International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

Deterministic Software-Defined Micro-Segmentation and Cryptographic Continuous Authentication in Zero Trust Networks: A Literature Review

Kshitij Singh Bhati[1], Mr. Nitin Phulwani[2], Mr. Amit Kumar[3]

[1]Department of Computer Engineering, Poornima Institute of Engineering & Technology, Jaipur, India
[2]Asst. Professor Department of Computer Engineering, Poornima Institute of Engineering & Technology, Jaipur, India
[3]Assistant Professor & Deputy Head of Department (Administration), Department of Computer Engineering Poornima Institute of Engineering & Technology, Jaipur

---

Abstract - *Perimeter-Based Network Security assumes that everyone who is on the network is trusted, but because of remote working, cloud computing, and Internet of Things deployments, this assumption becomes tough. In Zero Trust Architecture, there is no inherent trust, but instead there is verification on a per-request basis. The current literature review will analyze literature published between 2022 and 2025 concerning the basics of ZTA, the means of authentication used in ZTA, access control, micro segmentation, and SDP solutions, and security automation techniques used in contemporary ZTA. Special focus is made on the deterministic policy server-based architectures combining centralized Policy Decision Points with continuous mutual TLS, SDN-based micro segmentation, and hardware-backed trust. The application of zero-trust architecture principles in 6G networks is also analyzed; in the context of 6G networks, the density of devices, AI-based network management, and open radio access networks raise a new set of challenges for security. Using the results of five relevant literature surveys and design papers, this literature review gives an overview of current developments in a comparative manner and pinpoints the common recurring problems.*

Index Terms -- Zero Trust Architecture, continuous authentication, access control, micro-segmentation, Software-Defined Perimeter, 6G security.

I. INTRODUCTION

Enterprise security has been traditionally based on a "castle-and-moat" approach – firewalls and other gateway devices have been used to build a robust perimeter, which separated the trusted internal environment from the external untrusted one. Everything behind this perimeter could be considered trusted by default. However, remote work and cloud-based services have broken the idea of any reliable network boundary. It is also evident that the castle-and-moat paradigm suffers from several major vulnerabilities: it relies on the existence of a single point of failure that breaks the whole network upon compromise; it operates with a static set of rules that cannot be adapted to the user/device mobility; it lacks continuous identity/device validation after the authentication step; it makes the implementation of the least privilege policy difficult; and it is practically incapable of restricting the lateral movements in case of successful penetration of the perimeter [2].

The Zero Trust concept has been developed as an answer to these challenges. The term has been introduced in 2010 by John Kindervag from Forrester and described as the need to embed security into the network architecture itself instead of creating barriers around it [7]. Later, the NIST Special Publication 800-207 defined the zero trust paradigm as the framework of principles that provide the implementation of the least privilege access policies for every request, assuming that the network has been already compromised [6]. According to subsequent studies, the Zero Trust Architecture is not a ready-to-use solution but a design philosophy, which has to be implemented at the levels of identity, device, network, application, and data [1].

This paper does not introduce a new architecture. Rather, it offers a survey and synthesis of the most recent research on zero-trust realization, based on five works published between 2022 and 2025 [1]-[5]: a survey of ZTA techniques in a tenet-by-tenet approach [1]; a survey on implementing zero-trust concepts in 6G networks [2]; a design paper on the implementation of the zero-trust deterministic policy-server model relying on Single Package Authorization [3]; a paper on automation of micro-segmentation with open-source SDN tools [4]; and a paper on zero-trust hardening of open multi-vendor O-RAN interfaces [5]. Altogether, they cover the entire ZTA stack, from tenets to deterministics to 6G.

The rest of this paper is organized as follows. Section II reviews the foundations of ZTA, including tenets and logical components. Section III discusses authentication mechanisms, from traditional credentials to continuous and cryptographic authentication. Section IV discusses access control models. Section V describes micro-segmentation and Software-Defined Perimeter approaches, including deterministic SDN approaches. Section VI discusses security automation and trust computation. Section VII discusses zero

trust in 6G. Section VIII synthesizes the reviewed literature comparatively. Section IX gathers open research questions, while Section X concludes.

