging elements to improve user participation and retention of security concepts. In their work, Flores et al.[76] demonstrate better training outcomes with gamification for young people; this result provides additional evidence of how age, a sociodemographic factor, can influence cybersecurity training programs for improved outcomes. Baltutti et al.[21] analyzed cybersecurity behaviors among Western European knowledge workers, identifying user archetypes through cluster analysis based on traits like diligence and trust. They found that these types align with specific socio-demographic factors, emphasizing that tailored interventions can enhance the effectiveness and efficiency of organizational training strategies. Sociodemographic data refers to population characteristics that could be linked to specific behaviors, perceptions, and decision-making activities in various domains, including cybersecurity. They include variables such as age, educational level, culture, gender, work experience, and work sector. Wei et al.[295] highlights the relevance of some of the sociodemographic factors mentioned above in shaping secure behaviors, emphasizing the need for future research to further explore these aspects in conjunction with social theories. Such theories help explain subtle differences, address research gaps, and critically challenge assumptions about sociodemographics and security. Indeed, in social sciences, these factors are studied to understand how different groups of individuals respond to specific situations: numerous works have analyzed sociodemographic factors and highlighted their significance in evaluating security behaviors [81]. Ko-

vavcevic et al.[141] shows that sociodemographic factors, such as gender, educational background, and type of school, significantly influence individuals' cybersecurity knowledge and perception, which in turn affect their cybersecurity behaviors. Building on that, the hypothesis of this work is that sociodemographic factors influence cybersecurity behaviors and users' susceptibility to threats, particularly social engineering attacks. Social engineering exploits psychological and cognitive biases to manipulate individuals into disclosing sensitive information or performing actions compromising security. Since these biases can be shaped by educational, professional, and cultural experiences, individuals with different sociodemographic profiles may exhibit distinct risk perceptions and defensive behaviors [134]. Several social theories offer insight into how sociodemographic characteristics influence responses to cybersecurity threats. For instance, Technology Threat Avoidance Theory (TTAT) and Protection Motivation Theory (PMT) emphasize how perceived threat severity and self-efficacy shape protective behaviors [284]. Deterrence Theory (DT) focuses on the role of sanctions in discouraging risky actions, while the Theory of Planned Behavior (TPB) highlights how attitudes, norms, and perceived control drive intention and behavior [124]. Life-span Developmental Theory (LDT) and Socioemotional Selectivity Theory (SST) examine how age-related shifts in priorities and social context affect risk perception and cybersecurity awareness [162]. Theory of Mind (ToM) adds a cognitive dimension, linking interpretative abilities to social experience, education, and age [301]. Hofstede's cultural dimensions [120] offer a comparative lens for examining societal differences, while the Lifestyle and Routine Activities Theory (LRAT) investigates gendered patterns in cybersecurity behavior. Finally, the Transactional Theory of Stress and Coping (TTSC) associates age-related emotional responses with cybersecurity behavior. These theories help explain the complex interplay between sociodemographic factors and variations in cybersecurity awareness, attitudes, and behaviors.

4.2.2 Methodology

Data Collection and Search Strategy

This study was designed to align with the specific focus outlined in the introduction: examining the role of sociodemographic factors in cybersecurity behaviors and assessing the potential of tailored training programs. Peer-reviewed articles were collected from established publishers, including IEEE, Elsevier, Springer, ACM, and the database indexed by Scopus, using the following search query: ("socio-demographic" OR "human factors" OR "social factors") AND ("cybersecurity" OR "information security" OR "social engineering") AND ("training" OR "awareness" OR "education"). The first two components of the query were derived from RQ1 and RQ2, whereas the third component was derived from RQ3. Only articles written in English and published after 2008 were considered. This process resulted in an initial pool of 621 articles.

Inclusion and Exclusion Criteria

Duplicates, non-peer-reviewed papers, and studies that did not address sociodemographic influences were excluded. Screening was conducted using Rayyan [1] in line with Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework. PRISMA provides a structured, evidence-based approach that ensures key methodological steps are clearly documented and reported, facilitating critical appraisal, replication, and comparison across studies [212]. This approach is implemented through a two-step process: an abstract review followed by a full-text analysis. Inclusion criteria required studies to satisfy one of these three requirements:

- The study explores the link between sociodemographic factors and cybersecurity behavior.
- The study explains this link through theoretical frameworks.
- The study evaluates/proposes tailored training programs.

[1] https://www.rayyan.ai/

Studies focusing solely on psychological factors were excluded. The snowballing technique was applied, examining references of selected papers to capture additional relevant studies. Ultimately, 68 articles met the criteria and were included in the analysis. Given the heterogeneity of their research questions and methodologies, a meta-analysis was not feasible. Instead, A qualitative and quantitative synthesis was conducted to provide a comprehensive view aligned with the exploratory aim of this study. The following section presents the results of this synthesis.

4.2.3 Results

Most of the analyzed studies rely heavily on subjective measures, with surveys being the most commonly used method, accounting for 59% of the total. In contrast, only 4.5% of the studies propose frameworks or models to explore the relationship between sociodemographic factors and cybersecurity, reflecting a limited emphasis on theoretical development in this area. Additionally, 6% of the methods are pilot studies that focus on specific situations or populations, offering valuable insights into how cybersecurity behaviors occur in particular contexts but giving context-dependent findings that are not generalizable. As shown in Figure 36, research has addressed various sociodemographic factors over time: to fulfill the objective of this work, the focus was on the main ones.

Age and gender have been consistently explored in the earliest publications and remain prominent topics of discussion. Since 2015, there has been a noticeable increase in attention to education level, cultural factors, and job-related variables, signaling a broader interest in how professional and cultural dimensions influence cybersecurity behaviors. Most studies confirm the influence of sociodemographic factors on cybersecurity awareness, behavior, and attitude, as shown in Table 8. Factors such as age and gender emerge as particularly impactful, with most articles reporting significant differences across all three dimensions. However, while most studies examining factors like culture, education level, and professional sector also found evidence of differences, the total number of articles investigating these characteristics remains relatively small, especially regarding dimensions like attitude. Regarding the other sociodemographic factors,

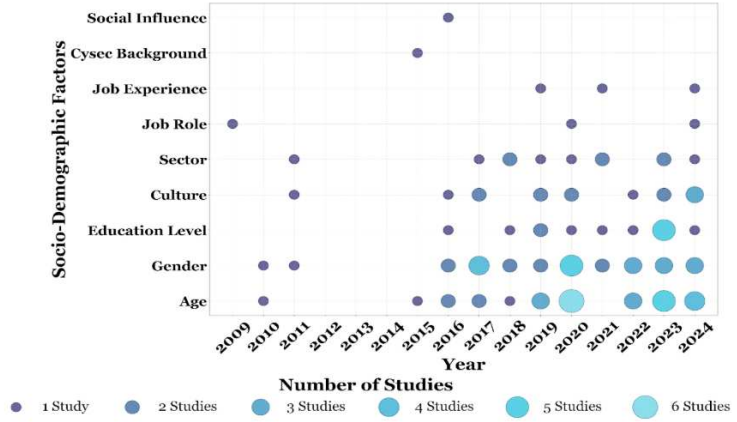


Figure 36: The distribution of sociodemographic factors studied over time.

there were not enough quantitative studies to compare the percentage of influence.

Regarding the geographical distribution of the studies, the SLR reveals that most studies have been conducted in Europe and in the US, with relatively fewer investigations emerging from Africa, Oceania, the Middle East (ME), and South/Central America, as shown in Figure 37b. The overrepresentation of certain regions may be due to more readily available sociodemographic data and a stronger emphasis on cybersecurity research, while underrepresented regions may face challenges related to funding, technological infrastructure, and differing academic paradigms. This geographic concentration is mirrored in the level of analysis employed in these studies, as the majority focus predominantly on individual-level variables, while relatively few investigate both individual and state-level factors or broader cultural dimensions (Figure 37a). However, some work does not specify the geographical area or the type of factor. Surprisingly, very few articles address the topic of cybersecurity in connection with the use of AI, and none of the ones found discuss it in relation to sociodemographic factors. The degree to which sociodemographic factors exert influence differs depending on the specific variable and theoretical frame-

Table 8: Significance of sociodemographic effects across awareness, behavior, and attitude, with darker shading indicating higher percentages.

Factor	Outcome	Awareness		Behavior		Attitude	
		n	%	n	%	n	%
Age	Significant	4	80%	10	83.4%	3	75%
	Non-Significant	1	20%	2	16.6%	1	25%
Gender	Significant	4	100%	9	75%	2	100%
	Non-Significant	0	0%	3	25%	0	0%
Culture	Significant	2	100%	3	100%	0	—
	Non-Significant	0	0%	0	0%	0	—
Education Level	Significant	2	66.6%	3	60%	1	100%
	Non-Significant	1	33.3%	2	40%	0	0%
Sector	Significant	2	100%	3	60%	1	100%
	Non-Significant	0	0%	2	40%	0	0%

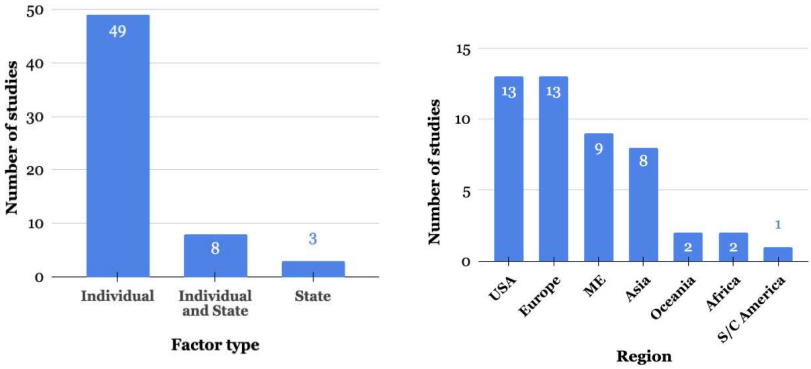
— indicates not reported / not applicable.

work; nevertheless, their impact on awareness and behavior is evident. This suggests that sociodemographic characteristics could be effectively leveraged to design tailored training interventions to maximize their effectiveness.

4.2.4 Discussion

Age

A common thread across studies is that younger individuals, especially those in the 18–25 age range, often exhibit higher vulnerability. For example, Salamah et al.[241] and Jeong et al.[134] report that younger employees and users display higher risk factors and lower levels of awareness. Ricci et al.[235] highlights that teenagers struggle to distinguish genuine content from phishing attempts, pointing to an overall deficit in awareness. Younger users are more prone to engage in risky behavior and are more likely to adopt proactive cybersecurity measures [243], albeit relying on automated solutions that may engender overconfidence. The TPB offers one lens for interpreting these differences by positing that attitudes,



(a) (b)

Figure 37: (a) Distribution of the eligible studies over categories of factors (a) and over different countries (b).

subjective norms, and perceived behavioral control shape an individual's intention to engage in specific behaviors. Alanazi et al.[8] have applied this framework to cybersecurity, showing that younger adults, immersed in digital environments and influenced by contemporary training and peer interactions, tend to form more favorable attitudes toward adopting new security measures. Their perceptions of control over digital tasks are often higher due to greater digital literacy, which bolsters their willingness to act; however, overconfidence and optimism bias can sometimes undermine effective decision-making. In contrast, older adults benefit from accumulated experience and tend to develop more cautious cyber hygiene attitudes, as noted by Giriraj et al.[92]. However, they may be particularly susceptible to phishing, as evidenced also by Salamah et al.[241]. LDT and SST help explain this paradox. LDT posits that the relevance of specific life domains evolves with age: older adults shift their priorities toward personally and emotionally significant issues. SST further suggests that individuals prioritize emotionally meaningful goals over purely informational ones because they perceive their future time as limited. As a result, older adults may be more receptive to phishing strategies

that employ emotional cues, such as reciprocation or liking, because these tactics align with their desire for social connection and emotional satisfaction. Morrison et al.[193] adds further nuance, highlighting how older adults perceive and engage with cybersecurity behaviors. When asked to evaluate and rank various protective actions, older participants expressed reluctance to engage for three key reasons: they did not want to, felt unable to, or did not see the need. Many lacked self-efficacy, often feeling overwhelmed or fearful, particularly in managing passwords. This anxiety aligns with the Transactional Theory of Stress and Coping (TTSC), where high stress leads to emotion-focused coping, like avoidance. The study underscores that cybersecurity is often a deeply emotional issue for older adults, marked by anxiety and fear of failure, yet also a domain where self-efficacy can be built with appropriate support. Additionally, research by Lin et al.[162] suggests that while cybersecurity training can reduce vulnerability among younger users, older users' ingrained trust perceptions and reduced sensitivity to deceptive cues limit the effectiveness of the training. ToM provides another perspective on the problem: since cognitive development is shaped by brain maturation, parenting, social relations, and education, differences across age groups can impact one's ability to accurately interpret intentions and detect deception. Younger individuals, who are still honing these abilities, may be less adept at discerning subtle cues of untrustworthiness, whereas older adults may rely on heuristics that make them more vulnerable to emotionally charged messages [301]. Even if industry practices often focus on older individuals, younger adults are more vulnerable to cybersecurity threats due to inexperience. However, they demonstrate higher prosecurity intentions and protective actions, defining some common vulnerabilities and strengths of specific groups. Older adults, while more cautious due to accumulated experience, remain highly susceptible to emotionally driven phishing tactics and often experience emotional barriers that hinder active engagement in protective behaviors. A targeted education could leverage the latter trait to create prosecurity attitudes, avoiding the former trait from leading to antisecurity behavior.

Gender

Another sociodemographic factor frequently examined in the studies is gender, as the relationship with cybersecurity can vary greatly depending on knowledge, behavior, or awareness. Focusing on knowledge, McGill et al.[182] and Anwar et al.[14] analyze this relationship through the PMT to investigate the key factors influencing the intention to protect ourselves; these key factors are then split into several constructs. McGill et al.[182] found that females have lower overall levels of information security behavior than males. Still, it has also been shown that women would not have a lower level of security self-efficacy and response efficacy, two of the constructs mentioned above, than men. This would allow the claim that the gender differences in security behavior do not appear to arise from them. Anwar et al.[14] further elaborates that while some variability exists between men and women, the overall relationship among the variables remains consistent, suggesting that the same theoretical or predictive model could be applied to both genders. Some studies argue that gender influences cybersecurity awareness, knowledge, and behavior more favorably for men, but an equally clear counterpart asserting the same for women was not found. Among the few, Sari et al.[243] note that women exhibit more secure behavior than men, but their different perception of technology influences their behavior, making them more vulnerable to such attacks [92]. In some cases, women have shown better knowledge than men, as evidenced by McCormac et al.[181] concerning the Information Security Awareness (ISA) score. Nevertheless, the same study highlights that women were more susceptible to phishing attacks via email compared to men. However, there are other opinions on the subject: Lee et al.[148] used the LRAT, hypothesizing that women increase their target attractiveness to cybercrime due, among other things, to their lower use of online platforms. The results refuted this hypothesis, showing that women are more frequently targeted in cyber attacks simply due to attackers' preferences. These conclusions provide an interesting point of reflection, suggesting that LRAT may not be an adequate theory to explain the correlation between gender-based lifestyle and cybersecu-

rity preparedness. Zwilling et al.[330] further elaborates on the TPB and asserts that gender would not be significantly associated with protective behavior, stating that there are no distinctions between men and women in protective activities. The considerations mentioned so far do not take into account another critical element: the impact of stereotypes. Wei et al.[296] specifies that gender stereotypes contribute to distinguishing between two groups: the "perceivers," who hold the stereotypes, and the "experiencers", i.e., the targets. The consequences affect both groups; however, focusing on the experiencers, the authors demonstrate that gender stereotypes are deeply rooted in the participants' perceptions, contribute to various individual outcomes, and generally reduce performance due to stereotype threat. In a significant proportion, the outcome was that both men and women felt that women's likelihood and/or ability to protect their privacy or security was lower than that of men. Studies show mixed results: while men often exhibit higher security awareness, women demonstrate secure behaviors. Still, they are more susceptible to specific cyber threats due to perception biases and stereotypes rather than actual knowledge gaps. Many results on gender were contradictory; to better understand these dynamics, future work could examine how different types of cybersecurity training programs interact with gender to shape security behaviors.

