Hindawi
Security and Communication Networks
Volume 2022, Article ID 3178760, 14 pages
https://doi.org/10.1155/2022/3178760

Research Article

A Maturity Framework for Zero-Trust Security in Multiaccess Edge Computing

Belal Ali,[1] **Simsam Hijjawi**,[2] **Leith H. Campbell**,[1] **Mark A. Gregory**,[1] **and Shuo Li**[1]

[1]*School of Engineering, RMIT University, Melbourne, Australia*
[2]*eSolutions-Cyber Risk and Resilience, Monash University, Melbourne, Australia*

Correspondence should be addressed to Belal Ali; s3775159@student.rmit.edu.au

Received 12 October 2021; Revised 29 April 2022; Accepted 24 May 2022; Published 29 June 2022

Academic Editor: Vijayakumar Pandi

Multiaccess Edge Computing (MEC) has been adopted to provide an environment that supports cloud computing capabilities and IT services at the network edge. The open architecture of cloud computing and network access at the edge provides malicious actors with many attack vectors. The MEC system entities cannot be permanently trusted due to the dynamic and shareable nature of MEC deployments. This paper presents a classification of MEC entities that can be used to define security controls based on the Zero-Trust Security approach. The security controls are organised into a maturity framework that can be used to guide the systematic development of trust and security in an MEC environment. In this framework, a Minimum Viable Security posture defines the first operational step towards full implementation of Zero-Trust Security in an MEC environment.

1. Introduction

Multiaccess Edge Computing (MEC) is described by the European Telecommunications Standards Institute (ETSI) as a new ecosystem and value chain that offers application developers and content providers an environment to support cloud computing capabilities and IT services at the network edge [1]. MEC aims to improve application and service reliability, reduce latency, and enable computation and storage offloading by end-user devices [2]. Reference [2] describes the MEC functional layers and security threats related to edge devices along with access, edge, and core networks.

Several authors [3–7] have proposed a zero-trust approach to security in MEC. Full implementation of Zero-Trust Security (ZTS) would be a major undertaking, and the implementation could be staged to reduce complexity.

Following a long-standing approach used in Enterprise Computing such as Forrester eXtended [8], Gartner's "Continuous Adaptive Risk and Trust Assessment" (CARTA) [9], MITRE's ATT&CK framework [10], NIST 800-207 [11], and NCCoE's zero-trust architecture [12], we propose a maturity framework for implementing ZTS in MEC. The framework can be used to guide developers during implementation. As part of the framework, we identify an initial stage, called Minimal Viable Security (MVS), during which MEC implementation can be protected from major attacks. The framework also describes a Fully Implemented Security (FIS) stage, in which all interactions between entities, that is, User Equipment (UE), device, user, application and service, are checked for trustworthiness before execution. In many cases, this optimal state may not be achieved, but it represents an implementation design goal.

Hardware-based security may also be used to complement software-based security and add efficiency to implementing and managing protections to the computing infrastructure. The Physical Unclonable Function (PUF) has emerged as the security primitive of choice due to its low power, high speed, and resiliency to side-channel, physical, and software attacks, for example, guessing, simulation, emulation, and protocol attacks [13].

This paper proposes a security framework for MEC architecture based on ZTS principles to enable MEC

operators to assess their zero-trust readiness and build a roadmap to adapting the ZTS model. The contributions proposed in this paper include the following:

(i) A review of the ZTS model, including the MEC trusted elements "pillars" and security processes "enablers."

(ii) A new conceptual ZTS model and maturity framework for MEC.

(iii) A practical realisation of the ZTS model by running a UE identity access simulation model.

(iv) A brief discussion on adapting the ZTS standard, adversary modelling, and future research directions.

The rest of this paper is organised as follows. Section 2 discusses the current research into MEC security. Section 3 illustrates the critical features of Zero-Trust Security that are used to construct the framework. Section 4 introduces authentication access for UE, demonstrates the performance of the proposed model, and provides our findings. Section 5 describes the MEC trust pillars and how to secure the operating processes. Section 6 outlines the ZTS approach and introduces our maturity framework, including MVS. Finally, Section 7 provides a summary and conclusion. Table 1 summarises the abbreviations used in this paper.

Table 1: List of main abbreviations.

Abbreviation	Full name
AI	Artificial intelligence
API	Application Programmable Interface
CASB	Cloud access security broker
DLP	Data Loss Prevention
ECC	Elliptic Curve Cryptography
ETDR	Endpoint threat detection and response
FIS	Fully Implemented Security
HTTP	Hypertext transfer protocol
IAM	Identity and Access Management
IdP	Identity Provider
M2M	Machine-to-Machine
MEC	Multiaccess Edge Computing
MFA	Multifactor authentication
MVS	Minimal Viable Security
NAC	Network Access Control
PUF	Physical Unclonable Function
SASE	Secure Access Service Edge
SIEM	Security Information and Event Management
SOA	Service-Oriented Architecture
SOAR	Security Orchestration, Automation, and Response
SSH	Secure shell protocol
SSO	Single Sign-On
UE	User Equipment
UEM	Unified Endpoint Management
VM	Virtual machine
WAF	Web Application Firewall
XDR	eXtended Detection and Response
ZTM	Zero-Trust Maturity
ZTS	Zero-Trust Security

2. Related Work

The security of MEC infrastructure is a topical research area and one of the most critical issues in the MEC environment. Trust plays an essential role in the MEC security architecture. A trust model can provide an approach to deal with the likelihood that security risks increase as edge computing technology becomes more prevalent. Trust models require properties and parameters for effective trust establishment in MEC. Trust customisation and trust aggregation challenges need to be accomplished for trust models. Furthermore, most trust models support subjective trust values, and trust models usually support distributed computation for storing trust information.

Many trust architectures, frameworks, and models have been proposed to develop confidence-based relationships between users and SPs.

In [14], security was assured by performing authentication of the user devices. Initially, the user devices were registered, and they were verified using identity, password, and net address.

The message from the device was hashed and then transferred. A secure, anonymous authentication scheme (S-SAAS) was proposed to authenticate users based on identity and session key [15]. A unique technology card with a built-in chip, that is, smart card, was used for storing the security credentials and required whenever authentication was requested. However, the smart card can be stolen or lost, so it fails in the authentication.

In [16], a lightweight mutual authentication protocol was proposed using Elliptic Curve Cryptography and a one-way hash function. The security credentials used for authentication are identity, pseudoidentity, and token. The ECC operator was involved in generating random values for the token prediction. At each step, the timestamp is verified, and then security credentials are validated. Traditionally, the authentication was handled based on the security credentials of individual users, that is, authentication factor.

A two-factor authentication scheme was proposed [17]. The two factors are user identity and password. However, the password is required to be secure. Hence, hashes are generated from random numbers and passwords using the XOR (eXclusive OR) operator. Apart from the operations and the credentials, the message authentication code was generated and verified before user access. This work also uses a smart card for entry of the security credentials. The password update was also introduced to improve security when the smart card was lost.