II. FOUNDATIONS OF ZERO TRUST ARCHITECTURE

Zero trust is defined by NIST as the shift from implicitly trusting anything in a network (the "trust but verify" attitude) to continuously evaluating access requests for every resource regardless of their origin in terms of identity, application, and device state [1], [6]. NIST does not offer any particular technology but rather lists seven tenets of a well-established ZTA: any data source and computing service is considered a resource; all communications must be secured regardless of their location; every access is granted per session; access decisions depend on the dynamic policy which takes into account the state of the requestor identity, application, and device; the organization tracks and maintains the security state of all owned and associated devices; all authentication and authorization are performed dynamically with strict enforcement before allowing the access; the organization collects telemetry on network state and communications to enhance its security posture [1].

Zero Trust eXtended ecosystem proposed by Forrester addresses the same issues with different terminology, concentrating on protection of users, devices, networks, workloads, and data through continuous analysis and automation [1], [7]. In other words, both approaches focus on the same issues: resource identification, encrypted communication, session-scoped access, dynamic and context-aware policies, device health, continuous authentication, and logging.

ZTA is implemented using three core logical components. The Policy Engine calculates an access decision according to the trust algorithm taking into account the policy database, user identity and attributes, behavioral history, and threat intelligence. The Policy Administrator accepts this decision and configures the Policy Enforcement Points accordingly, granting, denying, or terminating connections. The Policy Enforcement Point is placed between the requesting subject and the resource, consisting of client-side agent and the resource-side gateway; whatever is beyond it is considered the trust zone [1]. They are summarized in Table I.

TABLE I. CORE LOGICAL COMPONENTS OF ZERO TRUST ARCHITECTURE [1]

Component	Function
Policy Engine (PE)	Computes the access decision via a trust algorithm fed by policy, identity, behavioral, and threat-intelligence data.
Policy Administrator (PA)	Receives the PE's decision and instructs Policy Enforcement Points to permit, deny, or terminate the session.
Policy Enforcement Point (PEP)	Client- and gateway-side component that establishes, monitors, and terminates the connection to the resource.

The trust algorithm can be criteria-based, requiring a fixed combination of attributes before granting access, or score-based, where weighted inputs produce a confidence value compared against a threshold. It can also be singular, ignoring the requester's history, or contextual, weighing prior behavior to flag deviations from normal usage. NIST guidance favors the contextual, score-based approach for its ability to detect subtler anomalies [1]. As later sections show, this adaptive philosophy sits at odds with the fully deterministic, rule-matrix approach favored by some recent design proposals.

Beyond these logical components, the surveyed literature describes three broad ways an organization can realize a working ZTA, each defined by which data sources principally drive the Policy Engine. An identity-governance-centric approach treats the validity of user and device identities, together with contextual information and asset sensitivity, as the primary basis for access decisions. A micro-segmentation-centric approach places enforcement points as close as possible to each protected resource, using virtual firewalls to block unauthorized lateral movement. A network-infrastructure approach builds an overlay network a Software-Defined Perimeter that hides resources from any client not yet authenticated through the overlay's own controller [1]. These three approaches are not mutually exclusive. Sections IV and V examine the access-control and micro-segmentation techniques that realize them in more detail.

III. AUTHENTICATION SCHEMES FOR ZERO TRUST NETWORKS

A. Traditional Authentication and Context-Aware Authentication

Password authentication is the most used scheme and is also the most fragile one. It uses password reuse, which is prone to dictionary attacks; it uses low-quality passwords; it may be subject to side-channel attacks. The use of physical biometrics does not make authentication better since fingerprints can be lifted and recreated, facial recognition can be spoofed using a picture or a 3D printed model, while iris scanners are prone to spoofing using a picture of the iris implanted into a contact lens [1]. While MFA decreases the risks brought by one-factor authentication, it creates friction – it may use hardware tokens, which are cumbersome to use, SMS delivered one-time code, which is vulnerable to interception, and usually it uses another device that becomes unusable in case of loss and in case its battery dies [1].