Culture

The relationship between cultural factors and cybersecurity awareness is a recurring theme in multiple studies, highlighting how these characteristics influence security behaviors at both the individual and organizational levels. To systematically understand cultural differences, Hofstede's cultural dimensions [120] provide a valuable framework for understanding variations in cybersecurity behaviors across different societies. One key dimension, Uncertainty Avoidance (UAI), refers to the extent to which members of a society feel uncomfortable with ambiguity and risk. This is particularly relevant in Zwilling et al.[330], where Turkish respondents, considered part of a high-UAI society, viewed cybersecurity as highly risky and threatening, which may explain their greater engagement in protective

behaviors. Conversely, Israeli respondents, reflecting a lower UAI culture, exhibited lower levels of personal cyber threat avoidance, relying instead on institutional cybersecurity infrastructures. Language emerges as another key cultural determinant affecting security awareness. Zwilling et al.[330] found that Turkish respondents, by answering an awareness questionnaire in their native language, exhibited different perceptions than their counterparts who responded in English. These findings align with the Individualism vs. Collectivism framework, as collectivist cultures tend to rely more on shared social knowledge and localized language rather than standardized global frameworks. Chen et al.[49] provides key insights into cultural differences between Chinese and American users in cybersecurity behaviors, integrating PMT and TTAT to explain coping strategies. Building on this, they further emphasize that self-efficacy and response efficacy must be addressed differently across cultures to improve cybersecurity outcomes. In individualistic cultures like the U.S., users focus on personal efficacy and proactive security actions, while in collectivist and high Power Distance societies like China, individuals rely more on government and community protection. This reliance reduces the perceived need for individual security measures, a pattern also seen in avoidance behaviors. TTAT suggests that while Americans trust protective tools, Chinese users, influenced by modesty and communal values, are less inclined to take independent action. Similarly, PMT highlights differences in self-efficacy: Westerners exhibit optimism tied to personal success, whereas Chinese users have a more realistic but stronger predictive self-efficacy, making them more likely to comply with authority-endorsed security solutions rather than question them. The findings suggest that UAI societies tend to engage more in protective actions, collectivist cultures rely on communal knowledge and institutional support, while individualistic cultures emphasize personal responsibility in security practices. While developed nations focus on technological solutions, research highlights the need to prioritize cybersecurity education and social awareness.

Education Level

Research on education level reveals both common patterns and notable differences. A common theme among the studies is that educational background plays a positive and influential role in shaping cybersecurity practices. For instance, Salamah et al.[241] demonstrates that employees with higher educational qualifications are generally less vulnerable because they better understand phishing, adhere more closely to password security policies, and can identify illegitimate websites, thus exhibiting better cognitive skills. This finding is echoed by Jeong et al.[134], who report that higher education levels are associated with less risky actions and greater compliance with cybersecurity-related activities. In contrast, sari et al.[243] compared different educational groups among users, finding that diploma holders exhibit better cybersecurity behavior and a higher propensity for prosecurity actions. In contrast, postgraduate users are less inclined to engage in such practices, aside from specific behaviors, such as opening email attachments from trusted senders. However, high school users appear to be the most likely to engage in anti-security behaviors, as also highlighted by the Age results. This stratification suggests that the relationship between education and cybersecurity is not strictly linear; instead, the type and context of the educational experience may influence behavior in distinct ways. Zwilling et al.[330] employed the TPB to examine how education influences cybersecurity awareness. Individuals who perceive their education as positively impacting their cybersecurity awareness tend to report higher overall awareness, further reinforcing the connection between educational experiences and the ability to recognize cybersecurity hazards, as well as the importance of self-efficacy. Vrhovec et al.[284] utilizes PMT to investigate trust perceptions and the effectiveness of awareness campaigns between students and employees. Their findings suggest that while employees tend to be influenced by messages that underscore the effectiveness of state authorities, students may respond more positively to cybersecurity campaigns. Hong et al.[123] provides additional insights by comparing working graduates and university students, finding that non-final-year and final-year students score higher in cybersecurity

attitudes and behaviors than full-time working graduates. Their results suggest that prolonged exposure to a work environment where security is not proactively pursued might seriously deteriorate ISA, highlighting the moderating effect of the work environment. Hong et al.[124] employed TPB to understand how attitudes, subjective norms, and perceived behavioral control shape cybersecurity intentions, while PMT emphasized the appraisal of threats and evaluation of coping responses, and DT provided insight into the decision-making processes under risk. Higher education, associated with enhanced cognitive skills and critical thinking, should foster more effective coping appraisals and improve an individual's capacity to assess the severity of cybersecurity threats and the cost-benefit analysis. They confirmed through survey data that situational support promotes robust cybersecurity behavioral habits via serial mediating effects involving self-efficacy and behavioral comprehensiveness, as well as response efficacy and behavioral comprehensiveness. However, their study found that first-year students reported significantly higher perceived situational support than their more advanced counterparts. These findings indicate that higher education is generally associated with better cybersecurity awareness and prudent behavior; however, the relationship is nuanced and context-dependent. Early stages of this educational path present a crucial opportunity to instill effective cybersecurity behaviors, as age also influences how individuals acquire cybersecurity knowledge [232].

Job and Study Sector

Salamah et al.[241] explores the professional domain by comparing various sectors and cybersecurity behaviors. Their findings indicate that IT employees demonstrate higher cybersecurity awareness; for example, they consistently classify phishing emails as security incidents and adhere strictly to antivirus updates. In contrast, employees in the education, business, and finance sectors tend to be less vigilant, often neglecting crucial practices such as updating antivirus software or verifying the authenticity of links. This suggests that the working sector can significantly affect how cybersecurity measures are understood and implemented. Gallo et al.[81], comparing STEM and non-STEM employees, reveals that a

substantially higher percentage of STEM employees actively report suspicious emails to their security departments, implying that technical training or exposure to STEM-related disciplines fosters a more significant commitment to proactive cybersecurity practices. This may also reflect greater exposure to cybersecurity training among individuals in STEM fields. Tanriverdi et al.[265] provides another layer of insight by demonstrating a relationship between an employee's profession and their knowledge about external IT security threats. Interestingly, they did not find a similar association regarding insider threats or in performing prosecurity behaviors, highlighting that the type of threat considered may influence the perceived impact of one's professional background. Mittal et al.[190] contributes to the discussion by focusing on students from different academic disciplines. They found that students in arts and commerce tend to access potentially harmful websites more frequently, which points to a relative lack of cybersecurity awareness or a different risk tolerance in non-technical fields. Their work notes that even within advanced levels of study, such as among PhD students, training is still necessary to underscore the risks of entering personal information on insecure websites. Similarly, Huraj et al.[126] compares cybersecurity practices between computer science and media studies students, finding that the former tend to adopt stronger technical safeguards, whereas the latter, with a focus on qualitative analysis and critical reflection, are less likely to install unverified software, reflecting distinct risk perceptions. Hong et al.[124] uses DT, PMT, and TPB to explain the correlation between self-efficacy and its positive impact on behavioral comprehensiveness. They found that students majoring in Science and Engineering had significantly higher self-efficacy than others, thus following better prosecurity patterns. Findings show that IT and STEM employees demonstrate higher awareness and proactive security practices, while those in non-technical fields, such as education, business, and media studies, exhibit lower vigilance and differing risk perceptions. Professional background shapes cybersecurity practices: training should focus on enhancing awareness and promoting proactive practices in non-technical fields, while building on the existing security knowledge and practices of STEM professionals.

4.2.5 Concluding remarks

Cybersecurity awareness and behavior are shaped by sociodemographic factor, including age, education, professional sector, culture, and gender, that influence how individuals perceive and manage cyber risk. These attributes are best treated as a non-discriminatory starting point for personalised training: when combined with role, context, and individual learning needs, they enable content that fits participants more closely and improves outcomes. Despite this promise, AI-driven methodologies remain underutilized in behavioral cybersecurity, partly due to the novelty of the techniques and the scarcity of high-quality datasets. As empirical evidence accumulates and data availability improves, robust AI models can more accurately capture sociodemographic effects and help scale tailored training across diverse HCPS settings.

Personalization alone, however, is insufficient against a threat surface increasingly shaped by AI-mediated manipulation. Deception undermines trust and integrity in digital communication, now operating on a large scale across email, social media, messaging platforms, virtual environments, and AI-driven systems. A salient driver is the rise of deepfakes, synthetic video, audio, image, or text that is partially altered or entirely generated (e.g., via generative adversarial networks), capable of credibly imitating a person's face or voice and fabricating events that never occurred [82]. While such media can serve benign purposes, malicious uses, such as misinformation, disinformation, fraud, and identity manipulation, make deepfakes a particularly insidious form of computer-mediated deception [109, 110, 113].

Compared with face-to-face (F2F) settings, computer-mediated deception benefits from anonymity, asynchronicity, and the absence or reduction of nonverbal cues, which complicate detection and lower the cost of perpetration. Attackers exploit these affordances through identity spoofing, spear-phishing, bot-amplified narratives, and synthetic voice/video prompts that leverage authority, urgency, and familiarity biases [40, 143, 106]. The societal and organizational consequences include erosion of public trust, distorted decision processes, and substantial financial losses [24,

252].

Given these dynamics, effective HCPS security must pair personalized, data-informed training with interdisciplinary detection approaches that integrate linguistic, behavioral, physiological, and contextual indicators, prioritizing those that remain reliable under adversarial pressure. The next chapter develops that perspective and delineates the methodological and ethical considerations required to identify and resist manipulation at scale.

4.3 Deception in the AI era

4.3.1 Deepfake Challenges

Research on deception has evolved substantially. Early work examined face-to-face (F2F) interactions and focused on physiological cues such as microexpressions and gaze patterns [189, 278]. More recent studies investigate linguistic and behavioral markers in computer-mediated communication (CMC), where deception appears through textual ambiguity, response latency, and stylistic inconsistencies ([113, 326]). Existing detection approaches include automated systems [189], reality monitoring [79], veracity assessment [252], physiological measures such as finger tapping tests and rigidity of body movement (citebastick2021would, proudfoot2016man, twyman2014rigidity, twyman2014autonomous, twyman2015robustness, textual and linguistic analyses [161, 239, 253, 326, 327], and machine learning classifiers [68, 113, 117, 118, 143, 252, 282]. These methods were not designed or developed to detect deepfake-driven deception. Social engineering and phishing exploit human psychological weaknesses rather than technical flaws [40]. Deepfakes intensify these attacks by increasing perceived authenticity and emotional impact, making them harder to detect and shifting the focus onto cognitive vulnerabilities. Attackers can use AI-generated voice clones of executives, colleagues, or family members to induce transfers, credential sharing, or disclosure of sensitive data. Synthetic audio and video also serve as persuasive delivery vectors, for example in spear-phishing emails. These tactics leverage

authority and familiarity biases to lower scrutiny and create cues such as urgency or confidentiality, which raise compliance and inhibit verification. Current challenges include orchestrated disinformation campaigns and AI-generated content [57, 108, 161].

4.3.2 Research Gaps

Despite substantial progress, several gaps remain. Deception research is fragmented across computer science and engineering, information systems, psychology, and criminology, which hampers a cohesive interdisciplinary view of modern deception. AI-powered tools, such as machine learning models that analyze language patterns, show promise but often falter in real-world settings because of cultural biases, the diversity of manipulation strategies targeting psychological vulnerabilities, and advances in adversarial AI. Each scenario is context-dependent, and a universal detector is difficult to achieve. Ethical concerns, including privacy and the risk of false accusations, are also underexplored [86]. These issues motivate a systematic review that traces the evolution of deception research and extracts actionable insights for mitigation. Therefore, this review asks: What is the trajectory and current state of deception research in information systems, and how can it inform our understanding of deepfakes? This question centers on identifying and grounding deceptive cues in physical and digital contexts and on the methods used in prior studies. It is divided into two sub-questions. First, cue categories across physical and online settings were analyzed (i.e., physiological, linguistic, behavioral, and multiple signals) as markers of deception. Second, explanatory frameworks that link these cues to deceptive behavior across contexts were examined. Drawing on 57 peer-reviewed studies in information systems conferences and journals from 2004 to 2024, this review pursues three objectives: to synthesize evidence-based indicators of deceptive cues across contexts, to evaluate the effectiveness of detection methodologies amid emerging technologies, and to propose a roadmap for interdisciplinary research suited to an increasingly complex media landscape.

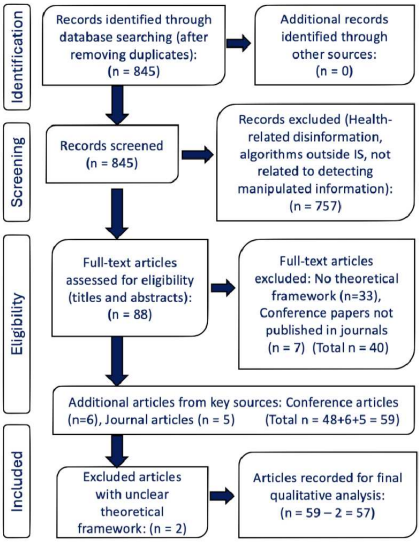


Figure 38: The PRISMA process applied in this study.

4.3.3 Methodology

To ensure transparency, consistency, and completeness in the systematic literature review, the PRISMA framework was adopted, as described in Section 4.2.2. The process is illustrated in Fig. 38.

Identification and Search Strategy

The search was initiated using Google Scholar with terms such as ”deepfake,” ”disinformation,” ”misinformation,” ”fake news,” ”online deception,” and ”computer-mediated deception.” These terms were selected to capture both traditional forms of digital deception and emerging AI-mediated manipulations. The goal was to review 20 years of peer-reviewed publications (2004–2024) across information systems and behavioral sciences to inform future research on computer-mediated deception, with particular attention to deepfakes. However, this approach did not capture several key information systems journals. To address this, the search

was expanded to include the Web of Science and Wiley Online Library; however, both posed limitations for journal-specific queries. EBSCO was selected as the primary database, supplementing it with Google Scholar and ScienceDirect. Ultimately, articles from fifteen leading, authoritative, and well-regarded journals in the field of information systems were incorporated into the final database.

Screening and Selection Process

After removing duplicates, 845 unique articles were imported into Covidence, a systematic review management tool. Title and abstracts were independently screened by two researchers, with key information recorded in a structured matrix covering source, discipline, theory, methods, data, findings, and contributions. Papers were evaluated for peer-review status, theoretical articulation, and relevance, and disagreements were resolved through discussion. Only studies explicitly grounded in theory related to deception and demonstrating methodological transparency were retained. A second screening ensured consistency, followed by full-text review by six researchers, with collective resolution of any disputes. Metadata, including authorship, publication year, theoretical framework, and research method, was extracted. Ultimately, 757 articles were excluded because they fell outside the scope of information systems, focused solely on health-related contributions (e.g., COVID-19), emphasizing algorithms (e.g., bot or phishing detection) without a deception component, or were otherwise irrelevant. This process yielded 88 eligible articles.

Eligibility Criteria

From this set, 33 articles lacking theoretical grounding and 7 conference papers not published in journals were removed, as these works typically do not form the basis for continued theoretical development. This left 48 journal articles with substantive theoretical foundations. To ensure comprehensive coverage, 11 key studies were added (six conference papers and five journal articles), recognized as essential contributions to the literature, as they represented a continuation of prior scholarly work. In

the final screening round, two articles were excluded because they lacked a clear theoretical foundation. The resulting dataset comprised 57 peer-reviewed articles, selected by a rigorous and consistent process.

Inclusion and Exclusion Criteria

The review was limited to peer-reviewed, English-language journal articles published between 2004 and 2024. This timeframe was chosen to capture the evolution of deceptive technology from early forms of digital fraud to advanced synthetic media, and to examine how detection methodologies have adapted over time. Only high-quality, peer-reviewed literature was retained to ensure rigor and consistency. Studies were excluded if they primarily focused on health-related disinformation (e.g., COVID-19), phishing, malware, social engineering, financial fraud, online fundraising, fake reviews, cyberbullying, or social media addiction. For example, a large amount of papers published between 2020 and 2023 were related to the COVID-19 pandemic; these studies addressed misinformation and disinformation about COVID-19 vaccinations, including public health and pandemic-related narratives without a clear focus on deception behaviors through algorithmic content manipulation, transactional or operational attacks—categorized under criminology, fraud prevention, or cybersecurity engineering—rather than interaction tactics and strategies with humans. These were excluded because they either fell outside the scope of generalizability, were overly technique-driven without theoretical grounding, or lacked a clear human or deception component.