Security provisioning was presented in MEC as an enhanced Fine-Grained Access Control mechanism [18]. The authentication was based on attribute matching. The trust relationship between users was determined from direct and indirect interactions that took place. In this work, the users were grouped, and the trust was predicted based on the group. A user group update module was designed to update the user constraints. The threshold for matching degree was validated and allowed access for the user. The user attributes were not specified, which plays a vital role in the authentication. The evaluation of trust results in the improvement of security in MEC.

A multitier MEC structure was proposed in [5] for the implementation of a degree of trust. The degree of trust was determined and categorised into three levels: high, medium, and low trust. A low-security level will perform three security methods: user authentication, edge authentication, and data encryption.

Similarly, based on the security level, the scheme includes isolation and integrity security methods. At the same time, encryption (SHA-256 algorithm) and authentication are set to a default setting, which increases the complexity in a large-scale network.

A ZTS model was designed in [19] to provide security using continuous validation of the device. The continuous process of authentication and authorisation was considered to be essential in ZTS. In [20], a packet-based authentication was proposed with the use of ZTS. A firewall Access Control List (ACL) was created to allow access for authenticated devices. The packet features of each device differ based on their requirement; hence, it could not predict illegitimate devices. A conceptual zero-trust strategy was proposed in [21]. In this work, a trust engine was illustrated based on the structure of the zero-trust triangle between user, device, and application. The engine was enabled to compute trust scores dynamically based on multiple factors. The security factors were not identified.

In [22], a hybrid trust model was proposed using a fuzzy logic method and Q-learning algorithm. The direct and indirect trust values were computed based on interaction and witnesses, respectively. On combining these trust values, the decision was undertaken in accordance with the fuzzification rules. The trust value was determined from the Q-learning algorithm. After computing the result from the reinforcement algorithm, it is possible to make the fuzzy decision. Hence, the delay in decision-making will be primarily due to the sequential process.

A dynamic access control model in zero-trust architecture was designed specifically for the healthcare system [23]. The security model was in four dimensions: subject, object, environment, and behaviour. In this work, zero-trust dynamic access control was created, and it is composed of two engines and an access agent. Identity-based authentication takes place to differentiate the registered devices from fake user devices. Identity forging can be possible, which would allow fake users into the network.

Reinforcement learning and blockchain technology was combined to create a trust mechanism in edge computing [6]. The edge node is either a legitimate or selfish node intended to compromise other edges and limit resource consumption. The service trust was computed and maintained as ciphertext in the blockchain. In this way, security was stringent in the MEC environment.

The previous research works in MEC security have been developed based on authenticating the devices to ensure they are trusted. The ZTS model was depicted with the process of authentication and trust validation. The common issues illustrated in the state-of-the-art methods are addressed in the proposed ZTS maturity framework shown in Figure 1.

3. Zero-Trust Security

Zero-Trust Security was introduced by Forrester Research [19] based on the realisation that traditional security models operate on the outdated assumption that every entity inside an organisation's network should be trusted. The Zero-Trust Security framework is defined by various industry guidelines [8–11]. In essence, ZTS is a multilayer cybersecurity approach, including defence-in-depth controls [24]. It enables a resilient, consistent, and responsive approach to mitigate threat risks that could materialise due to sophisticated attacks and anomalies [12, 25]. The practice is rooted in the principle of "never trust, always verify."

The MEC entities cannot be permanently trusted due to the dynamic and shareable nature of MEC deployments. In heterogeneous environments like MEC, the infrastructure needs to examine the pertinent aspects of access requests instead of assuming trust based on a point-in-time assurance, that is, admission control or network location. Trust management for MEC introduces challenges as identified in [7, 26–29]. In the ZTS approach, the security posture of applications, users, and UE [11, 24] will be continually reassessed.

3.1. Preconditions for Zero-Trust Security. ZTS is built on a foundation of fundamental approaches and processes that must be implemented to support trust. We summarised them here for the MEC environment.

(i) *Mitigating Cloud-Native Cybersecurity Risks.* Cloud-native computing teams can quickly build and run cloud services using containers, immutable infrastructure, microservices, and declarative APIs, without disturbing the underlying infrastructure. Cloud-native security remains a challenge. For instance, MEC operators and developers will require administrative access to the network, and MEC interfaces can rely upon insecure web-based communications. Specific approaches for the MEC environment may be required.

(ii) *Aggregation of Trust Information.* During the trust establishment process, quantitative (e.g., real-time measurements) and qualitative information (e.g., subjective logic-based reputation) is brought together. Trust management models must provide the means for identifying trustworthy service providers in terms of different attributes, that is, data governance, compliance, and information security.

(iii) *Identity Management.* MEC-based Identity and Access Management (IAM) services pose fundamental challenges, for example, assessment of the existing ICT infrastructure, identity lifecycle management, maintaining a single identity (ID) across multiple platforms and MEC nodes, and compliance visibility of "who has access to what."

(iv) *Trust Evaluation and Assessment Approach.* The system must ensure that an entity that requests information access is authentic and that the request is granted the appropriate access level. The