Context-aware authentication includes additional contextual information in the process of deciding whether to demand stronger evidence or not – geolocation, time of day, proximity to networks, device sensors. Among the schemes analyzed, there are location-and-time based risk scores, as well as multimodal FIDO-based authenticators using PIN, face, and behavioral data. The problem with them all is that some authentication methods can be implemented only on specific types of devices, making them difficult to implement uniformly throughout an enterprise [1].

B. Continuous Authentication and Device Authentication

While traditional authentication is performed once when logging in, continuous authentication can be used to ensure that the same person who logged in remains the owner of a device during the whole session. The techniques analyzed range from behavioral biometrics – typing rhythm, touch dynamic, gait pattern – to passive physiological signals like EEG, ECG, EMG which do not require user action. Each of them can be used only in certain scenarios – gait detection works only when the user is walking, typing rhythm works only when typing is taking place. None of these methods works in all three types of devices used in a company – smartphones, laptops, desktop computers [1].

Devices themselves need to be authenticated independently from the user, especially in cases of machine-to-machine interaction in IoT and OT systems. Physically Unclonable Functions generate a unique cryptographic identifier using the inherent variance of the device hardware in the manufacturing process. They are still vulnerable to machine-learning attacks predicting the challenge-response pair accurately enough to enable cloners to fake them. Other identity signals that may complement PUFs include device connectivity with known gateways, statically assigned identifiers such as serial numbers, and baseline behavior of the device in interacting with nearby devices [1].

C. Deterministic Cryptographic Continuous Authentication

There are recent designs which try to replace the adaptive trust evaluation scheme using risk scoring with the deterministic authentication scheme. One such scheme includes six steps: the client requests access; the policy server verifies credentials, organizational registration, and applicable policy; it grants or denies access; after successful authentication, the access request is transferred to the destination server, which follows the same path in reverse [3].

This same scheme introduces Single Package Authorization alongside "endogenous" security architecture which tries to increase the diversity of the attack surface of the system beyond what can be sustained by a single resource-constrained policy engine [3]. Security of sessions is increased through continuous mutual TLS handshaking – short-lived cryptographic tokens are used to authenticate the ongoing session at deterministic intervals rather than just at the login – thus addressing the risk of session hijacking that cannot be prevented with one-time credentials alone [3]. Device identity is secured with hardware means in the form of Trusted Platform Module, combining software-level authentication with hardware root of trust so that just stealing the credentials would not allow impersonating the device [3].

In comparison to the adaptive trust evaluation schemes in Section II, this deterministic scheme gives up the flexibility of risk scoring in favor of predictability and auditability. Instead of performing the access decision according to the statistical probability measure, every decision in the system follows an explicit rule [3].

D. Authentication Standards and Assurance Levels

The standardization organizations specify the minimum strength of authentication needed for a particular resource. According to NIST, there are three Authenticator Assurance Levels: AAL1 requires single-factor or multi-factor authentication sufficient to provide basic control over the authenticator; AAL2 requires two separate authentication factors and approved cryptographic algorithms; AAL3 requires hardware-based impersonation-resistant

authenticator along with proof of key possession [1]. There are several interoperable standards implementing these levels: FIDO and FIDO2/WebAuthn for public-key authentication, U2F for hardware security keys, and OAuth 2.0 for token-based authorization [1]. This deterministic design from Section III-C tries to reach AAL3 level by combining hardware-rooted TPM check with continuously issued cryptographic session tokens although the source cited here does not frame it in NIST assurance level terms.