4.3.4 Deception across Physical and Digital Contexts

Deceptive behavior occurs in both physical and digital contexts, but cues and detection methods differ. In physical settings, deception research often leverages automated detection systems to analyze behavioral and hybrid cues, focusing on involuntary "tells" that indicate deception [247]. These systems, commonly used in law enforcement and border security, detect subtle leakage behaviors such as micro-expression, facial movements, hand gestures, and head velocity [189, 278]. Credibility assessment

in interview settings examines kinesics cues (e.g., body movements, gestures, posture, eye contact) and behavioral patterns such as multitasking or error rates to evaluate credibility [278, 279, 277, 280]. Eye-tracking studies have identified pupil dilation and eye-gaze fixation patterns as additional indicators of deception [226]. These systems often rely on automated tools to detect inconsistencies in nonverbal behavior as cues for deception. In contrast, digital environments lack access to physiological cues, so deception has relied primarily on linguistic and contextual analysis. Automated systems analyze patterns in online communication to identify deception through cues such as evasiveness or indirect responding [228, 327, 326]. In CMC, deception is defined as the intentional use of digital channels (e.g., emails, messaging apps, or social media) to mislead. Sender credibility is key, correlating with deception success and false alarm rates [86, 88]. Cultural and contextual factors further shape online deception. Communication norms vary across cultures: high-context cultures rely on implicit forms of deception, whereas low-context cultures use more explicit cues [80, 85, 89, 87]. Deceivers often communicate less directly, using ambiguous or opaque messages to obscure intent [113, 326]. Media richness also matters: richer media, such as video, provide more cues and reduce opportunities for deception. Leaner media, such as text, allow for greater message control [110]. Deception is also prevalent in e-commerce and social media, including misrepresented products and fraudulent reviews. Xiao et al.[298] theorizes how unethical practices exploit information asymmetry to mislead buyers, while Banerjee et al.[24] and Kumar et al.[143] emphasize the role of linguistic cues in deceptive online reviews. Online deception encompasses a broader range of tactics, such as fake profiles [246, 282], gender deception [115], and deceptive sources [25, 106, 143]. Phishing emails, which exploit appeals to urgency and authority [40], and deceptive behaviors shaped by social structure and network dynamics [213] further illustrate the breadth of online deception. Collectively, these studies demonstrate how platform affordances facilitate deception, making it more difficult to detect deceptive practices. A key distinction exists between interpersonal deception and group deception. In interpersonal contexts, deceivers often use linguistic strategies such as ambiguous

wordings, unusual delays, or reduced use of first-person pronouns to conceal intent [109, 110, 113, 117, 118, 326]. Cultural norms and gender also influence both deceptive behavior and the ability to detect it [80, 115]. In group or organizational settings, collective awareness improves detection [87, 280], as members cross-check statements and identify inconsistencies in behavior or speech. Insider threats often involve subtle attempts to conceal activities by masking behavioral cues, which are sometimes exposed through unusual communication patterns in team contexts [108, 116, 112]. Communication modality also plays a role: liars experience a greater cognitive load in synchronous environments (e.g., live chats [256]), making deception harder to sustain. In contrast, asynchronous environments (e.g., email) allow more time for message construction, thereby complicating efforts to detect malicious messages. Below, the characteristics of interpersonal deception and deception in group communication are reviewed.

4.3.5 Physiological, Linguistic, Behavioral, and Multimodal Cues

Computer-mediated deceptive cues can be grouped into four complementary categories: physiological, behavioral, linguistic, and multimodal.

Physiological cues

In face-to-face (F2F) contexts, deception detection studies often rely on automated systems to capture subtle nonverbal "leakage" cues. These "leakage" cues are involuntary signals of deceit that are difficult to suppress. Such systems are widely deployed in security and investigative settings, including border control and law enforcement [189, 277]. While CMC typically limits access to visual or physiological signals, certain scenarios (e.g., video calls, biometric-enabled platforms) still permit the observation of physiological markers such as micro-expressions, facial muscle activity, eye-gaze fixation, pupil dilation, hand positioning, and head velocity [189, 225], as well as finger tapping speed [26]. These cues reflect involuntary physiological responses to the cognitive and emotional de-

mands of lying, making them valuable inputs for automated credibility assessment systems [277, 279].

Behavioral cues

Behavioral cues of deception often emerge through subtle shifts caused by cognitive strain or strategic pressure. In automated interview settings, unobtrusive measurements of facial and hand responses [189], along with nonstrategic oculometric patterns [225], reveal involuntary leakage when deceivers encounter novel stimuli—typically showing initial spikes in pupil dilation followed by rapid decreases due to repetition priming. Deceptive individuals also tend to fixate more frequently on neutral screen areas as a defensive tactic. In the phishing email domain, users' interaction patterns with message elements serve as behavioral signals: most participants hovered over links fewer than five times on average, whereas those who inspected URLs more frequently demonstrated significantly better discrimination between genuine and phishing emails [40].

Linguistic cues

In fully text-based environments, linguistic features become the primary evidence of deception, often reflecting a deceiver's attempts to control the narrative [147]. Deceptive messages exhibit measurable differences in quantity (shorter or more verbose), complexity (at both vocabulary and sentence levels), specificity, expressiveness, and formality [38, 228, 326]. Chat-based studies show deceivers take longer to respond, revise their replies more often, and produce shorter messages with reduced lexical diversity ([57]). Rubin and Lukoianova (2015) employed discourse structure analysis to measure the distance between truthful and deceptive centers and detect rhetorical inconsistencies as indicators of deception. Deception detection research has increasingly taken into account the interactional complexity within organizations. Language-action cues—such as pronoun usage, sentiment, and turn-taking behavior—can reveal both defensive and promotive strategies [147]. Deception in group-level studies reveals that linguistic patterns shift significantly after an insider has been com-

promised, with marked changes in inclusivity/exclusivity markers, cognitive effort, and moral conflict terms [106, 116, 111, 114]. Research on fraud detection in crowdfunding [252] and online reviews [24] reinforces the predictive value of affect, complexity, specificity, and exaggeration. Planned deception may allow for linguistic refinement, but spontaneous deception—particularly in synchronous interactions—tends to reveal more cues, such as longer pauses, reduced use of negations, and overly friendly or ingratiating language [110, 113, 117, 118]. Computational models that leverage these linguistic and interactional cues have achieved high accuracy in detecting deception across both dyadic and group contexts (Ho and Hancock, 2018; 2019).

Multimodal cues

Lewis et al.[153] emphasizes that the richness of a communication medium depends in part on its ability to convey multiple cues simultaneously, including verbal, nonverbal, and contextual cues. Richer media, such as F2F interaction, naturally provide more of these cues, whereas CMC restricts their availability. From a detection standpoint, integrating all available cues within a medium, even when limited, enhances veracity assessment by enabling cross-referencing across modalities and exposing inconsistencies that may be overlooked when focusing on a single channel. For instance, incongruence between verbal statements and behavioral patterns often serves as a stronger indicator of deception than either cue by itself.

4.3.6 Multimodal Deception Theoretical Foundation

A foundational model in the study of deception is Interpersonal Deception Theory (IDT)[37, 38]. IDT identified three main fundamental forms of interpersonal deception: equivocation (avoiding comments), concealment (omitting facts), and falsification (making false claims)[36]. Deception often occurs to gain advantages, preserve self-image, or maintain relationships with others. Identifying deception is challenging due to the truth bias, which makes trusting others a complex decision shaped by the interactive behaviors of both the sender and the receiver. Deception is

further influenced by factors such as motivation, communication skills, familiarity, and fear of being exposed. A key limitation in prior research is that deception has been most often studied in isolation, within a single context, which typically oversimplifies the phenomenon. Studying deception in isolation overlooks the complex, adaptive, and context-dependent nature of deceptive communication, thereby limiting the validity and applicability of the findings. Deception is inherently interactive, dynamic, and context-dependent by nature. 1. Deception is situational—the cues, strategies, and detection success rates vary widely depending on the social, cultural, and technological context. Studying it in only one context may yield misleading or non-generalizable conclusions. 2. Deception is interactive—it unfolds online in exchanges between sender and receiver. Focusing solely on one side (e.g., the deceiver's behavior) or on a single situation overlooks how responses, counterstrategies, and adaptations influence the outcome. 3. Deception involves variability—what works as a deceptive strategy in one setting (e.g., F2F) may fail in another (e.g., online or high-stakes). A single-context approach masks this variability. 4. Deception is multimodal: with the advent of generative AI (GenAI) technology, deception is no longer conveyed through a single channel (e.g., just words) but rather through multiple modes of communication that operate simultaneously and interactively. These include verbal content, paralinguistic cues (such as tone and hesitation), nonverbal behavior (e.g., facial expressions and gestures), physiological signals, and contextual/digital cues. Understanding deception in contemporary communication environments requires a strong theoretical grounding. Across the literature, several frameworks recur consistently, revealing a shared intellectual foundation among researchers of deception. These frameworks generally fall into four categories: cognitive theories, social theories, social psychology theories, and communication/ media theories (see Figure 39). Cognitive load and four-factor theories are frequently employed to identify behavioral and linguistic cues of deception. Social learning and social cognitive theories offer insight into deception in the workplace and group contexts, including insider threats. Media-based theories, such as media richness theory (MRT) and media synchronicity theory (MST), help explain how

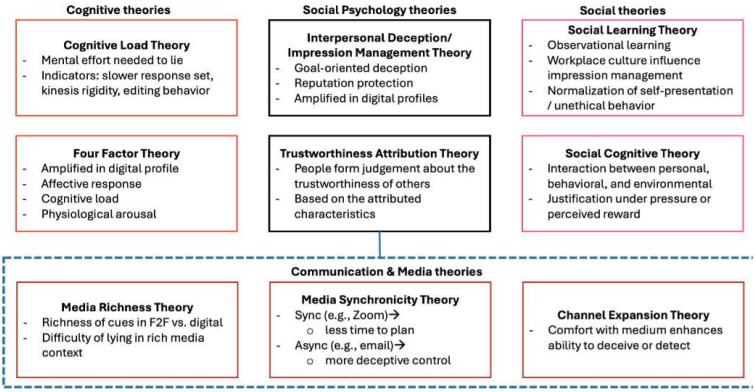


Figure 39: Theories adopted in the 57 deception research articles.

deception adapts across communication channels. What is needed is a more integrated approach—one that captures how cognition, social environments, and communication technologies interact in complex deception scenarios (Figure 39).

Cognitive theories

Cognitive theories focus on the mental processes involved in deception. Foundational among these is cognitive load theory, which posits that lying demands more cognitive effort than telling the truth [263]. Liars often fabricate details, manifest inconsistency, and suppress the truth, all of which increase the strain. This strain often manifests behaviorally—through longer pauses, reduced fluency, or rigid body language; a phenomenon known as kinesic rigidity [277]. The four-factor theory further outlines how deception reveals itself through (1) attempted behavioral control, (2) affective response (e.g., guilt or anxiety), (3) cognitive effort, and (4) physiological arousal (e.g., increased heart rate or sweating). These cues—both internal and external—are critical for identifying deception through both verbal and nonverbal behavior. In digital environments, deception may become even more prevalent and easier to execute. Online platforms enable users to curate or even reinvent their identities—whether on social

media, job boards, or dating apps. For example, someone might subtly exaggerate their qualifications on LinkedIn. Schroeder et al.[246] argue that this digital self-presentation often blurs the line between impression management and deception, raising complex questions about authenticity. Strategic deception, which involves deliberate planning, is particularly well-explained by motive-control theory. This theory suggests that individuals weigh the perceived risks and rewards of deception. If the potential benefits outweigh the costs, they may invest greater effort into crafting convincing falsehoods [39]. Over time, frequent deceivers may become more skilled, creating a feedback loop where success reinforces future deception. Disruption of this cycle—by increasing detection risks or reducing rewards—is a common strategy in deception detection research [279].

Social theories

Social theories highlight that deception is not only an individual act but can also be a learned social behavior. Social learning theory [23] suggests that individuals model their behavior on what they observe in others. In workplace settings, this may mean employees adopt deceptive practices because such behavior goes unpunished or is tacitly normalized [269]. Over time, unethical behavior can become ingrained in an organization's culture. Social cognitive theory [22] expands on this by incorporating personal beliefs, emotions, and environmental factors. Deception arises from the interplay between individual cognition and social context. A person may lie to avoid punishment, gain an advantage, or conform to perceived norms. It is this constant interaction between the personal thoughts, the observed actions, and the social cues that makes deception more than just a matter of one's own personality. This framework reveals that deception is often shaped by situational and systemic influences, not merely personal morality.

Social psychology theories

Social psychology theories emphasize the influence of social context on behavior. Impression management and self-presentation theories explain how individuals use deception to control how they are perceived—whether to appear more competent, likable, or trustworthy. Some individuals would make a strategic effort to manage the presentation of themselves or the artifacts of their identity to appear authentic. Not all deception is malicious; sometimes it is benevolent (e.g., white lies), aimed at preserving harmony or avoiding conflict. Goffman et al.[94] dramaturgical perspective reminds us that much of social life involves performance, often filtered through socially accepted norms rather than total transparency. In digital contexts, the absence of physiological cues places greater emphasis on verbal indicators. Online deceivers may use language that is more ambiguous or complicated, delay more, or refrain from using the first-person pronouns. These verbal patterns, now detectable by AI-based systems, are increasingly recognized as indicators of deception. One of the most complex theoretical frameworks in social psychology is the trustworthiness attribution theory, which posits that people form judgments about the trustworthiness of others, including individuals, organizations, or systems, based on attributed characteristics. The trustworthiness attribution theory is a psychological and communication theory that explains how and why people decide someone (or something) is trustworthy [108, 112]. In the attribution process, people observe behaviors or outcomes and attribute causes to them. In trust contexts, these attributions focus on why someone behaves in a trustworthy or untrustworthy way. Rooted in attribution theory, it posits that judgments are based on perceived ability, benevolence, and integrity [178]. In AI contexts, users form attributions based on system transparency, past performance, and inferred intent.

Communication and media theories

Communication and media theories examine how various communication modes, whether face-to-face, email, or through social media, impact both the act and detection of deception. That is, how people lie and how we

catch those lies depend heavily on where the deception happens. Media richness theory (MRT) holds that face-to-face (F2F) interaction, which offers the richest array of cues (e.g., tone, gesture, facial expression), makes deception more difficult. However, when the interaction moves online, the dynamics shift. Media synchronicity theory (MST) explains this shift by distinguishing between synchronous and asynchronous communication. That is, MST emphasizes the timing of communication. Asynchronous platforms (e.g., emails) allow deceivers time to craft and edit messages, thereby increasing the opportunities for deception. Synchronous interactions (e.g., phone calls, live chat, or video calls) limit this preparation time and often expose more behavioral leakage. Channel expansion theory (CET) adds another layer of nuance, suggesting that experience with a communication medium affects one's ability to use it effectively, including for deceptive purposes. Someone proficient with Discord or Slack, for instance, may know how to manipulate features to conceal deception, such as deleting messages, exploiting visibility controls, or selectively sharing content. This highlights how deception is not only psychological but also strategic and technical in nature. Deceivers often choose platforms that maximize their control—those that offer anonymity, editing features, or delayed responses. Meanwhile, researchers and technologists are developing new tools that analyze digital cues, such as typing speed, lexical complexity, and interaction timing, to flag potential deception. These tools aim to identify digital "tells" when traditional cues, such as eye contact, are unavailable.

4.3.7 Plural Methodologies

Researchers investigating deception utilize linguistic analysis, cognitive tests, and automated systems to detect lies through nonverbal, verbal, and physiological behavior (Figure 40). However, no single method is flawless. Each has limitations, ranging from contextual dependencies and the need for controlled environments to susceptibility to bias. Therefore, a multidisciplinary approach that integrates cognitive, behavioral, and automated techniques offers greater promise for accuracy.

Reality monitoring (RM) is a cognitive framework used to differentiate

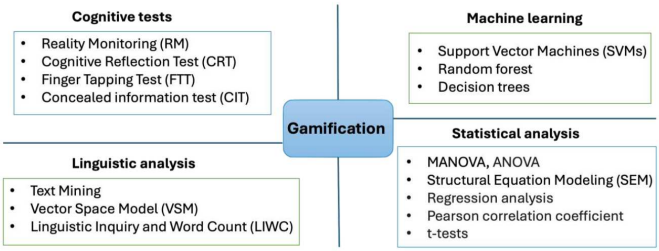


Figure 40: Methods adopted in the 57 deception research articles.

between truthful and fabricated statements based on memory characteristics. Truthful accounts generally include perceptual, contextual, and emotional details, whereas deceptive narratives often rely on reasoning and speculation. Fuller et al.[79] demonstrated that automated linguistic tools based on RM principles can detect deception by identifying these differences: truthful narratives tend to be more vivid, while deceptive ones are more abstract. The cognitive reflection test (CRT) measures an individual's ability to override intuitive but incorrect answers in favor of analytical thinking. Butavicius et al.[40] found that individuals with higher CRT scores were better at detecting phishing emails, suggesting that analytical thinkers are less susceptible to deception. However, CRT is limited by its focus on numerical reasoning and may suffer from familiarity bias due to repeated exposure, making it unsuitable as a standalone deception detection tool. The concealed information test (CIT) detects deception by analyzing physiological and behavioral reactions to crime-related stimuli. Twyman et al.[277] noted that deceptive individuals often exhibit reduced physical movement, a sign of cognitive load. Nevertheless, informed guilty individuals can use countermeasures to evade detection [225]. Despite such drawbacks, automated CIT systems have demonstrated potential in improving detection accuracy and reducing evaluator bias. Advancements in deception detection have emerged by combining cognitive, linguistic, and automated methods, each offering unique insights into how deception manifests. For instance, the finger tapping test (FTT), a cognitive approach, captures subconscious behav-

ioral changes (i.e., motor changes). Bastick et al.[26] found that in-depth exposure to fake news increased users' tapping flow by 5.15%, indicating that deception can alter motor behavior without conscious awareness. Similarly, CRT and RM detect deliberate deception by revealing logical inconsistencies and discrepancies in memory detail. Linguistic methods, such as the vector space model (VSM) and text mining, are essential for identifying deception in narrative structures. Rubin et al.[239] applied VSM to classify rhetorical structures and found that truthful narratives typically include more evidence and causal links. Liang et al.[161] employed text mining to identify personality traits and behavioral indicators associated with insider threats. Tools like Linguistic Inquiry and Word Count (LIWC) categorize words into linguistic and psychological categories. Ho et al.[106, 116, 109, 115] used these tools to analyze virtual team interactions, revealing deceptive behaviors during communication. These linguistic techniques complement cognitive approaches by quantifying narrative irregularities that may suggest deceit. Linguistic analysis has also been expanded into the automated detection of nonverbal behavior. Machine learning, in particular, has significantly advanced the field by enabling real-time analysis of both verbal and nonverbal cues. Meservy et al.[189] showed that AI could detect deception in video footage with 71.1% accuracy: far outperforming humans. However, limitations remain, such as dependency on video quality and the need for controlled settings. The integration of text mining with machine learning has proven particularly effective in identifying linguistic patterns within deceptive contexts, such as fake reviews [24], criminal statements [79], and crowdfunding scams [252]. These models, ranging from neural networks to random forests, leverage linguistic features that can reliably distinguish deceptive from truthful content.