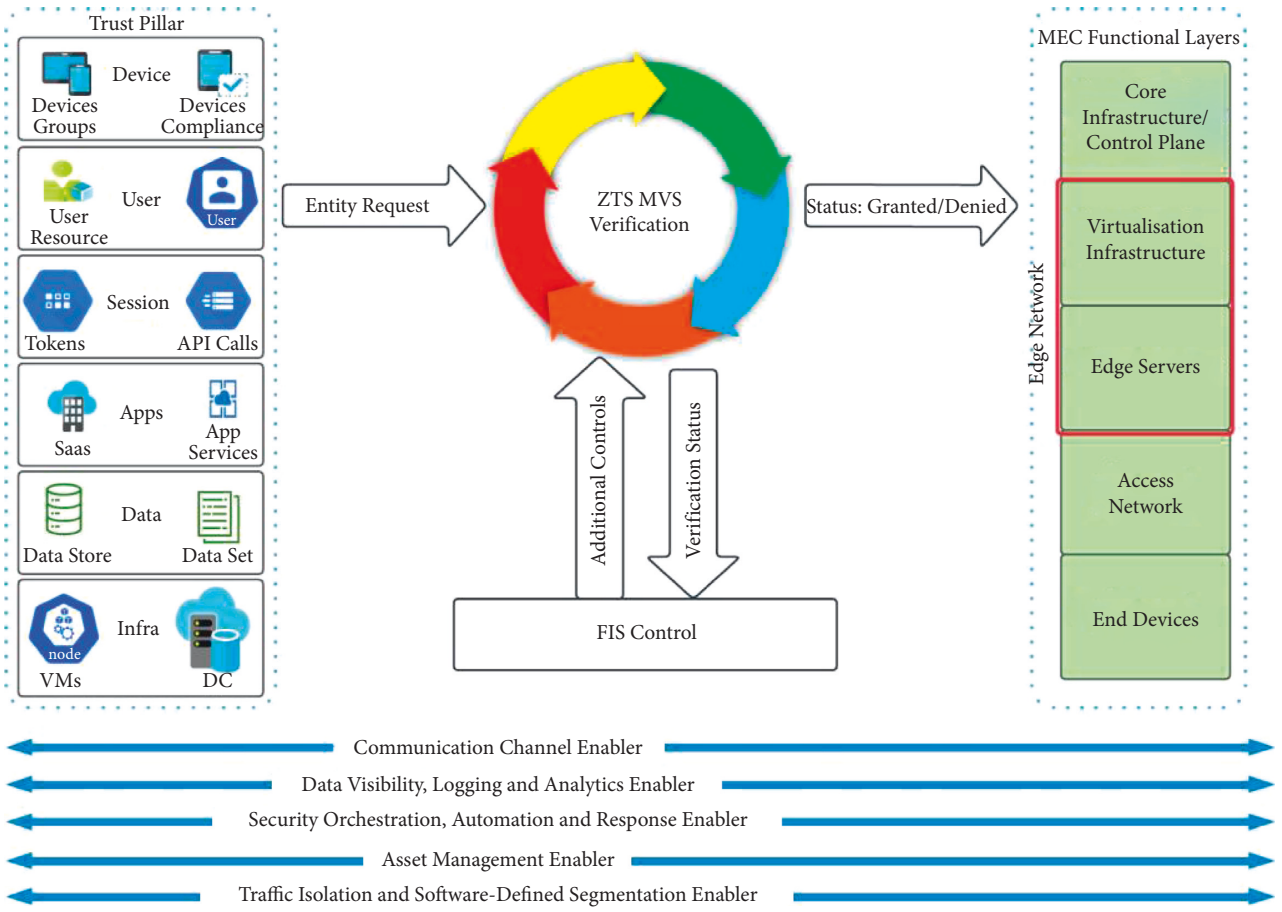

Figure 1: Zero-Trust Maturity lifecycle.

challenge is the rational and unbiased assignment of weights to the trust assessments' various trust values. Dynamic adaptability in weight assignment is a prerequisite in the MEC environment.

(v) *Session Trust.* A plethora of vulnerabilities are introduced with improper session management, which is primarily caused by poor software development practices. Threat actors could leverage session management-related vulnerabilities to take advantage of an authenticated user session. Adversaries could impersonate the user and perform actions from their account.

(vi) *Environment Visibility and Analytics.* Comprehensive visibility ensures effective operations. MEC operators must meet Service Level Agreements (SLA) and reduce the time to detect and remedy security-related issues. A lack of visibility into the workforce, workload, and data obscures security threats to the MEC environment.

(vii) *Policy Enforcement Capabilities.* The rapid evolution of MEC technology poses new security policy challenges. Threats arise not only from external factors but also from internal discrepancies. Establishing a dynamic policy control and enforcement layer is essential as the MEC service lifecycle is automated.

3.2. Existing Technologies to Support Zero-Trust Security. The proposed ZTS framework relies heavily on already existing technologies. Those technologies and architectural approaches are summarised as follows:

(i) *Identity and Access Management.* Adequate IAM is crucial to ensuring that access to resources is authenticated, authorised, and logged. The resources may be physical or virtual. Traditionally, nonhuman identification has been based on identity proxies, such as IP addresses, which are vulnerable to attack as they can be spoofed. Advanced IAM techniques use digital certificates, signatures, and multifactor authentication for robustness.

(ii) *Microsegmentation of MEC Networks.* Network segmentation is a technique for separating subnetworks initially for performance and then for security purposes. Security is now a primary motivation for the separation of network segments, and devices such as the organisation firewall can be used to inspect pass-through network traffic between segments, logging problematic traffic, permitting traffic from approved IP addresses, or conforming to specific protocols. With the ZTS framework, the need arises to do this with granularity. Essentially, each connected device (or at most a group of like-functionality devices) is in its own microsegment.

The intersegment firewall would restrict traffic to the minimal set of required protocols or traffic types.

(iii) *Ubiquitous Encryption.* Since adversaries are effectively inside the *perimeter* of the MEC environment and may snoop on traffic, all interactions across the MEC network should be encrypted end-to-end (E2E) where possible. In the context of ZT, E2E will drastically mitigate session management-related risks by protecting session information like tokens and payload being transferred. With the ZTS framework, data encryption and Digital Rights Management (DRM) may enable additional protections by limiting which MEC resources can be accessed and which actions can be taken with sensitive data, even if access is allowed. Analytic capabilities continuously monitor anomalous activity in accounts, data access, and mission-critical applications. Should a compromise occur, the threat is contained, the damage is limited, and the time for defensive systems to detect and initiate appropriate mitigating responses is significantly reduced.

(iv) *Dynamic and Adaptive Policies.* Security policy protects MEC assets from threats and disruptions and helps to optimally allocate network resources for enhancing productivity and efficiency in business processes. Individual security policy rules must determine whether to block or allow a session based on traffic attributes, such as the application, the user, the source and destination IP addresses, and the service. As an attribute is updated or a configuration changed, the security policy dynamically imports changes.

(v) *Inspection, Logging, and Analytics.* Awareness of threat actors is still needed. This is best enabled by extensive traffic inspection using microsegments and logging. Rather than manual inspection, logs are typically processed by security analytics and machine learning algorithms to detect anomalies that indicate attack attempts. The resulting analytics are applied to generate insightful reports and visualisations as required for the governance of the ZTS framework.

(vi) *Secure Access Service Edge (SASE).* SASE is a new generation of networking and security architectures [30]. It is built from the ground up using cloud-native technology, microservices, secure web gateways, cloud access security brokers, firewalls, and zero-trust network access. It requires that MEC entities, whether within or outside the originating network, be authenticated, authorised, and continuously validated for security posture and configuration before being granted or maintaining access to services, data, and applications. It refers to a security threat model assuming that all traffic is untrusted, whether generated from an internal or external network.

4. Proof-of-Concept Experimental Evaluation

The main objective of our Proof-of-Concept (PoC) experiments was to showcase the viability and potential benefits of our proposal to support the ZTS approach. We consider the UE IAM scenario where authentication should be handled as a norm. The experimental evaluation exhibits the proposal's potential benefits to dealing with a seamless authentication process utilising the PRESENT cryptography algorithm [31] to secure the UE credentials with guaranteed service continuity. The ZTS model has been implemented based on identity and continuous monitoring. In this work, constant monitoring is defined as verifying the UE identity for each request. The simulation and results are provided.

4.1. Proposed System. This section provides a practical realisation of our proposed architecture using the OMNeT++ network simulator v4.6, Java Development Kit v1.8, and Simulation of Urban MObility (SUMO-0.21.0).