IV. ACCESS CONTROL SCHEMES

Access control specifies what an authenticated user may do. The surveyed literature classifies these schemes into several increasingly expressive models. Identity-Based Access Control assigns the permissions to users directly by their identifier using access control lists – simple but impractical due to lack of scalability, every personnel/resource change requires manual update of the list [1]. Role-Based Access Control assigns the permissions to roles, simplifying the management but suffering from the problem of role explosion due to accumulated exceptions [1]. Attribute-Based Access Control generalizes both of them by evaluating a Boolean function over attributes of the user, object and environment, such as time, location, risk level. It can be implemented either with XACML or with the newer NGAC standard, providing single non-conflicting decision algorithm, better separation between policy logic and environment, and easy per-user auditing [1].

Beyond these three foundational models, Risk-Based Access Control computes an access decision from an estimated risk score, commonly derived through fuzzy inference over the requester's context and history, though it does not yet scale to resource-constrained IoT deployments. The Capability-based access control scheme distributes tokens that carry cryptographically signed privileges of the particular subject but demands each participant device to behave like an enforcement point on its own, which is not practical due to hardware limitations. Usage Control provides the usual models with the mutable attributes that can be changed within a live session, thus revoking permissions during the session in case of a particular condition change; the policies of a smart home that depend on instantaneous reactions to a change in occupancy state provide a motivating example here. Access control via blockchain technology has been discussed to eliminate a single point of failure in a centralized policy administration but requires smart contracts to govern and audit access requests; consensus problems and transparency vs. privacy dilemma have shifted interests towards permissioned blockchains [1].

The deterministic design in Section III-C follows a different path: instead of enriching ABAC and RBAC with additional attributes, it introduces the single Policy Decision Point to evaluate user tokens, device certificates, and the exact resource request according to the pre-defined policy matrix [3]. According to its authors, the ABAC and RBAC approaches are practically unusable in the organizations that scale to thousands of containerized microservices due to rule-rigidity vs. scalability problem that the deterministic policy matrix is meant to resolve [3]. Table II gives the summary of pros and cons of all three schemes.

TABLE II. COMPARISON OF ACCESS CONTROL MODELS [1], [3]

Model	Strength	Limitation
IBAC	Simple to reason about	Does not scale with subject/object churn
RBAC	Centralized role management	Prone to role explosion over time
ABAC / NGAC	Fine-grained, context-aware	Higher policy-authoring complexity
RbAC	Adapts to estimated risk	Costly risk computation at IoT scale
Capability-based	Distributed, cryptographically enforced	Every device must act as a PDP
UCON	Mid-session revocation	Requires an attribute manager per node
Blockchain-based	No single trusted third party	Consensus overhead; privacy trade-off
Deterministic PDP	Predictable, auditable at scale	Less adaptive than risk-based scoring

The difference between using XACML and NGAC to implement ABAC is far from superficial. The use of NGAC in assessing policy decisions is based on a single, linear decision-making algorithm on non-conflicting rules; XACML needs to reconcile multiple, possibly conflicting rule sets via a more involved decision process. Furthermore, NGAC has better separation between the access control decision-making and the environment, natural support for discretionary access control, and allows the examination of composed policies per subject/object [1]. For IoT-centric smart-home deployments, the surveyed literature favors ABAC via NGAC over XACML because of this more flexible policy-management model.

V. MICRO-SEGMENTATION AND SOFTWARE-DEFINED PERIMETER

A. Principles and Deployment Models

Micro-segmentation places security policy as close as possible to the protected resource, breaking a flat network into small logical segments so that a compromised host cannot move laterally toward unrelated resources. Deployment can be native, using the hypervisor or operating system already running the workload; delegated to third-party virtual firewalls between virtualized servers; realized through an overlay of lightweight agents reporting to a central controller; or, most commonly, some hybrid of the three [1]. Network-dependent implementations translate application-level policy into IP-and-port firewall rules simple, but exposed to spoofing. Network-independent alternatives, including transport-level packet tagging, workload-identity labeling, deep packet inspection, and API-aware container segmentation, bind policy to a workload's identity rather than its address, at the cost of greater implementation complexity [1].