Multimodal approaches combine textual data with other media types to provide powerful detection tools. Singh et al.[253] demonstrated improved fake news detection by jointly analyzing text and image data. Similarly, random forest models have shown promise in detecting social media deception, effectively navigating the complex linguistic patterns often found in digital content [282]. Zhou et al.[327] were pioneers in

automating the analysis of linguistic cues in written communication, finding that deceivers unconsciously tailor their language to serve deceptive goals. However, their work did not incorporate gamified elements. Building on traditional frameworks, George et al.[87] proposed an individual-centric model of deception detection within group support systems (GSS). Gamification has since emerged as a novel method for studying deception, especially in the context of CMC. Pak et al.[213] linked deception to social network structures, while Ho et al.[108] explored its impact on trust dynamics in virtual teams. Gamified experimental designs enable researchers to simulate real-world deception in interactive yet controlled environments. For example, Ho et al.[107] used gamified platforms to study insider threats by analyzing language-action cues during collaborative tasks. Derrick et al.[57] examined typing behaviors in gamified settings, revealing that message lengths and response time can indicate deceptive intent. These approaches offer deeper insight than traditional surveys by capturing behavioral nuances. Beyond detection, gamification has contributed to training and system development. Ho et al.[110] developed an interactive framework to detect lying, while Dunbar et al.[66] designed a video game to teach deception detection skills. Ho et al.[113] incorporated machine learning models like SVMs and decision trees into gamified experiments, enhancing the adaptability of detection methods. Recent research has further explored the detection of collective deception in disinformation scenarios. Ho et al.[106] simulated collaborative efforts to identify misleading content, underlying the role of group dynamics in deception detection. These applications have practical relevance for fields like digital forensics, organizational trust, and cybersecurity. Statistical methods are integral in deception research. Correlation analysis examines the relationships between variables, while t-tests determine whether there are significant group differences. For instance, Pak et al.[213] applied paired-sample t-tests to investigate deception in online games. Schroeder et al.[246] used independent t-tests to analyze the manipulation of social media profiles by job applications. Regression analysis quantifies the influence of specific variables on deceptive behaviors. Lewis et al.[153] applied logistic regression to compare deception patterns across cultures,

while Ho et al.[109, 110] used it to distinguish between truthful and deceptive digital messages. Marett et al.[175] used linear regression to uncover cultural influences on deception. ANOVA and MANOVA extend these findings by comparing multiple groups or dependent variables. Twyman et al.[279] used ANOVA to analyze vocal deception cues, while Barfar et al.[25] applied MANOVA to examine and analyze emotional and cognitive responses to disinformation. Structural equation modeling (SEM) enables the analysis of complex causal relationships. For example, Posey et al.[224] examined how organizational commitment influences insider threats, while George et al.[88] linked sender credibility to detection accuracy. Other qualitative methods, such as the Delphi method and interviews, can provide expert and in-depth perspectives on deceptive behavior. For example, Padayachee et al.[211] used Delphi to collect expert consensus on insider threats while Lensvelt et al.[150] employed interviews to explore the motivations behind deceptive survey responses.

4.3.8 Towards a Multimodal Deception Theory

As a form of computer-mediated deception, deepfakes represent a multimodal synthetic media, combining video, audio, image, and text, often partially manipulated or entirely generated using GenAI. While some deepfakes serve benign purposes (e.g., entertainment or satire), others are used maliciously for mis-/dis-information, fraud, or identity manipulation. These multimodal artifacts aim to mislead, fabricate, or obscure reality, extending deception research to multiple communication channels simultaneously. This systematic review supports the development of a Multimodal Deception Theory (MDT), which integrates human cognition, social behavior, media communication, and AI-driven technological interaction to address the complexity of modern deception. Buller and Burgoon's interpersonal deception theory (IDT)(1996) explores the behavioral dynamics between senders and deceivers during deceptive exchanges. However, IDT focuses on one channel at a time in detail, unlike MDT, which uses multimodal evidence integration to find patterns across channels.Given the inherently multimodal nature of digital information, deception research must also adopt multimodal analysis techniques. Lewis et

al.[152] emphasized that combining different types of multimedia can enhance content retrieval and semantic interpretation. As Oviatt et al.[209] noted, people prefer multimodal communication for cognitive-demanding tasks, as it reduces cognitive load. Consequently, many systems now incorporate AI-powered multimodal analysis that detects deception via language, behavior, and environmental cues. These include linguistic algorithms [110, 113] and behavioral sensors [278, 277, 279]. Multimodal detection systems have achieved an accuracy rate exceeding 83%, significantly outperforming human judgments [113, 117].Despite growing attention, there is no unified theory to account for deception in multimodal contexts.Deception involves cognitive effort [95], emotional manipulation, and is shaped by social interaction [231]. Deceivers adapt their behavior in response to feedback, adjusting their strategies to manage others' perceptions [231]. MDT asserts that deception must be understood as a cross-modal phenomenon, where inconsistencies across channels, such as mismatched audio-visual signals, serve as indicators of deceit.Deepfakes exacerbate the trust crisis in digital information. As AI-generated content becomes increasingly realistic and accessible, opportunities for malicious use grow [100]. MDT thus incorporates both behavioral insights and technological advancements to enhance detection strategies for AI-generated content, including deepfakes. As manipulated technology continues to evolve, deception research studies have become increasingly complex, transitioning from textual deception to deepfake deception that integrates multiple forms of media content. Although the multimodal deception framework integrates existing theories and encompasses multiple cues, its practicality remains theoretical. To evaluate its predictive validity and practical applicability, future research can conduct validation in diverse real-world scenarios. The research question: "What is the trajectory and the current state of deception research in information systems, and how can it inform our understanding of deepfakes?", is best addressed through the lens of MDT. MDT provides an integrative framework that evaluates deception across multiple channels of communication simultaneously, including visual, auditory, physiological, linguistic, and behavioral

signals ([2]). By situating deception within a multimodal paradigm, MDT overcomes the limitations of single-cue or single-domain approaches that dominate much of the existing research.First, MDT enables the systematic classification of deceptive cues across both physiological (e.g., micro-expressions, body movement rigidity) and computer-mediated contexts (e.g., linguistic markers, temporal irregularities). This aligns directly with the first sub-question, which seeks to identify categories of cues across contexts and modalities. MDT treats these cues not in isolation but in interaction, acknowledging that deception is rarely communicated through a single channel.Second, MDT emphasizes how context and individual differences shape the blend of cues, particularly in deepfake scenarios where synthesized media blur the boundary between genuine and manipulated content.A critical next step is empirical validation of this framework by testing models that integrate cognitive load measures, social context factors, and media richness in both laboratory and real-world settings. Such studies can reveal whether integrated cues have the potential to predict deception more effectively than single-channel approaches.Third, MDT provides a theoretical scaffold for comparing explanatory frameworks across disciplines.Traditional models, such as reality monitoring, veracity assessment, and automated linguistic analysis, tend to focus on a single domain (e.g., text, physiology). In contrast, MDT highlights the interplay of cognitive, social, and computational mechanisms that underlie deceptive behaviors. This directly informs this investigation into how different theoretical frameworks explain the intersections between cognitive science and social behavior in both F2F and digital communication contexts.Fourth, MDT is particularly relevant for deepfake research because these AI-generated deceptions deliberately manipulate multiple modalities at once. Deepfakes do not simply alter visual or auditory signals; they blur the boundaries between them, fabricating coherent multimodal performances designed to suppress "cognitive leakage" and mask unintentional cues. This makes them a reasonable stress test for MDT: if traditional deception cues lose reliability in the presence of synthetic media, then MDT must evolve to incorporate computational and adversarial

[2]https://veracity.cci.fsu.edu/

AI perspectives. By applying MDT to the phenomenon of deepfakes, this review not only maps the trajectory of deception research but also extends theoretical foundations to confront emerging AI-driven challenges.Finally, comparative research across cultures and languages is needed to determine whether cue patterns are universal or require adaptation to local norms. Understanding cross-cultural variability is crucial for designing detection tools that operate effectively globally, rather than only within specific linguistic contexts. From a security perspective, future work should focus on addressing adversarial resilience. Machine-learning detectors should be tested against countermeasures, such as generative deepfakes and text obfuscation, with defenses designed to evolve alongside the capabilities of attackers. Ultimately, large-scale deployment raises critical ethical and privacy considerations. Research into consent models, data governance, and strategies for minimizing false positives will be necessary to ensure that automated detection systems benefit society without compromising individual rights.

4.3.9 Implications and Future Work

The key components of MDT increasingly encompass both behavioral and computational studies. MDT is an interdisciplinary framework that examines multiple channels of human communication simultaneously: visual, auditory, physiological, and linguistic signals, to interpret deceptive behavior ([3]). It is considered a comprehensive theory because it integrates cognitive science, social psychology, communication, linguistics, and computational perspectives—including machine learning—to enable rigorous analysis of deception detection. This study identifies the central theoretical components and methodological frameworks of MDT. Building on this foundation, the systematic review explores the nature of deepfakes that obscure distinctions between physiological and behavioral cues, while fabricating verbal, auditory, and visual elements, often merging them into a single deceptive act. Unlike human deception, which typically results in "cognitive leakage" and unintentional cues, deepfakes minimize these

[3]https://veracity.cci.fsu.edu/

indicators. This makes deepfake detection a critical context for extending MDT, as it challenges the reliability of traditional deception cues and highlights the need to reassess deception through the integration of multiple signals. Such an approach will lead to a stronger theoretical model for identifying AI-generated deception, offering new insights into the mechanisms of deception.

Theoretical implications

Deception is becoming increasingly sophisticated and is no longer a one-size-fits-all behavior. It emerges at the intersection of individual cognition, social learning, psychological motivation, collective trustworthiness attribution, communication, and technological affordance. By integrating theories across disciplines (i.e., psychology, sociology, communication, and information systems), deception can be more fully understood as a dynamic and socially embedded process. Emerging scholars in deception research should adopt integrative frameworks that reflect this complexity. Rather than choosing between theories, future research should synthesize them to reflect the multidimensional nature of deception in the digital age. Underutilized frameworks, such as moral disengagement theory and technological affordance theory, may also offer valuable perspectives on ethical and design considerations. Cross-cultural validation and research on evolving strategies in the age of AI, deepfakes, and anonymous messaging should be a top priority. In this evolving landscape, deception is not only adapting but also becoming more sophisticated, and so must our understanding of it evolve.

Methodological implications

Among the various approaches to deception research, three noticeable shifts stand out: The first concerns detection technology, which has moved from physiological-based polygraph systems to 'online polygraph' approaches [106] that rely on language-action cues as the foundation for deception analysis. The second shift centers on research focus, expanding from traditional language-based analysis to a broader examination of deceivers'

behavior. This progression extends from identifying simple linguistic cues to conducting experimental studies on cognitive processes, and more recently, to applying machine learning for modeling deceptive behavior. In today's digital environment, this trajectory highlights the need for a multidimensional lens that incorporates multimedia elements. Relying solely on linguistic analysis is no longer sufficient; effective detection now requires integrating linguistic algorithms, visual signals, and behavioral patterns within a unified analytical framework. The third shift emphasizes that, at its core, deception research is the study of human behavior. Gamified experimental designs, which place participants in controlled yet engaging environments, have proven especially valuable for observing real-time decision-making and are increasingly adopted in the field. Looking ahead, future studies must build more sensitive models that better reflect real-world conditions, enabling researchers to capture how deceivers dynamically adapt their strategies across diverse contexts.

Privacy concerns and ethical considerations

With advances in deception detection by integrating physiological, linguistic, behavioral, and AI-driven analyses, systems continue to improve and appear promising for combating deception and deepfakes. However, their use raises significant concerns regarding privacy and ethics. Because these systems often collect sensitive data (e.g., facial expressions, eye movement, and speech patterns), they risk intruding on individuals' personal lives and privacy. In many contexts, such as border control, workplace monitoring, or consumer tracking in shopping malls, obtaining informed consent may be impractical. The wide range of potential applications only heightens these privacy risks. Cultural and linguistic bias present another major concern. Systems developed around Western cultural norms may misclassify other culturally normative behaviors as deception. For example, lowering one's gaze, considered respectful in many Middle Eastern countries, could be misinterpreted as evasive or deceptive. Such misclassifications risk reinforcing systematic bias and unfairly flagging certain groups. Organizations that adopt deception detection tools also face challenges related to transparency and accountability in handling sensitive data. Errors in

these systems can produce false positives, wrongly labeling innocent individuals as deceptive, potentially damaging their reputations, and causing long-term harm. In workplace settings, while deception detection might be introduced to promote safety and trust, inadequate data protection could expose employees to privacy breaches. If governments compel organizations to share employees' private data without consent, this would create profound ethical and legal dilemmas.

Future work

As deception detection technologies advance, future research must examine the intersection of culture, human judgment, and artificial intelligence. Current approaches risk oversimplifying deception by applying uniform models across diverse social and cultural contexts. To ensure these systems are accurate, ethical, and broadly applicable, future studies should pursue the following critical directions: understanding cross-cultural deception dynamics, bolstering cyber self-efficacy and trustworthiness attribution, and developing trustworthy human-AI collaboration frameworks. Rather than replacing human judgment, deception detection approaches should be designed to support collaborative decision-making. AI can analyze multimodal data, such as image patterns ([4]), linguistic markers, and behavioral signals, while humans contribute contextual insights, ethical reasoning, and situational awareness. Future research should investigate how employees, managers, and security professionals interact with one another and with GenAI technologies, and how such collaboration can foster trust, accountability, and the ethical use of these technologies.

4.3.10 Conclusion and Contribution

This chapter extended the perspective from CPS to HCPS, treating security and resilience as sociotechnical properties. The SLR on sociodemographic factors indicates that age, gender, education, sector, and culture shape awareness, attitudes, and behavior. Effects are clearest for age and gender and consistent for education and sector, while limits remain due to

[4] https://veracity.cci.fsu.edu/

geographic concentration and reliance on surveys. These findings justify personalized training programs that start from available attributes and combine them with role, context, and learning needs, with evaluations centered on behavior.

The second part focused on the analysis of deception cues and the shift in deception research in the AI era, performing an SLR over deception research from 2004 to 2024, identifying key definitions, theories, and methodologies across academic disciplines. Despite longstanding scholarly interest, deception as a phenomenon remains underexplored, especially in light of recent transformations in digital communication. This review has highlighted the evolution of deception research alongside technological shifts while calling for more integrative, interdisciplinary frameworks. The emergence of AI-generated misinformation, deepfakes, and post-pandemic shifts in communication underscores the urgency of deception detection. Approaches such as gamification, multimodal analysis, and automated systems offer promising paths forward. Building on these insights, this study contributes an MDT, a framework that captures the complexity of contemporary deception by integrating cognitive, linguistic, and computational perspectives.

The contributions are threefold. First, an action-oriented synthesis of evidence demonstrates how sociodemographic factors can inform the personalization of training in healthcare professional settings. Second, a unified reading of deception across physical and digital contexts that clarifies when traditional cues remain useful and when they lose reliability in the presence of synthetic media. Third, the definition of MDT as a theoretical basis for multimodal detection methods and for interventions that combine personalization with adversarial robustness. Together, the SLR and the MDT perspectives show that human factors are integral to a system's attack surface and resilience. They open the door to future empirical work that links sociodemographic signals and multimodal cues to measurable security behaviors and that validates personalized training and detection strategies in operational settings.

Chapter 5

Conclusion

5.1 Unified vision

5.1.1 Integrating metrics, architectures, and human factors

This dissertation presents a coherent view of security for cyber–physical and human–cyber–physical systems. The central idea is that measurable evidence, enforceable architectures, and human context must operate together as parts of a single assurance process.