OMNeT++ runs on a 64-bit processor architecture host (ESXi server v6.7) with a Windows 10 operating system that enables the development of MEC's architecture. It uses the edge devices implementation. The network switches are employed as edge devices to incorporate the ZTS model. Default packages in OMNeT++ provision the communication technology. The implemented wireless connectivity is based on LTE technology. The simulation setup has the following technical specifications: 16 GB RAM, 80 GB storage, and Quad processors CPU.

The proposed model is constructed in three layers: UE, edge, and cloud, as depicted in Figure 2. The layers and their capabilities are discussed in our previous work [3]. The MEC ZTS architecture consists of the following entities:

(i) *UE Layer.* UE typically integrates wireless or mobile technologies. The UE has a unique identity created during the registration process and connects to the closest access point. In our proposed model, UE is a mobile user. This layer connects to the nearest access point and forms a unique identity during registration. Each access point serves 50 mobile users. We used two access points in our simulation.

(ii) *Edge Layer.* The edge layer includes the physical infrastructure that incorporates computing and storage in distributed hyperconverged systems. In our proposed model, switches are the edge nodes equipped with the ZTS model to verify the credentials of the mobile users. Switches are responsible for forwarding the received traffic to the requested MEC resource for processing. We used three switches in our simulation.

(iii) *Cloud Layer.* The cloud provides computing and storage for applications and services that run locally. In our proposed model, the cloud represents a back-end server with multiple running applications called a cloud server. The cloud server is responsible for performing authentication and can serve the mobile user's traffic request. We used a single cloud server in our simulation.

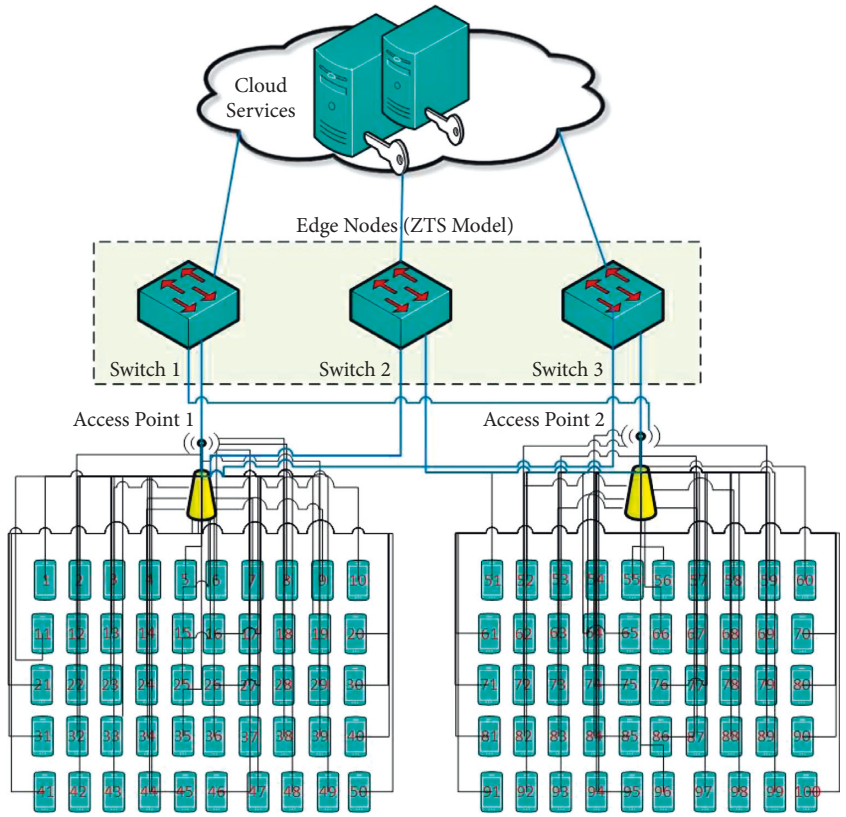

Figure 2: MEC ZTS model.

Identity and location are the two credentials that are utilised during the authentication process. The simulation parameters used for this model are summarised in Table 2. We used the PRESENT algorithm with 80-bit key size and 64-bit block size and processed 32 cycles per block.

In [3], we have incorporated the fundamental ZTS model into the MEC reference architecture. The ZTS model in the MEC host layer is not supposed to trust any user due to increasing insider threats. Hence, the user's trust is assured after successful authentication.

The ZTS model consists of two main components, policy decision point and policy enforcement point, as depicted in Figure 3.

The policy decision point is composed of the policy engine and policy administrator. The decision point is responsible for validating the user based on the credentials. This process is defined as authentication. Then the decision is initiated to the enforcement point for commanding the user device regarding authentication determination.

The authentication process takes place based on the registered user's credentials. Each adversary has unique behaviour in hijacking the communication channels and extracting secure information during transmission.

The working principles of our proposed solution can be seen in Figure 4.

4.2. Comparative Analysis. The simulation results were evaluated for success rate and authentication time. The risk of authentication information leakage is mitigated through key encryption. The efficiency of ZTS in the MEC environment was identified as a computing load to encrypt the UE identity to facilitate secure traffic flows. The success rate defines the permitted access requests for the submitted traffic flows. The edge nodes will verify each UE node's request and respond to the corresponding application.

Table 2: Simulation specifications.

Parameter	Setting
Network model	
Network area	2500 m × 2500 m
Number of mobile users	100
Number of access points	2
Number of edge nodes (switches)	3
Number of cloud servers	1
Speed of mobile users	50 Mps
Mobility model	TraCIMobility
Simulation run time	1000 sec
Security algorithm	PRESENT
Queue size	2 MB
Packet model	
Packet interval	5 sec
Number of packets	50–100
Flow timeout	2 sec
Service time	0.0098 ms
Delay	1 μs
Communication model	
Data rate	300 Mbps
Link bandwidth	5 Mbps
Transmission range	80 m

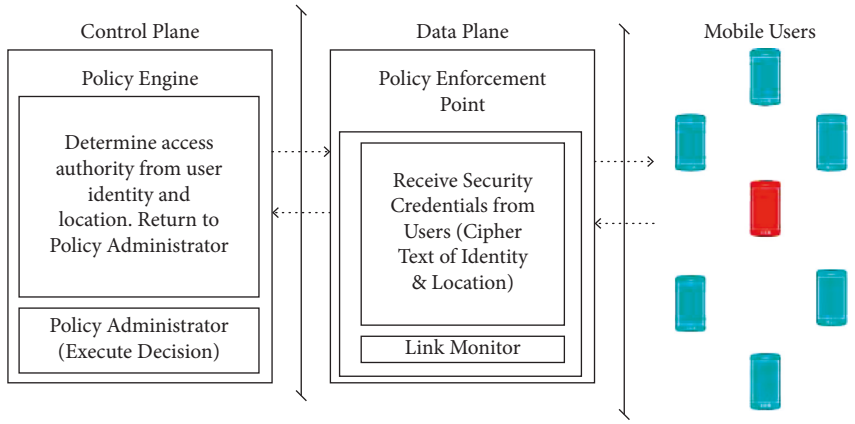

Figure 3: MEC reference architecture incorporating the ZTS model.