B. Software-Defined Perimeter

The Software-Defined Perimeter generalizes micro-segmentation into an overlay network that hides protected resources until a client has been authenticated, inverting the conventional connect-then-authenticate order. An SDP architecture generally implements five layers of security: Single Packet Authorization, where the controller silently rejects any connection attempt unless it follows a valid authorization packet, meaning that reconnaissance tools don't see any possible surface to attack; mutual TLS, which establishes that both client and server must have verifiable identities, rather than just the server; device validation, which ensures that the one presenting the credentials isn't stolen or cloned; firewall rules dynamically established only after a successful authentication, and encrypted tunnels tied to the authorized application [1]. The biggest hurdle in implementing SDP is that it requires significant modifications to client and server code, and the controller itself becomes an appealing target [1].

C. Automated and Deterministic Micro-Segmentation

Curating the micro-segmentation rules manually doesn't scale well in environments where there are many workloads, which is why automation has become a necessity with SDN. In one reviewed example, the automation process uses an open-source SDN controller in order to create policy through infrastructure-as-code templates, and automatically provision isolated micro-perimeters without the need of manual configurations. This gets rid of the proprietary license costs, but still requires careful testing to avoid introducing misconfiguration with template-driven provisioning [4].

The deterministic architecture described above implements this approach using a specific set of tools that was formally verified. Firstly, there is Mininet-based simulation of the topology to measure the network's behavior and throughput, then, there is an OpenDaylight or Ryu OpenFlow controller that provisions isolated VLANs or container subnets from a security playbook hardcoded into it. The Docker containers along with HashiCorp Vault host the microservices of the network, issue and rotate short-lived cryptography keys, and Scyther is the protocol analysis tool used to formally verify that the handshake protocol is resistant to session hijacking attacks [3]. It is a recurring theme in the deterministic literature that automation and formal verification are combined to make a rule-based system workable at scale, and replace configuration reviews manually.

In case of O-RAN deployment, there is a similar hardening problem because of the open interfaces between equipment from different vendors. The reported solutions include mutually authenticated TLS handshake and strict API authorization token at each interface, where each command is executed after a hardware-rooted cryptographic check; the drawback is that the reported cost of such a solution is a big increase in initial handshake connection time [5]. There is very little source material available about this design, so it is provided here at the general level.

VI. SECURITY AUTOMATION AND TRUST COMPUTATION

Since a zero-trust policy engine is supposed to analyze heterogeneous and voluminous telemetry threat intelligence, the state of devices, logs of networks and applications, and continuous diagnostics manual analysis is impractical in an enterprise environment. The research literature focuses extensively on machine learning methods to decrease the volume of data to the point where actionable signals are revealed: there are various studies that use support-vector-machine and deep learning classifiers to identify distributed denial-of-service traffic in SDN-controlled networks and Manufacturer Usage Description profiles to capture the expected network behavior of an IoT device to detect deviations automatically [1]. However, the issue of trust computation remains unsolved outside the ad hoc and peer-to-peer settings, because the algorithms employed in those cases cannot be used in the enterprise and critical infrastructure networks due to much greater heterogeneity of numerical and imprecise contextual information [1].

The learning-oriented method stands in direct contradiction to the deterministic framework introduced in this review, as the proponents of the latter philosophy prefer reliable rule-based enforcement over statistical variations introduced by the machine-learning policy engine and argue that the deterministic structure, which does not contain "the unpredictability or processing anomalies of algorithmic AI additions," is much more reliable in terms of security [3]. The two approaches are not mutually exclusive the deterministic Policy Decision Point could theoretically consume the results produced by the machine-learning algorithm as one of many inputs but the surveyed literature lacks consensus regarding integration of the two approaches. This issue will be discussed in detail in Section IX.