First, a systematic analysis distilled a validated catalogue of security metrics for Industrial CPS. The study combined literature collection, domain filtering, and formal vetting under Conditions for Sound Security Metrics. The result is a set of metrics with explicit scope, data requirements, and computation methods that support both design-time assurance and run-time monitoring. The catalogue and the method to obtain it provide a reusable pathway for other CPS domains.

Second, two architectures demonstrated how metrics can drive action. GRAPH4 links attack graph analytics with programmable data planes, enabling monitoring and mitigation to follow the evolving exposure of the network. The proposed architecture shows that metric computation can be placed where it matters, with bounded overhead and with clear triggers for control plane reaction. Pk–IOTA addresses certificate publication, validation, and revocation for OPC UA by combining P4 enforcement in

the network with a DAG ledger as an authentic record. Experiments on a physical testbed indicate that in–network checks introduce modest latency during handshakes and that both Layer 1 and Layer 2 configurations on IOTA remain viable as deployments scale. The analysis clarifies the trade-offs between simplicity and richer lifecycle automation.

Third, the work extends trust to collaborative machine learning. DAGTrustFL adapts the tip selection mechanism of a DAG ledger to weight model updates based on trust signals and to mitigate the influence of suspected poisoning. The evaluation on an image classification task shows improved selectivity under adversarial pressure while preserving utility. Ledger costs scale well with the number of participants, and throughput remains stable compared to blockchain baselines. Together, these results position DAGs as a practical substrate for trust management in federated settings.

Fourth, the thesis brings the human factor into scope. The HCPS perspective explains why operators, supervisors, and collaborating stakeholders are not external to security but act as integral components. A systematic review of sociodemographic influences reveals that age, education, sector, culture, and gender are often correlated with awareness and behavior, supporting targeted and non-discriminatory training strategies. A second review examines deception in digital environments, with a specific focus on deepfakes, and motivates a multimodal theory that integrates linguistic, behavioral, physiological, and contextual indicators. These findings link technical assurance with organizational practice and personalized capacity building.

The contributions are therefore complementary. Metrics provide observable and comparable signals. Architectures turn those signals into timely enforcement and auditable records. Human-centred analyses guide adoption, training, and resilience in the face of manipulation. Treated together, these elements support measurable, explainable, and scalable security in HCPS.

5.1.2 Towards self-defending and adaptive CPS

The results point to a practical control loop for self-defending systems. Observation collects validated metrics at the right place in the stack.

Inference combines models such as attack graphs and trust evaluations to interpret those signals. Enforcement utilizes programmable switches and ledgers to implement policies, validate identities, and record decisions for audit purposes. Learning closes the loop with feedback into models, configurations, and training programmes that reflect incidents and near misses.

This loop enhances three key properties that are important in both industrial and societal contexts. Scalability follows from placing computation in the data plane and from the parallelism of DAG ledgers. Verifiability follows from authentic records, reproducible metrics, and transparent decision paths. Human alignment follows from HCPS principles, where interfaces, training, and governance reflect how people actually work and decide under time and information constraints.

5.2 Future research directions

Several directions can be explored to consolidate the foundations laid out in this thesis. A first line of work involves standardizing security metrics for CPS and ICPS using explicit soundness conditions, open definitions, and reference datasets. Shared taxonomies, machine-readable schemas, and benchmark suites would improve comparability and accelerate adoption in certification workflows.

A second line concerns trust management for distributed learning. The adaptation of tip selection shows that a DAG can act as a trust engine. Future work should define a common representation for trust signals, extend poisoning-aware indicators to richer tasks, and evaluate governance choices for decentralized aggregation with smart contracts once Layer 1 execution becomes widely available over a DAG-based ledger, such as IOTA. Reference implementations and interoperability profiles would support cross-domain deployments.

A third line addresses deception in human-centered systems. The proposed multimodal perspective should be validated through controlled and field studies that combine language, interaction patterns, and sensor data, all under clear privacy and ethics constraints. Robustness against adver-

sarial content and cross-cultural generalization requires particular care. The outcome should inform training that adapts to roles and sociodemographic signals without resorting to stereotypes.

5.2.1 Bridging theory and deployment in real infrastructures

Translating the results into practice calls for pilots in live environments. For GRAPH4, this means integrating with production controllers and switches, linking attack graph generation with change management, and quantifying benefits during red team exercises. For Pk–IOTA, it means onboarding OPC UA Part 21, assessing public versus private node strategies, and measuring lifecycle gains for issuance and revocation at plant scale. For DAGTrustFL, it means embedding the trust layer into existing FL frameworks, profiling costs on constrained devices, and validating resistance to stronger threat models and non-IID data.

On the human side, organizations should move from one-off awareness to continuous, role-aware training with feedback loops. Studies that correlate sociodemographic factors with measurable behaviors can inform content and cadence, while deception-aware drills can enhance sensemaking for ambiguous incidents. Metrics should track competence growth and guide the retirement of ineffective practices.

Beyond individual prototypes and reviews, this dissertation advances three shifts in how industrial and human–cyber–physical systems can be secured: from static security by design to measurement-centered assurance where validated metrics anchor every decision, from configuration scattered across endpoints to programmable enforcement with verifiable records on distributed ledgers, and from identity as a gatekeeping prerequisite to behavior-grounded trust that preserves openness while sustaining accountability. Overall, the thesis argues for security that is observable, enforceable, and learnable. The methods and architecture presented show that it is possible to connect formal measurement with programmable enforcement and with human-centered practice. The next steps involve codifying these connections into shared frameworks and demonstrating their value in diverse, real operational contexts.

Bibliography

[1] Syed Ghazanfar Abbas et al. "{SAIN}: Improving {ICS} Attack Detection Sensitivity via {State-Aware} Invariants". In: 33rd USENIX Security Symposium (USENIX Security 24). 2024, pp. 6597–6613.

[2] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. "A survey of network anomaly detection techniques". In: Journal of Network and Computer Applications 60 (2016), pp. 19–31.

[3] Rana Khudhair Abbas Ahmed. "Security metrics and the risks: an overview". In: International Journal of Computer Trends and Technology (IJCTT) 41 (2016), pp. 106–112.

[4] Andreas Aigner and Abdelmajid Khelil. "A Benchmark of Security Metrics in Cyber-Physical Systems". In: 2020 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops). IEEE. 2020, pp. 1–6.

[5] Assiya Akli and Khalid Chougdali. "A Survey on IOTA Based Technology for Enhanced IoT Security". In: 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE. 2024, pp. 1–6.

[6] Amir Al Sadi et al. "Unleashing Dynamic Pipeline Reconfiguration of P4 Switches for Efficient Network Monitoring". In: IEEE Transactions on Network and Service Management (2024).

[7] Abdullah Alabdulatif, Navod Neranjan Thilakarathne, and Zaharaddeen Karami Lawal. "A Review on Security and Privacy Issues Pertaining to Cyber-Physical Systems in the Industry 5.0 Era." In: Computers, Materials & Continua 80.3 (2024).

[8] Marfua Alanazi, Mark Freeman, and Holly Tootell. "Exploring the factors that influence the cybersecurity behaviors of young adults". In: Computers in Human Behavior 136 (2022), p. 107376.

[9] Nikolaos Alexopoulos. “New Approaches to Software Security Metrics and Measurements”. PhD thesis. Technische Universität Darmstadt, 2022.

[10] Yara Alghofaili and Murad A Rassam. “A trust management model for IoT devices and services based on the multi-criteria decision-making approach and deep long short-term memory technique”. In: Sensors 22.2 (2022), p. 634.

[11] Rasim Alguliyev, Yadigar Imamverdiyev, and Lyudmila Sukhostat. “Cyber-physical systems and their security issues”. In: Computers in Industry 100 (2018), pp. 212–223.

[12] Amir Alsadi et al. “A Security Monitoring Architecture based on Data Plane Programmability”. In: 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit). IEEE. 2021, pp. 389–394.

[13] Uchenna P Daniel Ani, Hongmei He, and Ashutosh Tiwari. “A framework for Operational Security Metrics Development for industrial control environment”. In: Journal of Cyber Security Technology 2.3-4 (2018), pp. 201–237.

[14] Mohd Anwar et al. “Gender difference and employees’ cybersecurity behaviors”. In: Computers in Human Behavior 69 (2017), pp. 437–443.

[15] Arc Insight. OPC Technology Well-positioned for Further Growth in Tomorrow’s Connected World. [Online]. 2018. url: https://opcfoundation.org/wp-content/uploads/2018/02/ARC-Report-OPC-Installed-Base-Insights.pdf.

[16] Duncan Ki-Aries and Shamal Faily. “Persona-centred information security awareness”. In: Computers & security 70 (2017), pp. 663–674.

[17] Asier 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. issn: 0166-3615. doi: https://doi.org/10.1016/j.compind.2022.103802. url: https://www.sciencedirect.com/science/article/pii/S0166361522001981.

[18] Louise Axon and Michael Goldsmith. “PB-PKI: A privacy-aware blockchain-based PKI”. In: 14th International Conference on Security and Cryptography (SECRYPT 2017). Vol. 6. SciTePress. 2016.

[19] M Azuwa et al. “Technical security metrics model in compliance with ISO/IEC 27001 standard”. In: International Journal of CyberSecurity and Digital Forensics 1.4 (2012), pp. 280–288.

[20] Leemon Baird, Mance Harmon, and Paul Madsen. “Hedera: A public hashgraph network & governing council”. In: White Paper 1.1 (2019), pp. 9–10.

[21] Dennik Baltuttis, Timm Teubner, and Marc TP Adam. “A typology of cybersecurity behavior among knowledge workers”. In: Computers & Security 140 (2024), p. 103741.

[22] Albert Bandura et al. “Social foundations of thought and action”. In: Englewood Cliffs, NJ 1986.23-28 (1986), p. 2.

[23] Albert Bandura and Richard H Walters. Social learning theory. Vol. 1. Prentice hall Englewood Cliffs, NJ, 1977.

[24] Snehasish Banerjee, Alton YK Chua, and Jung-Jae Kim. “Don’t be deceived: Using linguistic analysis to learn how to discern online review authenticity”. In: Journal of the Association for Information Science and Technology 68.6 (2017), pp. 1525–1538.

[25] Arash Barfar. “Cognitive and affective responses to political disinformation in Facebook”. In: Computers in Human Behavior 101 (2019), pp. 173–179.

[26] Zach Bastick. “Would you notice if fake news changed your behavior? An experiment on the unconscious effects of disinformation”. In: Computers in human behavior 116 (2021), p. 106633.

[27] Seema Gupta Bhol, JR Mohanty, and Prasant Kumar Pattnaik. “Taxonomy of cyber security metrics to measure strength of cyber security”. In: Materials Today: Proceedings 80 (2023), pp. 2274–2279.

[28] Lirui Bi, Tasiu Muazu, and Omaji Samuel. “Iot: a decentralized trust management system using blockchain-empowered federated learning”. In: Sustainability 15.1 (2022), p. 374.

[29] Sharon Boeyen et al. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280. May 2008. doi: 10.17487/RFC5280. url: https://www.rfc-editor.org/info/rfc5280.

[30] Pat Bosshart et al. “P4: Programming Protocol-Independent Packet Processors”. In: 44.3 (July 2014), pp. 87–95. issn: 0146-4833. doi: 10.1145/2656877.2656890. url: https://doi.org/10.1145/2656877.2656890.

[31] Pat Bosshart et al. “P4: Programming protocol-independent packet processors”. In: ACM SIGCOMM Computer Communication Review 44.3 (2014), pp. 87–95.

[32] Michele Bottone, Franco Raimondi, and Giuseppe Primiero. “Multi-agent based simulations of block-free distributed ledgers”. In: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE. 2018, pp. 585–590.

[33] Wayne Boyer and Miles McQueen. “Ideal based cyber security technical metrics for control systems”. In: Critical Information Infrastructures Security: Second International Workshop, CRITIS 2007, Málaga, Spain, October 3-5, 2007. Revised Papers 2. Springer. 2008, pp. 246–260.

[34] W Krag Brotby and Gary Hinson. Pragmatic security metrics: applying metametrics to information security. CRC Press, 2013.

[35] BSI. OPC UA Security Analysis. Tech. Rep. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/OPCUA/OPCUA.html. Bundesamt für Sicherheit in der Informationstechnik, 2017.

[36] David B Buller and Judee K Burgoon. “Interpersonal deception theory”. In: Communication theory 6.3 (1996), pp. 203–242.

[37] David B Buller et al. “Interpersonal deception VII: Behavioral profiles of falsification, equivocation, and concealment”. In: Journal of language and social psychology 13.4 (1994), pp. 366–395.

[38] Judee K Burgoon et al. “Detecting deception through linguistic analysis”. In: International Conference on Intelligence and Security Informatics. Springer. 2003, pp. 91–101.

[39] AJ Burns et al. “Going beyond deterrence: A middle-range theory of motives and controls for insider computer abuse”. In: Information Systems Research 34.1 (2023), pp. 342–362.

[40] Marcus Butavicius, Ronnie Taib, and Simon J Han. “Why people keep falling for phishing scams: The effects of time pressure and deception cues on the detection of phishing emails”. In: Computers & Security 123 (2022), p. 102937.

[41] Bin Cao et al. “Performance analysis and comparison of PoW, PoS and DAG based blockchains”. In: Digital Communications and Networks 6.4 (2020), pp. 480–485.

[42] Di Cao et al. “Understanding distributed poisoning attack in federated learning”. In: 2019 IEEE 25th international conference on parallel and distributed systems (ICPADS). IEEE. 2019, pp. 233–239.

[43] Xiaoyu Cao et al. “Fltrust: Byzantine-robust federated learning via trust bootstrapping”. In: arXiv preprint arXiv:2012.13995 (2020).

[44] Bruno Casella et al. “Experimenting with normalization layers in federated learning on non-iid scenarios”. In: IEEE Access 12 (2024), pp. 47961–47971.

[45] Ebrima N Ceesay, K Myers, and Paul Watters. “Human-centered strategies for cyber-physical systems security”. In: (2018).

[46] Stefano Ceri, Georg Gottlob, Letizia Tanca, et al. “What you always wanted to know about Datalog(and never dared to ask)”. In: IEEE transactions on knowledge and data engineering 1.1 (1989), pp. 146–166.

[47] Chunjiang Che et al. “A decentralized federated learning framework via committee mechanism with convergence guarantee”. In: IEEE Transactions on Parallel and Distributed Systems 33.12 (2022), pp. 4783–4800.

[48] Xiaoqi Chen. “Implementing AES encryption on programmable switches via scrambled lookup tables”. In: Proceedings of the Workshop on Secure Programmable Network Infrastructure. 2020, pp. 8–14.

[49] Yan Chen and Fatemeh Mariam Zahedi. “Individuals’ internet security perceptions and behaviors”. In: Mis Quarterly 40.1 (2016), pp. 205–222.

[50] Zesheng Chen and Chuanyi Ji. “Measuring network-aware worm spreading ability”. In: IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications. IEEE. 2007, pp. 116–124.

[51] Elizabeth Chew et al. Sp 800-55 rev. 1. performance measurement guide for information security. 2008.

[52] The P4 Language Consortium. P4Runtime Specification. 2020. url: https://p4.org/p4-spec/p4runtime/v1.3.0/P4Runtime-Spec.pdf.

[53] Mauro Conti, Denis Donadel, and Federico Turrin. "A survey on industrial control system testbeds and datasets for security research". In: IEEE Communications Surveys & Tutorials 23.4 (2021), pp. 2248–2294.

[54] Mauro Conti et al. "A survey on security challenges and solutions in the IOTA". In: Journal of Network and Computer Applications 203 (2022), p. 103383.

[55] Markus Dahlmanns et al. "Easing the conscience with OPC UA: An internet-wide study on insecure deployments". In: Proceedings of the ACM Internet Measurement Conference. 2020, pp. 101–110.

[56] John Demme et al. "Side-channel vulnerability factor: A metric for measuring information leakage". In: ACM SIGARCH computer architecture news 40.3 (2012), pp. 106–117.

[57] Douglas C Derrick et al. "Detecting deceptive chat-based communication using typing behavior and message cues". In: ACM Transactions on Management Information Systems (TMIS) 4.2 (2013), pp. 1–21.

[58] Damu Ding, Marco Savi, and Domenico Siracusa. "Tracking Normalized Network Traffic Entropy to Detect DDoS Attacks in P4". In: IEEE Transactions on Dependable and Secure Computing (2021).

[59] Danny Dolev and Andrew Yao. "On the security of public key protocols". In: IEEE Transactions on information theory 29.2 (1983), pp. 198–208.

[60] Zhongli Dong et al. "Dagbench: A performance evaluation framework for dag distributed ledgers". In: 2019 IEEE 12th international conference on cloud computing (CLOUD). IEEE. 2019, pp. 264–271.