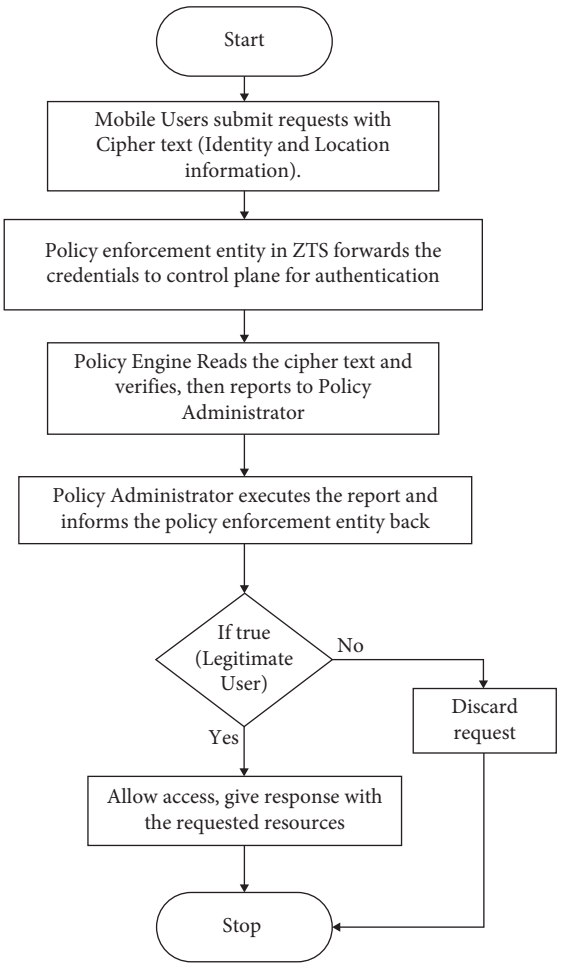

Figure 4: Flow working principles.

The success rate formula is calculated as follows:

$$\text{Sucess rate} = \frac{AU}{TU}. \quad (1)$$

The term AU denotes the number of authenticated users, while the TU denotes the total number of user requests.

The success rate performance measured is depicted in Figure 5(a), where the ZTS MEC attains a higher success rate than a legacy MEC environment. The success rate increase in ZTS MEC is due to the authentication by which the registered UEs are provided with access to the network resources. In contrast, in the absence of the proposed ZTS model, the MEC environment will follow the default verification of only UE identity. It allows UE requests for further processing without UE location verification.

The UE credentials are continuously managed in edge nodes and cross-verified with cloud-based authentication. As a result, Figure 5(a) shows an average of 80% success rate for ZTS MEC, while it is a 54% success rate for MEC without ZTS. Adding ZTS capability mitigates access being provided to unregistered UE and traffic from attackers. Blocking unauthorised access requests will improve the success rate since the submission of forged and compromised identities is reduced. The improvement in success rate can be further increased by strengthening the security posture.

The authentication time depicted in Figure 5(b) is defined as the time taken to authenticate UEs based on the identity and location values. The authentication time is plotted concerning the increment of connected UEs. The authentication time has minor variations since the ZTS model takes time to extract the credentials and carry out the verification process. A lightweight algorithm has been implemented, and this should reduce the cipher conversion time. A generalised relationship occurs between the number of UEs (authentication requests) and authentication time. However, this is unlikely the only linkage, as the computing and storage resources will be shared with other tasks and processes occurring in the devices hosting the ZTS controls.

4.3. Discussion and Future Directions. A ZTS implementation with a lightweight algorithm indicates that the ZTS model is a practical approach to improving security and reducing risk. Still, there is scope for further research into additional algorithms that can be developed for the MEC environment.

We aim to extend the proposed approach to incorporate an MFA framework to enhance authentication further and mitigate targeted attacks on edge nodes. Future research includes investigating methods to support multiple identity factors across edge nodes, for example, biometric (single or

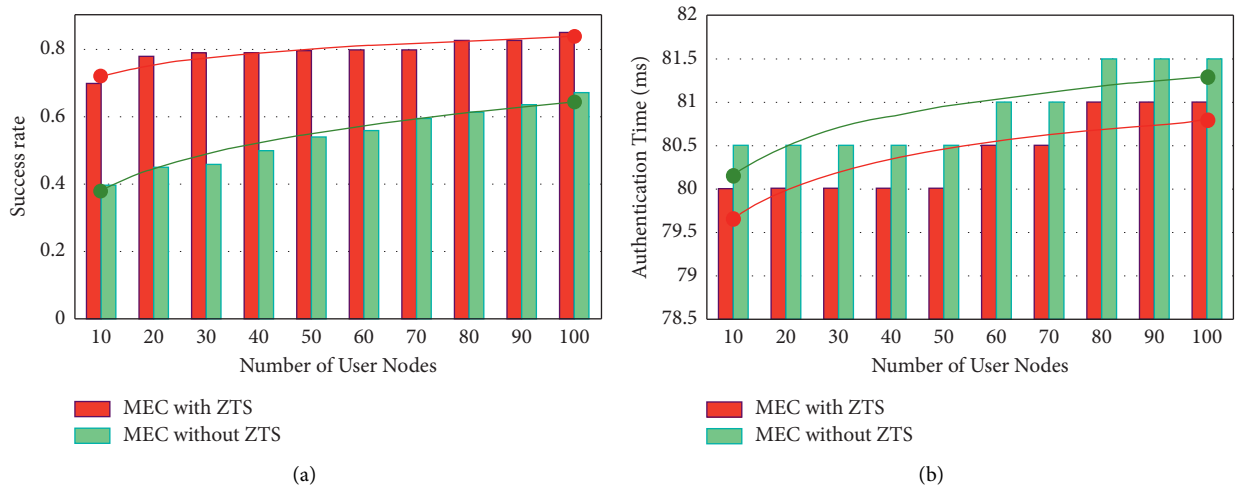

Figure 5: Zero-trust model performance metrics. (a) Comparison of success rate. (b) Comparison of authentication time.

hybrid) and Physical Unclonable Function (PUF). As edge nodes have limited resources, efficient authentication algorithms should be developed to enforce identity verification and authentication and ensure seamless service continuity across different edge nodes.

Furthermore, future research includes developing a centralised monitoring authority that validates the connected edge nodes according to fixed time intervals. Hereby, it assures that the edge nodes are not compromised and are still trusted for performing the authentication process.

Last but not least, future research includes introducing a comprehensive threat model that can offer a comparable and transparent classification of security properties.