Machine learning-based approaches used in the surveyed literature cover a wide range of algorithms. The support-vector-machine was used to classify denial-of-service traffic on the basis of packet-count and duration. Both convolutional and feed-forward neural networks were employed to detect multiple attacks simultaneously, namely denial-of-service, reconnaissance, and information-theft attacks from general packet-header features. Long short-term memory networks have been reportedly used to detect novel attacks, which did not fit any of the previously identified models, better than traditional classifiers [1]. STIX documents combined with lightweight communication protocols like XMPP were proposed to formalize the process of distributing indicators of compromise in order to trigger an automatic policy change of enforcement tools like SDN controller and DNS sinkholes after detecting a certain threat [1]. In the surveyed literature, however, there is no endorsement of the deterministic approach and its claim about superiority of rule-based over statistical learning approaches.

VII. ZERO TRUST FOR 6G AND NEXT-GENERATION NETWORKS

6G networks, on the other hand, will presumably connect ten times as many devices as 5G networks do, include built-in AI in their network management, and rely on open, multi-vendor interfaces of radio-access systems like O-RAN. Each of these three features broadens the scope of what a security architecture should cover. A recent survey identifies the corresponding issues as belonging to four recurring challenges in 6G security: the size of the network makes it hard for a non-scaling architecture to implement security; the heterogeneity and multi-operation of the network make it difficult to manage policies consistently across operators; the openness of the evolving O-RAN ecosystem adds more interfaces and vendors, as well as more potential points of failures; and finally, machine-to-machine autonomous communication in a humanless environment requires particularly fine-grained access control [2].

Zero trust is offered in this literature as a natural, even if not quite thoroughly validated yet, response. Its underlying assumption about no entities being trusted as such based on their physical network location works well in the context of a network without a fixed perimeter and multiple independently managed segments. 6G-oriented ZTA architecture extends the usual PEP/PDP/PA scheme with an additional metadata plane to provide a policy engine with continuous diagnostics, threat intelligence, and activity logs so that the access decisions could be made with a wider and up-to-date view of the network state than would be needed in a typical enterprise setting [2].

The same survey offers a look at some publicly announced industry practices in adoption of zero trust that, even though they were not developed specifically for 6G, still demonstrate relevant design patterns: Google's BeyondCorp focuses on continuous identity verification of users and devices instead of relying on network locations; Cisco integrates continuous behavioral monitoring with adaptive access controls and uses machine learning; Capital One pays particular attention to dynamic and continuous recalculated risk assessment; Adobe has been implementing a phased transition from traditional to zero-trust-oriented architecture; and Wells Fargo emphasizes data-centric security along with authentication [2]. Different migration strategies have been reported: NIST offers an incremental and piloted transition strategy; Cisco presents a migration across operational, managerial, and strategic perspectives; Google's BeyondCorp migration goes through successive steps in a technical sense, including device registration and network reconfiguration; and Microsoft's strategy highlights alignment of stakeholders across identity, device, access, and services dimensions [2]. None of these migration models has been designed for 6G scale, which the surveyed authors identify as an open problem for future research.

The O-RAN hardening discussed in Section V-C is directly relevant here. As 6G networks rely increasingly on open, disaggregated, multi-vendor radio-access equipment, the trust boundary between vendors that underlies mTLS hardening in the O-RAN design becomes a mainstream concern in 6G security instead of being a specialized edge case in telecommunication [5]. Bridging general 6G security literature and O-RAN equipment-level hardening as described in Section V-C is one of the areas left unexplored in this survey.

TABLE IV. ZTA MIGRATION STRATEGIES REVIEWED IN [2]

Framework	Migration Approach
NIST SP 800-207	Incremental rollout starting from a small pilot group before onboarding further users and assets.
Cisco	Frames migration across operational, managerial, and strategic perspectives, prioritizing device visibility.
Google BeyondCorp	Iterative technical rollout: device enrollment, network reconfiguration, then staged user onboarding.
Microsoft	Stakeholder-driven migration spanning identity, device, access, and service components.