[61] Zhongxu Dong et al. "Trustworthy VANET: Hierarchical DAG-Based Blockchain Solution with Proof of Reputation Consensus Algorithm". In: 2023 IEEE International Conference on Blockchain (Blockchain). IEEE. 2023, pp. 127–132.

[62] George T Doran et al. “There’s a SMART way to write management’s goals and objectives”. In: Management review 70.11 (1981), pp. 35–36.

[63] Evaldas Drąsutis. “Iota smart contracts”. In: (Nov. 2021).

[64] Jannik Dreier et al. “Formally and practically verifying flow properties in industrial systems”. In: Computers & Security 86 (2019), pp. 453–470.

[65] Jannik Dreier et al. “Formally verifying flow properties in industrial systems”. In: SECRYPT 2017-14th International Conference on Security and Cryptography. 2017, pp. 55–66.

[66] Norah E Dunbar et al. “Reliable deception cues training in an interactive video game”. In: Computers in Human Behavior 85 (2018), pp. 74–85.

[67] Carl Ellison and Bruce Schneier. “Ten risks of PKI: What you’re not being told about public key infrastructure”. In: Comput Secur J 16.1 (2000), pp. 1–7.

[68] Nebrase Elmrabit et al. “Insider threat risk prediction based on Bayesian network”. In: Computers & Security 96 (2020), p. 101908.

[69] Simon Yusuf Enoch et al. “Composite metrics for network security analysis”. In: arXiv preprint arXiv:2007.03486 (2020).

[70] Alessandro Erba, Anne Müller, and Nils Ole Tippenhauer. “Security Analysis of Vendor Implementations of the OPC UA Protocol for Industrial Control Systems”. In: Proceedings of the 4th Workshop on CPS & IoT Security and Privacy. 2022, pp. 1–13.

[71] Alessandro Erba and Nils Ole Tippenhauer. “Assessing model-free anomaly detection in industrial control systems against generic concealment attacks”. In: Proceedings of the 38th Annual Computer Security Applications Conference. 2022, pp. 412–426.

[72] Caixiang Fan et al. “Performance analysis of the IOTA DAG-based distributed ledger”. In: ACM Transactions on Modeling and Performance Evaluation of Computing Systems 6.3 (2021), pp. 1–20.

[73] Minghong Fang et al. “Local model poisoning attacks to {Byzantine-Robust} federated learning”. In: 29th USENIX security symposium (USENIX Security 20). 2020, pp. 1605–1622.

[74] Xiaoqin Feng et al. “Regulatable and Hardware-Based Proof of Stake to Approach Nothing at Stake and Long Range Attacks”. In: IEEE Transactions on Services Computing 16 (2023), pp. 2114–2125. doi: 10.1109/TSC.2022.3201568.

[75] José Luis Fernández-Alemán et al. “Security and privacy in electronic health records: A systematic literature review”. In: Journal of biomedical informatics 46.3 (2013), pp. 541–562.

[76] César Flores et al. “Human factors for cybersecurity awareness in a remote work environment”. In: 9th International Conference on Information Systems Security and Privacy (ICISSP 2023), Lisbon, Portugal, 22–24 February 2023. Vol. 1. SciTePress. 2023, pp. 608–616.

[77] Edwin Donald Frauenstein et al. “Unraveling the behavioral influence of social media on phishing susceptibility: A Personality-Habit-Information Processing model”. In: Information & Management 60.7 (2023), p. 103858.

[78] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. “Certcoin: A namecoin based decentralized authentication system”. In: Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep 6 (2014), pp. 46–56.

[79] Christie M Fuller, David P Biros, and Rick L Wilson. “Decision support for determining veracity via linguistic-based cues”. In: Decision Support Systems 46.3 (2009), pp. 695–703.

[80] Christopher P Furner and Joey F George. “Cultural determinants of media choice for deception”. In: Computers in Human Behavior 28.4 (2012), pp. 1427–1438.

[81] Luigi Gallo et al. “The human factor in phishing: Collecting and analyzing user behavior when reading emails”. In: Computers & Security 139 (2024), p. 103671.

[82] Ángel Fernández Gambín et al. “Deepfakes: current and future trends”. In: Artificial Intelligence Review 57.3 (2024), p. 64.

[83] Sam Gao, Mark Handley, and Stefano Vissicchio. “Stats 101 in p4: towards in-switch anomaly detection”. In: Proceedings of the twentieth ACM workshop on hot topics in networks. 2021, pp. 84–90.

[84] Abba Garba et al. “LightCERT4IoTs: Blockchain-based lightweight certificates authentication for IoT applications”. In: IEEE Access 11 (2023), pp. 28370–28383.

[85] Joey F George and John R Carlson. “Lying at work: A deceiver’s view of media characteristics”. In: Communications of the Association for Information Systems 27.1 (2010), p. 44.

[86] Joey F George, Gabriel Giordano, and Patti A Tilley. “Website credibility and deceiver credibility: Expanding prominence-interpretation theory”. In: Computers in Human Behavior 54 (2016), pp. 83–93.

[87] Joey F George, Kent Marett, and Gabriel Giordano. “Deception: Toward an individualistic view of group support systems”. In: Journal of the Association for Information Systems 9.10 (2008), p. 3.

[88] Joey F George, Patti Tilley, and Gabriel Giordano. “Sender credibility and deception detection”. In: Computers in Human Behavior 35 (2014), pp. 1–11.

[89] Joey F George et al. “The effects of communication media and culture on deception detection accuracy”. In: MIS quarterly 42.2 (2018), pp. 551–576.

[90] Molka Gharbaoui et al. “An experimental study on latency-aware and self-adaptive service chaining orchestration in distributed NFV and SDN infrastructures”. In: Computer Networks 208 (2022), p. 108880.

[91] Anousheh Gholami, Nariman Torkzaban, and John S Baras. “Trusted decentralized federated learning”. In: 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE. 2022, pp. 1–6.

[92] Ashwinraj Giriraj, Sherif Haggag, and Hussein Haggag. “Human centric framework for customising and producing effective cybersecurity training materials”. In: Joint 4th International Workshop on Experience with SQuaRE Series and Its Future Direction and 1st Asia-Pacific Software Engineering and Diversity, Equity, and Inclusion Workshop, IWESQ 2022+ APSEDEI 2022, Tokyo, Japan, December 6, 2022. 2022, pp. 69–77.

[93] Jan Goebel, Thorsten Holz, and Carsten Willems. “Measurement and analysis of autonomous spreading malware in a university environment”. In: Detection of Intrusions and Malware, and Vulnerability Assessment: 4th International Conference, DIMVA 2007

Lucerne, Switzerland, July 12-13, 2007 Proceedings 4. Springer. 2007, pp. 109–128.

[94] Erving Goffman. “The presentation of self in everyday life”. In: Social theory re-wired. Routledge, 2023, pp. 450–459.

[95] Victor A Gombos. “The cognition of deception: The role of executive processes in producing lies”. In: Genetic, social, and general psychology monographs 132.3 (2006), pp. 197–214.

[96] Giacomo Gori et al. “A systematic analysis of security metrics for industrial cyber–physical systems”. In: Electronics 13.7 (2024), p. 1208.

[97] Giacomo Gori et al. “GRAPH4: A Security Monitoring Architecture Based on Data Plane Anomaly Detection Metrics Calculated over Attack Graphs”. In: Future Internet 15.11 (2023), p. 368.

[98] Giacomo Gori et al. “Metrics for Cyber-Physical Security: A call to action”. In: 2022 International Symposium on Networks, Computers and Communications (ISNCC). IEEE. 2022, pp. 1–4.

[99] Giacomo Gori et al. “Unraveling The Link Between Sociodemographics and Cybersecurity”. In: (2025).

[100] Matthew Groh et al. “Deepfake detection by human crowds, machines, and machine-informed crowds”. In: Proceedings of the National Academy of Sciences 119.1 (2022), e2110013119.

[101] Guofei Gu et al. “Measuring intrusion detection capability: An information-theoretic approach”. In: Proceedings of the 2006 ACM Symposium on Information, computer and communications security. 2006, pp. 90–101.

[102] Fengyang Guo et al. “Characterizing IOTA tangle with empirical data”. In: GLOBECOM 2020-2020 IEEE Global Communications Conference. IEEE. 2020, pp. 1–6.

[103] Shaoyong Guo et al. “Sandbox computing: A data privacy trusted sharing paradigm via blockchain and federated learning”. In: IEEE Transactions on Computers 72.3 (2022), pp. 800–810.

[104] Achref Haddaji, Samiha Ayed, and Lamia Chaari. “Federated learning with blockchain approach for trust management in IoV”. In: International Conference on Advanced Information Networking and Applications. Springer. 2022, pp. 411–423.

[105] Jean-Pierre Hauet. “ISA99/IEC 62443: a solution to cyber-security issues?” In: ISA Automation Conference. 2012.

[106] Shuyuan Ho, Jeffrey Nickerson, and Qian Zhang. “Hive Mind Online: Collective Sensing in Times of Disinformation”. In: Journal of Digital Social Research 4.4 (2022), pp. 89–129.

[107] Shuyuan Mary Ho. “Leader member exchange: An interactive framework to uncover a deceptive insider as revealed by human sensors”. In: (2019).

[108] Shuyuan Mary Ho and Izak Benbasat. “Dyadic attribution model: A mechanism to assess trustworthiness in virtual organizations”. In: Journal of the Association for Information Science and Technology 65.8 (2014), pp. 1555–1576.

[109] Shuyuan Mary Ho and Jeffrey T Hancock. “Computer-mediated deception: Collective language-action cues as stigmergic signals for computational intelligence”. In: (2018).

[110] Shuyuan Mary Ho and Jeffrey T Hancock. “Context in a bottle: Language-action cues in spontaneous computer-mediated deception”. In: Computers in Human Behavior 91 (2019), pp. 33–41.

[111] Shuyuan Mary Ho, Jeffrey T Hancock, and Cheryl Booth. “Ethical dilemma: Deception dynamics in computer-mediated group communication”. In: Journal of the Association for Information Science and Technology 68.12 (2017), pp. 2729–2742.

[112] Shuyuan Mary Ho, Michelle Kaarst-Brown, and Izak Benbasat. “Trustworthiness attribution: Inquiry into insider threat detection”. In: Journal of the Association for Information Science and Technology 69.2 (2018), pp. 271–280.

[113] Shuyuan Mary Ho et al. “Computer-mediated deception: Strategies revealed by language-action cues in spontaneous communication”. In: Journal of Management Information Systems 33.2 (2016), pp. 393–420.

[114] Shuyuan Mary Ho et al. “Demystifying insider threat: Language-action cues in group dynamics”. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE. 2016, pp. 2729–2738.

[115] Shuyuan Mary Ho et al. “Gender deception in asynchronous online communication: A path analysis”. In: Information Processing & Management 53.1 (2017), pp. 21–41.

[116] Shuyuan Mary Ho et al. “Insider threat: Language-action cues in group dynamics”. In: Proceedings of the 2015 ACM SIGMIS Conference on Computers and People Research. 2015, pp. 101–104.

[117] Shuyuan Mary Ho et al. “Real or Spiel? A decision tree approach for automated detection of deceptive language-action cues”. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE. 2016, pp. 3706–3715.

[118] Shuyuan Mary Ho et al. “Saint or Sinner? Language-action cues for modeling deception using support vector machines”. In: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. Springer. 2016, pp. 325–334.

[119] Shuyuan Mary Ho et al. “Synthetic Lies, Digital Truths: A Systematic Review of Computer-Mediated Deception Research in the Era of AI and Deepfakes”. In: (2025).

[120] Geert Hofstede. “Dimensionalizing cultures: The Hofstede model in context”. In: Online readings in psychology and culture 2.1 (2011), p. 8.

[121] John Homer, Xinming Ou, and David Schmidt. “A sound and practical approach to quantifying security risk in enterprise networks”. In: Kansas State University Technical Report (2009), pp. 1–15.

[122] Jin B Hong et al. “Dynamic security metrics for measuring the effectiveness of moving target defense techniques”. In: Computers & Security 79 (2018), pp. 33–52.

[123] Wilson Cheong Hin Hong et al. “The influence of social education level on cybersecurity awareness and behaviour: a comparative study of university students and working graduates”. In: Education and information technologies 28.1 (2023), pp. 439–470.

[124] Yuxiang Hong and Steven Furnell. “Understanding cybersecurity behavioral habits: Insights from situational support”. In: Journal of Information Security and Applications 57 (2021), p. 102710.

[125] Abdulmalik Humayed et al. “Cyber-physical systems security—A survey”. In: IEEE Internet of Things Journal 4.6 (2017), pp. 1802–1831.

[126] Ladislav Huraj et al. “Measuring Cyber Security Awareness: A Comparison between Computer Science and Media Science Students.” In: TEM Journal 12.2 (2023).

[127] Moses Ike et al. “Scaphy: Detecting modern ics attacks by correlating behaviors in scada and physical”. In: 2023 IEEE Symposium on Security and Privacy (SP). IEEE. 2023, pp. 20–37.

[128] Industry ARC. Increasing Penetration of Digitalization Across Industries to Optimize Operations in Terms of Productivity, Safety and Sustainability, Is Set to Create Significant Opportunities for Global OPC Server Market. [Online]. June 2022. url: https://www.industryarc.com/PressRelease/2753/opc-server-market-research.html.

[129] Industry ARC. OPC-UA Network Market - Forecast(2024 - 2030). [Online]. Apr. 2024. url: https://www.industryarc.com/Report/19409/opc-ua-network-market.html.

[130] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). ISO/IEC 62443-3-3: Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels. Available from ISO, https://www.iso.org/standard/65696.html. ISO/IEC. 2013.

[131] Jaakko Jalkanen. “Is human the weakest link in information security?: systematic literature review”. MA thesis. University of Jyväskylä, 2019.

[132] Farhana Javed et al. “Trustworthy reputation for federated learning in o-ran using blockchain and smart contracts”. In: IEEE Open Journal of the Communications Society (2025).

[133] Najeeb Moharram Jebreel and Josep Domingo-Ferrer. “Fl-defender: Combating targeted attacks in federated learning”. In: Knowledge-Based Systems 260 (2023), p. 110178.

[134] Jongkil Jeong et al. “Towards an improved understanding of human factors in cybersecurity”. In: 2019 IEEE 5th international conference on collaboration and internet computing (CIC). IEEE. 2019, pp. 338–345.

[135] Benjamin Johnson et al. “Metrics for Measuring ISP Badness: The Case of Spam: (Short Paper)”. In: Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers 16. Springer. 2012, pp. 89–97.

[136] Gajasri Karthikeyan and Stefan Heiss. “Pki and user access rights management for opc ua based applications”. In: 2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA). Vol. 1. IEEE. 2018, pp. 251–257.

[137] Florian Kohnhäuser, Sten Grüner, and Jens Heuschkel. “Secure Onboarding of IIoT Devices using OPC UA”. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE. 2022, pp. 1–4.

[138] Florian Kohnhäuser et al. “On the feasibility and performance of secure OPC UA communication with IIoT Devices”. In: International Conference on Computer Safety, Reliability, and Security. Springer. 2022, pp. 189–203.

[139] Florian Kohnhäuser et al. “On the security of IIoT deployments: An investigation of secure provisioning solutions for OPC UA”. In: IEEE access 9 (2021), pp. 99299–99311.

[140] Dezhang Kong et al. “Toward Security-Enhanced In-band Network Telemetry in Programmable Networks”. In: IEEE Transactions on Network and Service Management (2024).

[141] Ana Kovačević, Nenad Putnik, and Oliver Tošković. “Factors related to cyber security behavior”. In: Ieee Access 8 (2020), pp. 125140–125148.

[142] Murat Yasin Kubilay, Mehmet Sabir Kiraz, and Hacı Ali Mantar. “CertLedger: A new PKI model with certificate transparency based on blockchain”. In: Computers & Security 85 (2019), pp. 333–352.

[143] Naveen Kumar et al. “Detecting anomalous online reviewers: An unsupervised approach using mixture models”. In: Journal of Management Information Systems 36.4 (2019), pp. 1313–1346.

[144] Ashwin Lall et al. “Data streaming algorithms for estimating entropy of network traffic”. In: ACM SIGMETRICS Performance Evaluation Review 34.1 (2006), pp. 145–156.

[145] Ruggero Lanotte et al. “Formal impact metrics for cyber-physical attacks”. In: 2021 IEEE 34th Computer Security Foundations Symposium (CSF). IEEE. 2021, pp. 1–16.

[146] Sofiane Laridi, Gregory Palmer, and Kam-Ming Mark Tam. “Enhanced federated anomaly detection through autoencoders using summary statistics-based thresholding”. In: Scientific Reports 14.1 (2024), p. 26704.

[147] Chih-Chen Lee, Robert B Welker, and Marcus D Odom. “Features of computer-mediated, text-based messages that support automatable, linguistics-based indicators for deception detection”. In: Journal of Information Systems 23.1 (2009), pp. 5–24.