MITRE ATT&CK [10] is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK framework is used to develop specific threat models and methodologies in the cybersecurity community. With the help of the ATT&CK framework, we aim to develop a threat catalogue that can be employed to derive a threat model, realise adversary behaviour, and perform a risk analysis process of specific edge scenarios and also to demonstrate our threat-based security testing approach over a case study, allowing mitigation and detection methods against the techniques.

We anticipate the adversary modelling to focus on threat actors whose tactics, techniques, and procedures are related to the following:

(i) IAM frameworks and protocols vulnerabilities, including session management.

(ii) Network-level vulnerabilities.

5. Securing Multiaccess Edge Computing

The GSM Association [32] has defined MEC technical requirements, functional blocks, and interface characteristics. The definition maps the requirements and architecture to specifications released by international standards bodies to identify gaps. The potential threats and risk mitigation methods that can be applied to MEC external interfaces and applications were discussed in [33, 34]. The MEC security approach needs a robust end-to-end methodology to consider the ETSI MEC standards and current support for potential security and cybersecurity regulations applicable to MEC environments [33].

To secure the MEC environment, we need to identify the elements which should be secured when MEC is operating and, importantly, the processes for maintaining the MEC environment that must be secured. We think of the secure elements as the "pillars" on which MEC security stands. The secure processes used to maintain MEC are described as security "enablers." We define these entities in this section, and Figure 1 shows the broad relationships.

5.1. Trusted Elements: The "Pillars". We have classified the MEC architecture entities into six groups: User Equipment (UE), User, Session, Application, Data, and Infrastructure. These are the pillars on which MEC security stands.

(i) *User Equipment "Device."* A UE could be any physical or virtual entity sitting at the edge side of the MEC landscape. It could be a sensor in an electrical substation with a compute node associated with it. The more complex UEs have the processing power to run analytics, apply artificial intelligence (AI) rules, and even store data locally to enable operations at the edge, for example, an assembly machine on a factory floor, video analytics, or an automobile. The UEs could handle analysis and real-time inferencing without the involvement of the edge server or upper layer.

(ii) *User.* Users are the entities that run the planned logic. A user can also be the human who uses the applications and services or those accessing the MEC landscape to perform privileged tasks, for example, administrative actions such as patching and configuration management. Users should be continuously verified over their lifecycle in the MEC environment.

(iii) *Session*. Following the successful admission of a user from a trusted UE, it is vital to continuously control the session that was established for the user to access the MEC resource.

(iv) *Application*. Applications of different types, for example, legacy three-tier applications, serverless applications, or containerised applications, can coexist in the MEC environment as either resources or consumers. The applications have their unique threat landscapes that are not similar to that experienced by users or UE. Applications in the MEC context can be integrated using various methods, including Service-Oriented Architecture (SOA) and Application Programmable Interfaces (API) [35].

(v) *Data*. Besides control or signalling data that is secured under other trust elements, that is, Session Trust, data security and privacy are critical enablers for a robust security model [2]. MEC must protect data against leakage, eavesdropping, and tampering, while the data is in various states, at rest, in transit, and being processed.

(vi) *Infrastructure*. While Device Trust focuses on UEs at the edge of the MEC landscape, infrastructure, that is, compute and storage nodes, underlying network technologies and associated systems [2], represents a critical threat vector that is continually assessed and reverified under the ZTS model.

5.2. Secure Processes: The "Enablers". In addition to securing the MEC elements, it is also necessary to secure the MEC operating processes. We outline the security enablers in this section.

(1) *Secure Communication Channels Enabler*. Data exchange and information access in the MEC distributed architecture require data transmission through secure and reliable channels. A trusted channel is necessary to transfer data resistant to threats, for example, eavesdropping and tampering. Communication channels in the MEC environment are within the access network, among edge and core entities, and in the virtualisation infrastructure [2]. The networks must be built with the "security by design" concept to establish trusted channels [36]. The physical network should be secure: this includes protecting the physical connections using cable termination, secure access to the network devices, radios and antennas, and hardening the administrative access to those devices. The secure configuration of the backbone routing protocols, wireless backhaul communications, and visualisation kernel is key to a robust network design [37]. A potential physical and logical breach is mitigated through deterrent safeguards, for example, port security, encryption, and routing authentication. The aim is a high-availability validated design to improve communication resiliency across the MEC functional domains.

(2) *Data Visibility, Logging, and Analytics Enabler*. There is a need for pervasive visibility and deep insights into the application, devices, packets, processes, network flows, and workload communications within an MEC environment. Correlating logs and data analytics in the MEC environment would enable the identification of anomalies across the MEC ecosystem [38]. Security analysts should be able to monitor the threat landscape in real time, triage security incidents, and apply intelligent security controls.

(3) *Security Orchestration, Automation, and Response Enabler*. Once trust is established across the trust elements and integrated with other enablers such as Data Visibility, Logging, and Analytics (DVLA), automating remediation actions becomes possible [39, 40]. MEC operators can make informed decisions to grant or deny access. By establishing trust, MEC operators gain visibility and gather analytics across the digital workspace environment. The core functional role of automation is to convert threat detection into threat prevention. The security industry offers Security Orchestration, Automation, and Response (SOAR) [41] solutions to remediate discrepancies in trust levels and automate cyber incident response procedures.

(4) *Asset Management Enabler*. Asset management is a standalone control category that manages the lifecycle of MEC entities and supports the governance and the operational aspects of the ZTS model [42]. An asset management system requires a policy document, defined procedures, and a tool to enforce an asset register. A baseline asset register covers devices, applications, users, data classification, and session information at the early stages of their lifecycles and in a dynamic manner by incorporating logs of shared services, such as DHCP and DNS.

(5) *Traffic Isolation and Software-Defined Segmentation Enabler*. MEC nodes are located on the network edge and integrate with the underlying network systems and security practices [2]. Approaches are required to minimise the impact of security breaches and prevent threat-actor lateral movement by segmenting access across the network and applications. Security policies [25] are needed that consider identity and contextual information from the entities where appropriate. Segmentation at the network and application layers drastically reduces the risk of a threat actor laterally moving between network subnets and application tiers. If a threat actor managed to compromise a specific segment, the blast radius will be limited to that space until a ZT verification cycle appears where that actor is likely to be detected. Software-defined segmentation [43] can achieve microsegmentation that is more granular than just network or application level. Software-defined segmentation spans the network, compute, storage, and process layers to protect resources and simplify policy development through granularity.

6. The Maturity Framework

In this section, we outline our maturity framework for MEC based on ZTS. The full details are summarised in Table 3. We first describe the security required for the pillars introduced in Section 5. Then, we outline the MVS and FIS states that are the starting point and end state, respectively, of our Zero-Trust Maturity (ZTM) framework.

We have developed the ZTM framework to assist MEC service providers and operators in assessing their zero-trust readiness and build a roadmap to adopt the ZTS model.