VIII. COMPARATIVE SYNTHESIS OF REVIEWED LITERATURE

The five works reviewed here span the full zero-trust stack without duplicating one another's focus. Reference [1] establishes tenet-level foundations across authentication, access control, encryption, and micro-segmentation; reference [2] extends those foundations to the scale and openness of 6G networks; reference [3] proposes a deterministic, policy-server-centric design built around Single Package Authorization and hardware-rooted trust; reference [4] automates micro-segmentation through open-source SDN tooling; and reference [5] addresses the cross-vendor trust problem specific to open radio-access-network interfaces. The three design-oriented works [3]–[5] share a common direction: each moves away from a single, monolithic policy engine toward distributed, automated, or hardware-anchored trust mechanisms. None of the five, however, reports large-scale, real-world empirical validation of the proposed technique's performance or security under production conditions the most consistent limitation across the surveyed literature. Table III summarizes the five works comparatively.

TABLE III. SUMMARY OF REVIEWED LITERATURE

Ref.	Year	Focus Area	Key Contribution	Principal Limitation
[1]	2022	ZTA foundations (survey)	Tenet-by-tenet review of authentication, access control, encryption, micro-segmentation, and security automation across 180+ works.	Descriptive only; proposes no new architecture or empirical evaluation.
[2]	2024	ZTA for 6G networks (survey)	Maps 6G-specific security challenges to ZTA principles; reviews industry case studies and migration strategies.	Conceptual; no 6G-specific protocol, simulation, or quantitative evaluation.
[3]	2024	Deterministic ZTA design	Proposes Single Package Authorization and an endogenous architecture around a policy server, continuous mTLS, SDN micro-segmentation, and TPM-rooted trust.	Described narratively; no simulation, prototype, or performance results.
[4]	2024	SDN-automated micro-segmentation	Automates micro-perimeter provisioning via open-source SDN controllers and infrastructure-as-code templates.	Requires rigorous testing to avoid template-driven misconfiguration.
[5]	2025	O-RAN interface hardening	Applies mutually authenticated TLS and hardware-rooted checks across open, multi-vendor O-RAN interfaces.	Increases initial handshake connection time; source detail limited.

IX. OPEN RESEARCH CHALLENGES AND FUTURE DIRECTIONS

Several open problems recur across the reviewed literature. First, most authentication and access-control mechanisms in ZTA still depend on conventional public-key cryptography, which is expected to become vulnerable once sufficiently powerful quantum computers exist; post-quantum cryptographic standardization is underway but not yet reflected in the mTLS-heavy designs reviewed here [1]. Second, deterministic policy matrices avoid the administrative complexity of ABAC and RBAC at scale, but neither the deterministic literature nor the conventional access-control literature demonstrates this at the scale of real, production container fleets [3]. Third, continuous re-authentication whether through periodic mTLS handshakes or behavioral biometrics imposes latency and computational cost that has not been measured on resource-constrained edge or IoT hardware, where such overhead may be least affordable [1], [3].

Fourth, trust computation for enterprise and critical-infrastructure networks remains underdeveloped relative to ad hoc and peer-to-peer trust algorithms, particularly when combining heterogeneous numerical and imprecise contextual signals into a single, auditable trust score [1]. Fifth, the field has not reconciled the deterministic, rule-based philosophy examined in Sections III–VI with the adaptive, machine-learning-driven philosophy that dominates conventional ZTA security automation; whether the two should be combined in a single system, and how, remains an open design question. Finally, every design-oriented work reviewed here [3]–[5] lacks large-scale empirical validation, and standardized migration playbooks bridging legacy, cloud-native, and 6G-native environments remain underdeveloped [2]. Addressing these gaps is the most direct path toward maturing zero trust from a design philosophy into a validated engineering discipline.