[148] Claire Seungeun Lee and Ji Hye Kim. “Latent groups of cybersecurity preparedness in Europe: Sociodemographic factors and country-level contexts”. In: Computers & Security 97 (2020), p. 101995.

[149] Colin LeMahieu. “Nano: A feeless distributed cryptocurrency network”. In: Nano [Online resource]. URL: https://nano. org/en/whitepaper (date of access: 24.03. 2018) 16 (2018), p. 17.

[150] Gerty JLM Lensvelt-Mulders and Hennie R Boeije. “Evaluating compliance with a computer assisted randomized response technique: a qualitative study into the origins of lying and cheating”. In: Computers in Human Behavior 23.1 (2007), pp. 591–608.

[151] David John Leversage and Eric James Byres. “Estimating a system’s mean time-to-compromise”. In: IEEE Security & Privacy 6.1 (2008), pp. 52–60.

[152] Michael S Lew et al. “Content-based multimedia information retrieval: State of the art and challenges”. In: ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 2.1 (2006), pp. 1–19.

[153] Carmen C Lewis and Joey F George. “Cross-cultural deception in social networking sites and face-to-face communication”. In: Computers in Human Behavior 24.6 (2008), pp. 2945–2964.

[154] Cody Lewis, Vijay Varadharajan, and Nasimul Noman. “Attacks against federated learning defense systems and their mitigation”. In: Journal of Machine Learning Research 24.30 (2023), pp. 1–50.

[155] Karen Lewison and Francisco Corella. “Backing rich credentials with a blockchain PKI”. In: Pomcor. com (2016).

[156] Naipeng Li et al. “A partitioned DAG distributed ledger with local consistency for vehicular reputation management”. In: Wireless Communications and Mobile Computing 2022.1 (2022), p. 6833535.

[157] Qinbin Li et al. “A survey on federated learning systems: Vision, hype and reality for data privacy and protection”. In: IEEE Transactions on Knowledge and Data Engineering 35.4 (2021), pp. 3347–3366.

[158] Tian Li et al. “Federated learning: Challenges, methods, and future directions”. In: IEEE signal processing magazine 37.3 (2020), pp. 50–60.

[159] Xueyang Li et al. “Efficiently achieving privacy preservation and poisoning attack resistance in federated learning”. In: IEEE Transactions on Information Forensics and Security (2024).

[160] Yixin Li et al. “Direct acyclic graph-based ledger for Internet of Things: Performance and security analysis”. In: IEEE/ACM Transactions on Networking 28.4 (2020), pp. 1643–1656.

[161] Nan Liang, David P Biros, and Andy Luse. “An empirical validation of malicious insider characteristics”. In: Journal of Management Information Systems 33.2 (2016), pp. 361–392.

[162] Tian Lin et al. “Susceptibility to spear-phishing emails: Effects of internet user demographics and email content”. In: ACM Transactions on Computer-Human Interaction (TOCHI) 26.5 (2019), pp. 1–28.

[163] Richard Lippmann et al. “Validating and restoring defense in depth using attack graphs”. In: MILCOM 2006-2006 IEEE Military Communications Conference. IEEE. 2006, pp. 1–10.

[164] Ji Liu et al. “Enhancing trust and privacy in distributed networks: a comprehensive survey on blockchain-based federated learning”. In: Knowledge and Information Systems 66.8 (2024), pp. 4377–4403.

[165] Jiao Liu et al. “DefendFL: A privacy-preserving federated learning scheme against poisoning attacks”. In: IEEE Transactions on Neural Networks and Learning Systems (2024).

[166] Yijia Liu et al. “A survey on blockchain-based trust management for Internet of Things”. In: IEEE internet of Things Journal 10.7 (2023), pp. 5898–5922.

[167] Yuan Liu et al. “A semi-centralized trust management model based on blockchain for data exchange in iot system”. In: IEEE Transactions on Services Computing 16.2 (2022), pp. 858–871.

[168] Zelei Liu et al. “Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning”. In: ACM Transactions on intelligent Systems and Technology (TIST) 13.4 (2022), pp. 1–21.

[169] Ángel Longueira-Romerc et al. “How to quantify the security level of embedded systems? a taxonomy of security metrics”. In: 2020 IEEE 18th International Conference on Industrial Informatics (INDIN). Vol. 1. IEEE. 2020, pp. 153–158.

[170] Shanhe Lou et al. “Human-cyber-physical system for industry 5.0: A review from a human-centric perspective”. In: IEEE Transactions on Automation Science and Engineering 22 (2024), pp. 494–511.

[171] Zhirong Luan et al. “Robust federated learning: Maximum correntropy aggregation against byzantine attacks”. In: IEEE Transactions on Neural Networks and Learning Systems (2024).

[172] Xiaoting Lyu et al. “Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning”. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37. 7. 2023, pp. 9020–9028.

[173] Wolfgang Mahnke, Stefan-Helmut Leitner, and Matthias Damm. OPC Unified Architecture. 1st ed. Springer-Verlag Berlin Heidelberg, 2009. isbn: 978-3-540-68898-3.

[174] Pratyusa K Manadhata and Jeannette M Wing. “An attack surface metric”. In: IEEE Transactions on Software Engineering 37.3 (2010), pp. 371–386.

[175] Kent Marett et al. “Beware the dark side: Cultural preferences for lying online”. In: Computers in Human Behavior 75 (2017), pp. 834–844.

[176] Ismael Martinez, Sreya Francis, and Abdelhakim Senhaji Hafid. “Record and reward federated learning contributions with blockchain”. In: 2019 International conference on cyber-enabled distributed computing and knowledge discovery (CyberC). IEEE. 2019, pp. 50–57.

[177] Stephanos Matsumoto and Raphael M Reischuk. “IKP: Turning a PKI around with decentralized automated incentives”. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE. 2017, pp. 410–426.

[178] Roger C Mayer, James H Davis, and F David Schoorman. “An integrative model of organizational trust”. In: Academy of management review 20.3 (1995), pp. 709–734.

[179] Fateme Mazloomi, Shahram Shah Heydari, and Khalil El-Khatib. "Trust-based Knowledge Sharing Among Federated Learning Servers in Vehicular Edge Computing". In: Proceedings of the Int'l ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications. 2023, pp. 9–15.

[180] Carlo Mazzocca et al. "Enabling federated learning at the edge through the iota tangle". In: Future Generation Computer Systems 152 (2024), pp. 17–29.

[181] Agata McCormac et al. "Individual differences and information security awareness". In: Computers in Human Behavior 69 (2017), pp. 151–156.

[182] Tanya McGill and Nik Thompson. "Gender differences in information security perceptions and behaviour". In: Australasian Conference on Information Systems 2018. 2018.

[183] Nick McKeown et al. "OpenFlow: enabling innovation in campus networks". In: ACM SIGCOMM computer communication review 38.2 (2008), pp. 69–74.

[184] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: Artificial intelligence and statistics. PMLR. 2017, pp. 1273–1282.

[185] Vaibhav Mehta et al. "Ranking attack graphs". In: International Workshop on Recent Advances in Intrusion Detection. Springer. 2006, pp. 127–144.

[186] David Meier et al. "Secure Provisioning of OPC UA Applications Using the Asset Administration Shell". In: 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA). IEEE. 2022, pp. 144–149.

[187] Andrea Melis et al. "P-scor: Integration of constraint programming orchestration and programmable data plane". In: IEEE Transactions on Network and Service Management 18.1 (2020), pp. 402–414.

[188] Andrew Meneely et al. "When a patch goes bad: Exploring the properties of vulnerability-contributing commits". In: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE. 2013, pp. 65–74.

[189] Thomas O Meservy et al. “Deception detection through automatic, unobtrusive analysis of nonverbal behavior”. In: IEEE intelligent systems 20.5 (2005), pp. 36–43.

[190] Shweta Mittal and P Vigneswara Ilavarasan. “Demographic factors in cyber security: an empirical study”. In: Conference on e-Business, e-Services and e-Society. Springer. 2019, pp. 667–676.

[191] Fan Mo et al. “PPFL: Privacy-preserving federated learning with trusted execution environments”. In: Proceedings of the 19th annual international conference on mobile systems, applications, and services. 2021, pp. 94–108.

[192] Galamo F Monkam, Jie Yan, and Nathaniel D Bastian. “A Forensic Analysis Framework for Machine Learning Model Poisoning Detection”. In: Security and Privacy 8.5 (2025), e70079.

[193] Benjamin Morrison, Lynne Coventry, and Pam Briggs. “How do Older Adults feel about engaging with Cyber-Security?” In: Human behavior and emerging technologies 3.5 (2021), pp. 1033–1049.

[194] Patrick Morrison, David Moye, and Laurie Ann Williams. Mapping the field of software security metrics. Tech. rep. North Carolina State University. Dept. of Computer Science, 2014.

[195] Patrick Morrison et al. “Mapping the field of software life cycle security metrics”. In: Information and Software Technology 102 (2018), pp. 146–159.

[196] Xenia Mountrouidou et al. “Securing the human: a review of literature on broadening diversity in cybersecurity education”. In: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (2019), pp. 157–176.

[197] Carlos Murguia et al. “Security metrics and synthesis of secure control systems”. In: Automatica 115 (2020), p. 108757.

[198] Mathis Obadia et al. “A greedy approach for minimizing SDN control overhead”. In: Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft). IEEE. 2015, pp. 1–5.

[199] Daniela Oliveira et al. “Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing”. In: Proceedings of the 2017 chi conference on human factors in computing systems. 2017, pp. 6412–6424.

[200] OPC Foundation. OPC Unified Architecture Specification Part 1: Overview and Concepts. [Online]. 2022. url: https://reference.opcfoundation.org/Core/Part1/v105/docs/.

[201] OPC Foundation. OPC Unified Architecture Specification Part 12: Discovery and Global Services. [Online]. 2022. url: https://reference.opcfoundation.org/GDS/docs/.

[202] OPC Foundation. OPC Unified Architecture Specification Part 21: Device Onboarding. [Online]. 2024. url: https://reference.opcfoundation.org/Onboarding/v105/docs/.

[203] OPC Foundation. OPC Unified Architecture Specification Part 4: Services. [Online]. 2024. url: https://reference.opcfoundation.org/Core/Part4/v105/docs/.

[204] OPC Foundation. OPC Unified Architecture Specification Part 5: Information Model. [Online]. 2023. url: https://reference.opcfoundation.org/Core/Part5/v105/docs/.

[205] OPC Foundation. OPC Unified Architecture Specification Part 6: Mappings. [Online]. 2023. url: https://reference.opcfoundation.org/Core/Part6/v105/docs/.

[206] OPC UA Security Working Group. Practical Security Recommendations for building OPC UA Applications. Whitepaper version 3. https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Security-Advise-EN.pdf. OPC Foundation, June 2018.

[207] Xinming Ou, Wayne F Boyer, and Miles A McQueen. "A scalable approach to attack graph generation". In: Proceedings of the 13th ACM conference on Computer and communications security. 2006, pp. 336–345.

[208] Xinming Ou, Sudhakar Govindavajhala, Andrew W Appel, et al. "MulVAL: A logic-based network security analyzer." In: USENIX security symposium. Vol. 8. Baltimore, MD. 2005, pp. 113–128.

[209] Sharon Oviatt, Rachel Coulston, and Rebecca Lunsford. "When do we interact multimodally? Cognitive load and multimodal communication patterns". In: Proceedings of the 6th international conference on Multimodal interfaces. 2004, pp. 129–136.

[210] P4.org. Intel's Tofino P4 Software is Now Open Source. https://p4.org/intels-tofino-p4-software-is-now-open-source/. Accessed: 22 January 2025. 2025.

[211] Keshnee Padayachee. “An assessment of opportunity-reducing techniques in information security: An insider threat perspective”. In: Decision Support Systems 92 (2016), pp. 47–56.

[212] Matthew J Page et al. “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews”. In: bmj 372 (2021).

[213] Jinie Pak and Lina Zhou. “Social structural behavior of deception in computer-mediated communication”. In: Decision Support Systems 63 (2014), pp. 95–103.

[214] Joseph Pamula et al. “A weakest-adversary security metric for network configuration security analysis”. In: Proceedings of the 2nd ACM workshop on Quality of protection. 2006, pp. 31–38.

[215] Sharnil Pandya et al. “Federated learning for smart cities: A comprehensive survey”. In: Sustainable Energy Technologies and Assessments 55 (2023), p. 102987.

[216] Alexander Papageorgiou et al. “DPKI: a blockchain-based decentralized public key infrastructure system”. In: 2020 Global Internet of Things Summit (GIoTS). IEEE. 2020, pp. 1–5.

[217] Christos Patsonakis et al. “Towards a smart contract-based, decentralized, public-key infrastructure”. In: Cryptology and Network Security: 16th International Conference, Hong Kong, China, Springer. 2018, pp. 299–321.

[218] Marcus Pendleton et al. “A survey on systems security metrics”. In: ACM Computing Surveys (CSUR) 49.4 (2016), pp. 1–35.

[219] Eleni Philippou, Sylvain Frey, and Awais Rashid. “Contextualising and aligning security metrics and business objectives: A GQM-based methodology”. In: Computers & Security 88 (2020), p. 101634.

[220] Ryan Pickren et al. “Compromising industrial processes using web-based programmable logic controller malware”. In: Network and Distributed System Security (NDSS) Symposium. 2024.

[221] Nayot Poolsappasit, Rinku Dewri, and Indrajit Ray. “Dynamic security risk management using bayesian attack graphs”. In: IEEE Transactions on Dependable and Secure Computing 9.1 (2011), pp. 61–74.

[222] Serguei Popov. “The Tangle”. In: (Apr. 2018).

[223] Serguei Popov. “The tangle”. In: White paper 1.3 (2018), p. 30.

[224] Clay Posey, Tom L Roberts, and Paul Benjamin Lowry. “The impact of organizational commitment on insiders’ motivation to protect organizational information assets”. In: Journal of management information systems 32.4 (2015), pp. 179–214.

[225] Jeffrey G Proudfoot et al. “More than meets the eye: How oculometric behaviors evolve over the course of automated deception detection interactions”. In: Journal of Management Information Systems 33.2 (2016), pp. 332–360.

[226] Jeffrey Gainer Proudfoot, Randall Boyle, and Ryan M Schuetzler. “Man vs. machine: Investigating the effects of adversarial system use on end-user behavior in automated deception detection interviews”. In: Decision Support Systems 85 (2016), pp. 23–33.

[227] Maxime Puys, Marie-Laure Potet, and Pascal Lafourcade. “Formal analysis of security properties on the OPC-UA SCADA protocol”. In: International Conference on Computer Safety, Reliability, and Security. Springer. 2016, pp. 67–75.

[228] Tiantian Qin, Judee Burgoon, and Jay F Nunamaker. “An exploratory study on promising cues in deception detection and application of decision tree”. In: 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the. IEEE. 2004, pp. 23–32.

[229] Jamshir Qureshi and Samina Khan. “Deciphering deception–the impact of AI deepfakes on human cognition and emotion”. In: (2024).

[230] Alex Ramos et al. “Model-based quantitative network security metrics: A survey”. In: IEEE Communications Surveys & Tutorials 19.4 (2017), pp. 2704–2734.

[231] Vasudevi Reddy. “Getting back to the rough ground: deception and ‘social living’”. In: Philosophical Transactions of the Royal Society B: Biological Sciences 362.1480 (2007), pp. 621–637.

[232] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. “How i learned to be secure: a census-representative survey of security advice sources and behavior”. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016, pp. 666–677.

[233] Muhammad Habib ur Rehman et al. “Towards blockchain-based reputation-aware federated learning”. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE. 2020, pp. 183–188.

[234] Ricardo Alexandre Bentes Ribeiro. “Improving social engineering resilience in enterprises”. MA thesis. Universidade Aberta (Portugal), 2023.

[235] Sara Ricci et al. “Understanding cybersecurity education gaps in Europe”. In: IEEE Transactions on Education 67.2 (2024), pp. 190–201.

[236] Lorenzo Rinieri et al. “In-Network Encryption for Secure Industrial Control Systems Communications”. In: 2024 IEEE 10th International Conference on Network Softwarization (NetSoft). IEEE. 2024, pp. 190–194.

[237] Gaith Rjoub et al. “Trust-augmented deep reinforcement learning for federated learning client selection”. In: Information Systems Frontiers 26.4 (2024), pp. 1261–1278.

[238] Kevin A Roundy and Barton P Miller. “Binary-code obfuscations in prevalent packer tools”. In: ACM Computing Surveys (CSUR) 46.1 (2013), pp. 1–32.

[239] Victoria L Rubin and Tatiana Lukoianova. “Truth and deception at the rhetorical structure level”. In: Journal of the Association for Information Science and Technology 66.5 (2015), pp. 905–917.