6.1. Securing the Trust Pillars. The six trust elements—the "pillars"—on which MEC security is based must be secured. Here, we outline the approach to securing each pillar:

(i) *UE Trust.* Once an entity has been granted access to an MEC resource, the UE is granted a trust level. The MEC operator must monitor, manage, and control the security posture using policies based on a predetermined security baseline. MEC operators can determine if the UE can still be trusted and deemed compliant by interrogating the UE posture using Network Access Control (NAC) [38] with serial device profiling. A Unified Endpoint Management (UEM) [44] solution may enable a MEC operator to manage, monitor, and control all devices, that is, UE, nodes, and IoT across all MEC functional layers. MEC can embrace eXtended Detection and Response (XDR) [45] technology to improve device security by detecting anomalies or malicious activities against predefined configuration baselines.

(ii) *User Trust.* Users should continuously be verified over their lifecycle in the MEC environment. There are occasions when Machine-to-Machine (M2M) integration involves different types of users, for example, service accounts. User IAM [46] is crucial to the ZTS model. Users are authenticated and authorised by correlating information sources such as data points, user identity, service or workload, location, data classification, device health, and anomalies. Password-based user authentication has been proven to be ineffective and inefficient without enhancements like two-factor authentication requiring human intervention [47]. With the modernisation of user authentication, enabling Single Sign-On (SSO) capability may elevate access control and improve user experience. A highly available and robust conditional access engine can make decisions using dynamic and contextual data. Technology building blocks that enable access are password-less authentication, for example, biometrics, certificates, multifactor authentication (MFA), conditional access policies, and dynamic risk scoring [48]. Access requests are continuously reverified to evaluate user trustworthiness.

(iii) *Session Trust.* Session trust can include but is not limited to HTTP session parameters, SSH session parameters, and the changing status of an accessed resource. Session trust is a critical component of the ZTS model, where these parameters are continuously monitored and controlled. The approach is that the entities must have restricted access to only the resources needed to perform the task at hand. Secure session management mitigates the risks of session hijacking and other session management-related risks. MEC operators limit entity access with Just-In-Time (JIT), Just-Enough Access (JEA), and risk-based adaptive policies by employing the principle of least privileged access to resources [37]. Technology building blocks to implement session trust are transport layer security (data encryption at rest), segmentation, secret management, and Privilege Access Management (PAM) in the M2M context [11, 23].

(iv) *Application Trust.* Application trust uses vulnerability scanning and more specialised scans on API endpoints and containerised applications. Application trust is also responsible for ensuring only authorised applications can be installed using a trust mechanism. The ZTS framework implies control of application run time. It also suggests using real-time analytics, monitoring abnormal behaviour, and discovering rogue applications and covert channels. The MEC security architecture enables application instances to securely and seamlessly consume resources from an authorised device by authorised entities within a specified time window. For legacy applications not designed for the ZTS model, security would add protection in isolation and conduct continuous monitoring. To isolate and modernise legacy applications, one can utilise an application environment, API-based or SOA-driven secure middleware to bridge between the legacy and the future state architecture.

(v) *Data Trust.* Data trust aims to enforce data confidentiality, integrity, and availability. Technology controls, for example, Data Loss Prevention (DLP) [49], data flow, and encryption methods support data confidentiality by mitigating the risk of data exfiltration and private information leakage [24]. Infrastructure availability supports data availability, while robust change management and secure software development lifecycle practices support data integrity by implementing data integrity checks [37]. Employing data classification control would optimise data trust [11].

(vi) *Infrastructure Trust.* Infrastructure technology enablers can use telemetry data to detect attacks and anomalies and automatically flag risky behaviour. Also, protective actions may be taken against unauthorised equipment, a rogue device, or a VM or container that has been moved by an adversary or accidentally by a high-availability control. Where

Table 3: Zero-Trust Maturity framework.

Trust pillar	Minimal Viable Security	Fully Implemented Security
User Equipment (device)	NAC with device posture/profiling at admission time. Dynamic authorisation based on posture checks. Device unique identifications are part of the asset register and Identity Provider (IdP)	Ongoing device profiling. Dynamic authorisation: granular access control is gated on the device risk profile. Endpoint threat detection and response (ETDR) solutions [50] are deployed and centrally managed. NAC and ETDR logs ship to Security Information and Event Management (SIEM)
User	Highly available, central user Identity and Access Management (IAM) covering authentication options, authorisation methods, and accounting. Users are authorised based on the principle of least privilege with multiple authorisation profiles. MFA for all users. SSO where applicable. Access and access change logging. Full integration with all ZTS enablers. Federated access	MFA with passwordless authentication is enabled. User behaviour and location changes are incorporated into the authentication and authorisation decision
Session	User-initiated sessions set behind SSO. Central secret management solution for credentials, API keys, key pairs, and authentication passwords. Reverify sessions based on frequent predefined intervals and consider changes from other functional layers. All traffic (payload and signalling) is encrypted E2E. PAM controls the session lifecycle	Secrets management must be decentralised wherever possible at a functional layer level. Authentication and authorisation sessions do not persist. Once established, all sessions are continuously verified. Enable logging across sessions' lifecycle inclusive of cipher configuration changes. Develop security monitoring use-cases. Apply network DLP and ship DLP logs to SIEM
Application	Application patching and hardening. Continuous application vulnerability scanning. Continuous application discovery on-prem and in the cloud using cloud access security broker (CASB) solutions [51]. First release and major change application penetration testing. Initial static code review. Developer security training	Application whitelisting. Automated code review. Web Application Firewall (WAF) [52]
Data	Classify and label data. Govern access decisions by data classification	Augment data classification by unsupervised machine learning models. Govern access decisions by a centralised MEC security policy engine "orchestrator." Integrate DLP solution with data classification
Infrastructure	Workloads, that is, servers and VMs, are initially hardened, uniquely identified, and security-baselined. Monitor workloads and trigger alerts for abnormal behaviour or change against the security baseline. Build an interoperable "data centre" infrastructure (network, compute, and storage). Deploy ingress/egress cloud microperimeters, segmentation, and Just-In-Time IAM verification. Enable cloud-native filtering and protection for known threats. Encrypt user-to-application traffic	Prevent unauthorised deployments and trigger alerts. Granular access control and visibility are available across all compute workloads, network, block, and object-level storage. Deploy a distributed full suite microsegmentation. Apply DLP to all infrastructure-type entities and ship DLP logs to SIEM for correlation and dynamic policy enforcement. Enable ML-based threat protection and filtering with context-based signals. Encrypt all traffic (payload and signalling) E2E

possible, MEC infrastructure must employ real-time threat protection, end-to-end encryption, monitoring, analytics, and microsegmentation to prevent adversaries from moving laterally across the network [40].

6.2. Minimal Viable Security. MVS defines the primary measures that need to be in place to effectively protect against the most common types of cybersecurity attacks and suggests the minimum security controls required to achieve that. The details of what it takes to achieve MVS in each ZTS trust pillar are described in Table 3. The security measures have been adopted based on three considerations:

(i) What security measures are observed to be in place across a wide range of MEC deployments?