A more practical concern is that of interoperability. Each of the above-reviewed works on 6G and O-RAN relies on the requirement for multi-vendor equipment to be able to interoperate securely, but none of the above-reviewed articles proposes a common certification or interoperability standard similar to FIDO2 for continuous device authentication based on hardware root-of-trust [2], [5]. While the costs of continuous verification, whether in terms of repeated re-authentication challenges or computation time required for periodic mTLS handshakes, have been considered qualitatively in the literature reviewed here, they have not been quantified, making it difficult for practitioners to quantify the trade-off between the security benefits of more frequent re-authentication and its cost to legitimate users and limited devices [2], [3].

X. CONCLUSION

Zero Trust has evolved from a theoretical reaction to the inadequacy of perimeter security to a technical stack including techniques for authentication, access control, micro-segmentation, and security automation. This paper has followed the evolution of the Zero Trust stack through a set of five recent works – from the NIST-based foundational surveys of zero trust, through deterministic, server-oriented designs relying on SPAR and hardware-rooted trust, to the application of zero trust principles to 6G and open radio access networks. Two notable trends have emerged in current literature – namely, automation-based designs that replace risk-based adaptive mechanisms with deterministic mechanisms and the adaptation of zero-trust principles to large-scale and open networks unforeseen in NIST's original tenets. What the current literature reviewed here is still missing is large-scale empirical validation.

REFERENCES

1) N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "Zero Trust Architecture (ZTA): A Comprehensive Survey," IEEE Access, vol. 10, pp. 57143–57179, 2022.

2) N. Nahar, K. Andersson, O. Schelén, and S. Saguna, "A Survey on Zero Trust Architecture: Applications and Challenges of 6G Networks," IEEE Access, vol. 12, pp. 94753–94764, 2024.

3) A. Gupta, P. Gupta, U. P. Pandey, P. Kushwaha, B. P. Lohani, and K. Bhati, "ZTSA: Zero Trust Security Architecture A Comprehensive Survey," in Proc. 2024 Int. Conf. Communication, Computer Sciences and Engineering (IC3SE), 2024, pp. 378–383.

4) S. Arora et al., "Microsegmented Cloud Network Architecture Using Open-Source Tools," 2024. [Source not independently verified — confirm bibliographic details before final submission.]

5) A. Mehrban et al., "Integrating Zero Trust Architecture in O-RAN," 2025. [Source not independently verified — confirm bibliographic details before final submission.]

6) S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, Gaithersburg, MD, USA, 2019.

7) J. Kindervag, S. Balaouras, and L. Coit, “Build Security Into Your Networks’ DNA: The Zero Trust Network Architecture,” Forrester Research, Cambridge, MA, USA, 2010.

8) P. Phiayura and S. Teerakanok, "A Comprehensive Framework for Migrating to Zero Trust Architecture," IEEE Access, vol. 11, pp. 19487–19511, 2023. DOI: 10.1109/ACCESS.2023.3248622

9) P. Dhiman, N. Saini, Y. Gulzar, S. Turaev, A. Kaur, K. U. Nisa, and Y. Hamid, "A Review and Comparative Analysis of Relevant Approaches of Zero Trust Network Model," Sensors, vol. 24, no. 4, Art. no. 1328, 2024. DOI: 10.3390/s24041328

10) X. Feng and S. Hu, "Cyber-Physical Zero Trust Architecture for Industrial Cyber-Physical Systems," IEEE Transactions on Industrial Cyber-Physical Systems, vol. 1, pp. 394–405, 2023. DOI: 10.1109/TICPS.2023.3253104

11) Y. Ge and Q. Zhu, "GaZeTA: Game-Theoretic Zero-Trust Authentication for Defense Against Lateral Movement in 5G IoT Networks," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 540–554, 2024. DOI: 10.1109/TIFS.2023.3327445

12) L. Meng, D. Huang, J. An, X. Zhou, and F. Lin, "A Continuous Authentication Protocol Without Trust Authority for Zero Trust Architecture," China Communications, vol. 19, no. 8, pp. 198–213, 2022. DOI: 10.23919/JCC.2022.08.015