[240] Amir Al Sadi et al. “Real-time Pipeline Reconfiguration of P4 Programmable Switches to Efficiently Detect and Mitigate DDoS Attacks”. In: 2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). 2023, pp. 21–23. doi: 10.1109/ICIN56760.2023.10073501.

[241] Fai Ben Salamah et al. “Evaluating the Risks of Human Factors Associated with Social Media Cybersecurity Threats”. In: International Symposium on Human Aspects of Information Security and Assurance. Springer. 2023, pp. 349–363.

[242] Pedro Miguel Sánchez Sánchez et al. “Federatedtrust: A solution for trustworthy federated learning”. In: Future Generation Computer Systems 152 (2024), pp. 83–98.

[243] Puspita Kencana Sari, Putu Wuri Handayani, and Achmad Nizar Hidayanto. “Demographic comparison of information security behavior toward health information system protection: Survey study”. In: JMIR Formative Research 7.1 (2023), e49439.

[244] Reijo Savola. “Towards a security metrics taxonomy for the information and communication technology industry”. In: International Conference on Software Engineering Advances (ICSEA 2007). IEEE. 2007, pp. 60–60.

[245] Reijo M Savola. “Quality of security metrics and measurements”. In: Computers & Security 37 (2013), pp. 78–90.

[246] Amber N Schroeder and Jacqulyn M Cavanaugh. “Fake it’til you make it: Examining faking ability on social media pages”. In: Computers in Human Behavior 84 (2018), pp. 29–35.

[247] Ryan M Schuetzler, G Mark Grimes, and Justin Scott Giboney. “The effect of conversational agent skill on user behavior during deception”. In: Computers in Human Behavior 97 (2019), pp. 250–259.

[248] Claude Elwood Shannon. “A mathematical theory of communication”. In: The Bell system technical journal 27.3 (1948), pp. 379–423.

[249] Dilli P Sharma et al. “Dynamic security metrics for software-defined network-based moving target defense”. In: Journal of Network and Computer Applications 170 (2020), p. 102805.

[250] Andrew R Short, Helen C Leligou, and Efstathios Theocharis. “Execution of a Federated Learning process within a smart contract”. In: 2021 IEEE International Conference on Consumer Electronics (ICCE). IEEE. 2021, pp. 1–4.

[251] Adam Shostack. Threat modeling: Designing for security. John Wiley & Sons, 2014.

[252] Michael Siering, Jascha-Alexander Koch, and Amit V Deokar. “Detecting fraudulent behavior on crowdfunding platforms: The role of linguistic and content-based cues in static and dynamic contexts”. In: Journal of Management Information Systems 33.2 (2016), pp. 421–455.

[253] Vivek K Singh, Isha Ghosh, and Darshan Sonagara. “Detecting fake news stories via multimodal analysis”. In: Journal of the Association for Information Science and Technology 72.1 (2021), pp. 3–17.

[254] Ankush Singla and Elisa Bertino. “Blockchain-based PKI solutions for IoT”. In: 2018 IEEE 4th international conference on collaboration and internet computing (CIC). IEEE. 2018, pp. 9–15.

[255] Arunan Sivanathan, Hassan Habibi Gharakheili, and Vijay Sivaraman. “Managing IoT cyber-security using programmable telemetry and machine learning”. In: IEEE Transactions on Network and Service Management 17.1 (2020), pp. 60–74.

[256] Madeline E Smith et al. “Everyday deception or a few prolific liars? The prevalence of lies in text messaging”. In: Computers in Human Behavior 41 (2014), pp. 220–227.

[257] Xingshuo Song et al. “Performance analysis for DAG-based blockchain systems based on the markov process”. In: Journal of Systems Science and Systems Engineering 34.1 (2025), pp. 29–54.

[258] Yuxiao Song et al. “Blockchain Assisted Trust Management for Data-Parallel Distributed Learning”. In: IEEE Transactions on Mobile Computing (2024).

[259] Orly Stan et al. “Extending attack graphs to represent cyber-attacks in communication protocols and modern it networks”. In: IEEE Transactions on Dependable and Secure Computing 19.3 (2020), pp. 1936–1954.

[260] Kevin F Steinmetz, Alexandra Pimentel, and W Richard Goe. “Performing social engineering: A qualitative study of information security deceptions”. In: Computers in Human Behavior 124 (2021), p. 106930.

[261] Khalid Sultan, Abdeslam En-Nouaary, and Abdelwahab Hamou-Lhadj. “Catalog of metrics for assessing security risks of software throughout the software development life cycle”. In: 2008 International Conference on Information Security and Assurance (isa 2008). IEEE. 2008, pp. 461–465.

[262] Aditya Sundararajan, Arif I Sarwat, and Alexander Pons. “A survey on modality characteristics, performance evaluation metrics, and security for traditional and wearable biometric systems”. In: ACM Computing Surveys (CSUR) 52.2 (2019), pp. 1–36.

[263] John Sweller. “Cognitive load during problem solving: Effects on learning”. In: Cognitive science 12.2 (1988), pp. 257–285.

[264] Farnaz Tahmasebian, Jian Lou, and Li Xiong. “Robustfed: a truth inference approach for robust federated learning”. In: Proceedings of the 31st ACM international conference on information & knowledge management. 2022, pp. 1868–1877.

[265] N Tanrıverdi and Bilgin Metin. “Evaluation of IT security perception”. In: Twenty-third Americas Conference on Information Systems. 2017, pp. 10–12.

[266] Asadullah Tariq et al. “Trustworthy federated learning: A survey”. In: arXiv preprint arXiv:2305.11537 (2023).

[267] Andrea Tesei et al. “IOTA-VPKI: A DLT-based and resource efficient vehicular public key infrastructure”. In: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall). IEEE. 2018, pp. 1–6.

[268] Thin Tharaphe Thein, Yoshiaki Shiraishi, and Masakatu Morii. “Personalized federated learning-based intrusion detection system: Poisoning attack and defense”. In: Future Generation Computer Systems 153 (2024), pp. 182–192.

[269] Marianthi Theoharidou et al. “The insider threat to information systems and the effectiveness of ISO17799”. In: Computers & Security 24.6 (2005), pp. 472–484.

[270] Vale Tolpegin et al. “Data Poisoning Attacks Against Federated Learning Systems”. In: Computer Security – ESORICS 2020. Ed. by Liqun Chen et al. Cham: Springer International Publishing, 2020, pp. 480–501. isbn: 978-3-030-58951-6.

[271] Vale Tolpegin et al. “Data poisoning attacks against federated learning systems”. In: Computer security–ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25. Springer. 2020, pp. 480–501.

[272] Mohsen Toorani and Christian Gehrmann. “A decentralized dynamic PKI based on blockchain”. In: Proceedings of the 36th annual ACM symposium on applied computing. 2021, pp. 1646–1655.

[273] Jasper L Tran. “Navigating the Cybersecurity Act of 2015”. In: Chap. L. Rev. 19 (2016), p. 483.

[274] Anand Tripathi et al. “A coordination model for secure collaboration”. In: Process Coordination and Ubiquitous Computing. CRC Press, 2020, pp. 77–95.

[275] John Wilder Tukey et al. Exploratory data analysis. Vol. 2. Springer, 1977.

[276] Melanie Tupper and A Nur Zincir-Heywood. “VEA-bility security metric: A network security analysis tool”. In: 2008 Third International Conference on Availability, Reliability and Security. IEEE. 2008, pp. 950–957.

[277] Nathan W Twyman et al. “A rigidity detection system for automated credibility assessment”. In: Journal of Management Information Systems 31.1 (2014), pp. 173–202.

[278] Nathan W Twyman et al. “Autonomous scientifically controlled screening systems for detecting information purposely concealed by individuals”. In: Journal of Management Information Systems 31.3 (2014), pp. 106–137.

[279] Nathan W Twyman et al. “Robustness of multiple indicators in automated screening systems for deception detection”. In: Journal of Management Information Systems 32.4 (2015), pp. 215–245.

[280] Nathan W Twyman et al. “Too busy to be manipulated: How multitasking with technology improves deception detection in collaborative teamwork”. In: Journal of Management Information Systems 37.2 (2020), pp. 377–395.

[281] Himani Tyagi, Rajendra Kumar, and Santosh Kr Pandey. “A detailed study on trust management techniques for security and privacy in IoT: Challenges, trends, and research directions”. In: High-Confidence Computing 3.2 (2023), p. 100127.

[282] Estee Van der Walt, Jan HP Eloff, and Jacomine Grobler. “Cybersecurity: Identity deception detection on social media platforms”. In: Computers & Security 78 (2018), pp. 76–89.

[283] Carlos Villarrubia, Eduardo Fernández-Medina, and Mario Piattini. “Towards a Classification of Security Metrics.” In: WOSIS. 2004, pp. 342–350.

[284] Simon Vrhovec, Igor Bernik, and Blaž Markelj. “Explaining information seeking intentions: Insights from a Slovenian social engineering awareness campaign”. In: Computers & Security 125 (2023), p. 103038.

[285] Liang Wang, Yilin Li, and Lina Zuo. “Trust management for IoT devices based on federated learning and blockchain”. In: The Journal of Supercomputing 81.1 (2025), p. 232.

[286] Lingyu Wang, Sushil Jajodia, and Anoop Singhal. Network Security Metrics. Springer, 2017.

[287] Lingyu Wang, Anoop Singhal, and Sushil Jajodia. “Toward measuring network security using attack graphs”. In: Proceedings of the 2007 ACM workshop on Quality of protection. 2007, pp. 49–54.

[288] Lingyu Wang et al. “k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities”. In: IEEE Transactions on Dependable and Secure Computing 11.1 (2013), pp. 30–44.

[289] Lingyu Wang et al. “k-zero day safety: Measuring the security risk of networks against unknown attacks”. In: Computer Security–ESORICS 2010: 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings 15. Springer. 2010, pp. 573–587.

[290] Ning Wang et al. “Flare: defending federated learning against model poisoning attacks via latent space representations”. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. 2022, pp. 946–958.

[291] Shangping Wang et al. “DAG blockchain-based lightweight authentication and authorization scheme for IoT devices”. In: Journal of information security and applications 66 (2022), p. 103134.

[292] Yuwei Wang and Burak Kantarci. “Reputation-enabled federated learning model aggregation in mobile platforms”. In: ICC 2021-IEEE International Conference on Communications. IEEE. 2021, pp. 1–6.

[293] Ze Wang et al. “Blockchain-based certificate transparency and revocation transparency”. In: IEEE Transactions on Dependable and Secure Computing 19.1 (2020), pp. 681–697.

[294] Kang Wei et al. “Covert model poisoning against federated learning: Algorithm design and optimization”. In: IEEE Transactions on Dependable and Secure Computing (2023).

[295] Miranda Wei et al. “{SoK}(or {SoLK?)}: On the Quantitative Study of Sociodemographic Factors and Computer Security Behaviors”. In: 33rd USENIX Security Symposium (USENIX Security 24). 2024, pp. 7011–7030.

[296] Miranda Wei et al. “Skilled or Gulliblef Gender Stereotypes Related to Computer Security and Privacy”. In: 2023 IEEE Symposium on Security and Privacy (SP). IEEE. 2023, pp. 2050–2067.

[297] Theodore J Williams. “The Purdue enterprise reference architecture”. In: Computers in industry 24.2-3 (1994), pp. 141–158.

[298] Bo Xiao and Izak Benbasat. “Product-related deception in e-commerce: A theoretical perspective”. In: Mis Quarterly (2011), pp. 169–195.

[299] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: arXiv preprint arXiv:1708.07747 (2017).

[300] Junfeng Xie et al. “A survey on the scalability of blockchain systems”. In: IEEE network 33.5 (2019), pp. 166–173.

[301] Tianhao Xu, Kuldeep Singh, and Prashanth Rajivan. “Personalized persuasion: Quantifying susceptibility to information exploitation in spear-phishing attacks”. In: Applied Ergonomics 108 (2023), p. 103908.

[302] Amrendra Singh Yadav, Nikita Singh, and Dharmender Singh Kushwaha. “Evolution of Blockchain and consensus mechanisms & its real-world applications”. In: Multimedia Tools and Applications 82.22 (2023), pp. 34363–34408.

[303] Alexander Yakubov et al. “A blockchain-based PKI management framework”. In: The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain 23-27 April 2018. 2018.

[304] Duygu Nur Yaldiz, Tuo Zhang, and Salman Avestimehr. “Secure federated learning against model poisoning attacks via client filtering”. In: arXiv preprint arXiv:2304.00160 (2023).

[305] Di Yang et al. “A review on scalability of blockchain”. In: Proceedings of the 2020 2nd International Conference on Blockchain Technology. 2020, pp. 1–6.

[306] Weiwei Yang et al. “Trusted Mobile Edge Computing: DAG Blockchain-aided Trust Management and Resource Allocation”. In: IEEE Transactions on Wireless Communications (2023).

[307] Zhigang Yang et al. “Blockchain-enabled trust management model for the Internet of Vehicles”. In: IEEE Internet of Things Journal 10.14 (2021), pp. 12044–12054.

[308] George OM Yee. “Designing sound security metrics”. In: International Journal of Systems and Software Security and Protection (IJSSSP) 10.1 (2019), pp. 1–21.

[309] George OM Yee. “Improving the Derivation of Sound Security Metrics”. In: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE. 2022, pp. 1804–1809.

[310] Kimchai Yeow et al. “Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues”. In: IEEE Access 6 (2017), pp. 1513–1524.

[311] Beytullah Yigit et al. “Secured communication channels in software-defined networks”. In: IEEE Communications Magazine 57.10 (2019), pp. 63–69.

[312] Narges Yousefnezhad et al. “A comprehensive security architecture for information management throughout the lifecycle of IoT products”. In: Sensors 23.6 (2023), p. 3236.

[313] Manuel Zander, Tom Waite, and Dominik Harz. “DAGsim: Simulation of DAG-based distributed ledger protocols”. In: ACM SIGMETRICS Performance Evaluation Review 46.3 (2019), pp. 118–121.

[314] Hewa Majeed Zangana. “Human Factors in Cybersecurity: The Weakest Link”. In: Defense in Depth: Modern Cybersecurity Strategies and Evolving Threats (2025), pp. 127–152.

[315] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. “A characterization of cybersecurity posture from network telescope data”. In: Trusted Systems: 6th International Conference, INTRUST 2014, Beijing, China, December 16-17, 2014, Revised Selected Papers 6. Springer. 2015, pp. 105–126.

[316] Fan Zhang et al. “Federated learning meets blockchain: State channel-based distributed data-sharing trust supervision mechanism”. In: IEEE Internet of Things Journal 10.14 (2021), pp. 12066–12076.

[317] Jing Zhang et al. “On the Mismanagement and Maliciousness of Networks.” In: NDSS. Vol. 14. 2014, pp. 23–26.

[318] Kunwu Zhang et al. “Advancements in industrial cyber-physical systems: an overview and perspectives”. In: IEEE Transactions on Industrial Informatics (2022).

[319] Mengyuan Zhang et al. “Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks”. In: IEEE Transactions on Information Forensics and Security 11.5 (2016), pp. 1071–1086.

[320] Qinnan Zhang et al. “Blockchain empowered reliable federated learning by worker selection: A trustworthy reputation evaluation method”. In: 2021 IEEE wireless communications and networking conference workshops (WCNCW). IEEE. 2021, pp. 1–6.

[321] Yingrui Zhang and Osman Yağan. “Robustness of interdependent cyber-physical systems against cascading failures”. In: IEEE Transactions on Automatic Control 65.2 (2019), pp. 711–726.

[322] Zaixi Zhang et al. “Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients”. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. 2022, pp. 2545–2555.

[323] Jie Zheng et al. “Trust Management of Tiny Federated Learning in Internet of Unmanned Aerial Vehicles”. In: IEEE Internet of Things Journal (2024).

[324] Wei Zheng, Yang Cao, and Haining Tan. “Secure sharing of industrial IoT data based on distributed trust management and trusted execution environments: a federated learning approach”. In: Neural Computing and Applications 35.29 (2023), pp. 21499–21509.

[325] Zibin Zheng et al. “Blockchain challenges and opportunities: A survey”. In: International journal of web and grid services 14.4 (2018), pp. 352–375.

[326] Lina Zhou et al. “Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications”. In: Group decision and negotiation 13.1 (2004), pp. 81–106.

[327] Lina Zhou et al. “Language dominance in interpersonal deception in computer-mediated communication”. In: Computers in Human Behavior 20.3 (2004), pp. 381–402.

[328] Hai Zhu et al. “Zero Trust Consumer IoT With Robust Federated Learning Over Main-Side Blockchain”. In: IEEE Transactions on Consumer Electronics (2024).

[329] Ioannis Zografopoulos et al. “Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies”. In: IEEe Access 9 (2021), pp. 29775–29818.

[330] Moti Zwilling et al. “Cyber security awareness, knowledge and behavior: A comparative study”. In: Journal of Computer Information Systems 62.1 (2022), pp. 82–97.