(ii) What security measures were commonly lacking in widely known security breaches, the absence of which directly contributed to their occurrence?

(iii) Security measures put forward by a range of core industry cybersecurity standards are fundamental to implementing an effective MEC cybersecurity program.

MVS is not about being "bulletproof" or achieving "leading edge" security. It is about implementing a series of baseline protective measures that practical experience shows can work together to place MEC operators in a much better position to handle cyberattacks than those not embracing MVS.

Adversaries know that the MEC environment has multiple entry points through the functional layers [2]. Adopting the ZTS fundamentals would implement various security controls across the MEC ecosystem lifecycle to continuously verify trust [29]. Security controls would collaborate and constantly feed the ZTS enablers to maintain visibility across the known threats and become aware of emerging vulnerabilities. These controls collectively represent the ZTS policy enforcement points and significantly reduce the time required to detect and recover from a breach [11].

Although MVS is a set of basic, mandatory measures, we understand that adopting such controls may still not be possible in full at the beginning of the security journey for MEC environments. Therefore, within a given trust pillar, the security controls are prioritised to provide a clear set of step-by-step recommended actions to improve the MEC security control environment.

Figure 1 illustrates how MEC entities must persistently achieve MVS to stay connected and functional. It shows the need for a given entity within a given MEC functional layer to fulfil MVS requirements to remain in that layer and subsequently part of the MEC landscape. The security controls against MVS requirements will verify an entity's request. The entity's status will be reported to the control plane of the relevant MEC functional layer, and a decision will be made. As MEC operators improve their security practices and master continuous security operations, the MVS requirements can transition to more demanding requirements to achieve a Fully Implemented Security (FIS) state. The MVS verification processes feed back to the environment administrators and analytics engines to drive an informed decision about mandating more controls as part of the FIS requirements. The trust enablers are shown as foundational elements and prerequisites that support entities' compliance with the requirements of ZTS at MVS and FIS.

6.3. Fully Implemented Security. The ZTM framework embeds comprehensive security monitoring, granular risk-based access controls, and system security automation in a coordinated manner throughout every aspect of the infrastructure to protect critical assets, that is, people, infrastructure, process, and applications, in real time within a dynamic threat environment. This approach can be incremental, reduces risk at each step, ensures privacy and protection of sensitive data, improves visibility, and automates response times to reach optimal maturity.

In the MVS state, we assume that MEC operators have started their ZTS journey and have progressed in a few key areas. In the FIS state, MEC operators have already adopted the ZTM model and made significant security architecture improvements. The FIS state, described in Table 3, can be used by MEC operators and service providers to assess their ZTS readiness, define security controls, and plan to implement a practical ZTM framework. Our proposed ZTM framework would help MEC operators establish a comprehensive security strategy to create a cybersecurity solution that is

(i) *Flexible.* The ZTM framework is built for the future to protect against new threats, adapt to the landscape, and scale to meet the MEC's changing needs. It is easy to install, maintain, and operate and can be tailored to address each operator's unique requirements and protect MEC assets.

(ii) *Actionable.* Zero-trust assessment determines the health of entities across the MEC network. With real-time security posture assessment, operators can quickly identify and update policies and operating system settings that are outdated or increase risk. It enables ongoing, automatic monitoring, detection, and response capabilities.

(iii) *Focus on Identity.* The ZTM framework offers full identity audits and an understanding of accounts, protocols, and services. It enables multiple APIs into partner MFA/IAM providers, SIEM, and SOAR technologies, allowing operators to identify the connected entities and control those entities in real time.

(iv) *Comprehensive.* The ZTM framework provides continuous visibility and security across a variety of points, including users, device type, firmware versions, operating system versions, vulnerabilities, patch levels, and applications installed. Also, it provides insights into security incidents, programmatic logins in the case of identity incidents, and many types of lateral movement.

(v) *Open API Platform.* The ZTM framework provides a full-spectrum set of state-aware APIs that enable operators and the partner ecosystem to integrate third-party tools that help implement the zero-trust architecture seamlessly.

Adopting the ZTS approach is expected to uplift the security posture of MEC implementation significantly. However, the additional controls and processes required to implement ZTS would likely introduce overheads at the infrastructure performance and user experience levels. The cyberthreat mitigation process should always be a balance between process operational excellence and security.

By implementing a control suite consistent with MVS, it is expected that a successful compromise would be the result of a legitimately sophisticated cyberattack rather than a failure to enforce a control that could reasonably have been expected to be in place.

7. Conclusion

The ever increasing complexity of MEC environments and the potential for threat actors to compromise MEC hosts require a defensive focus change. This paper has presented the MEC trust management challenges and proposed a comprehensive ZTS model and maturity framework.

Adopting a zero-trust mindset and leveraging ZTS principles will enable MEC operators to control how entities engage with data and the underlying network and limit entity access while also looking for anomalous or malicious activity. The zero-trust mindset focuses on securing critical data and access paths by enhancing trust as much as possible, coupled with verifying and regularly reverifying access permissions. While a ZTS model is effective when it has been integrated across the entire MEC environment, operators must consider a phased approach to deployment that targets infrastructure and systems based on their maturity level, available resources, and critical security priorities.

ZTS can be a substantial investment and should be carefully aligned with current business objectives to be sustainable. The first step of the ZTS journey does not have to be a large shift towards an optimal maturity level. The MEC operators may benefit significantly by utilising hybrid infrastructure that leverages existing investments and realises the value of ZTS initiatives quickly. Fortunately, each step forward will make a significant difference in mitigating risk and raising trust in the MEC environment.

Research into MEC, ZTS, and ZTM is ongoing, and selected challenges have been identified. Security threats and proposed responses for the MEC environment have been provided. The potential impact of MEC on future network operations is significant. Hence, it should be anticipated that MEC ZTS controls and technologies will grow in importance.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] ETSI, "Multi-access edge computing (MEC); framework and reference architecture," Tech. Rep. GS MEC 003, ETSI, Sophia Antipolis, France, 2020.

[2] B. Ali, M. A. Gregory, and S. Li, "Multi-access edge computing architecture, data security and privacy: a review," *IEEE Access*, vol. 9, Article ID 18706, 2021.

[3] B. Ali, M. A. Gregory, and S. Li, "Uplifting healthcare cyber resilience with a multi-access edge computing zero-trust security model," in *Proceedings of the 2021 31st International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 192–197, IEEE, Sydney, Australia, 2021 November.

[4] W. Kong, X. Li, L. Hou, and Y. Li, "An efficient and credible multi-source trust fusion mechanism based on time decay for edge computing," *Electronics*, vol. 9, no. 3, p. 502, 2020.

[5] W. Kong, X. Li, L. Hou, J. Yuan, Y. Gao, and S. Yu, "A reliable and efficient task offloading strategy based on multi-feedback trust mechanism for IoT edge computing," *IEEE Internet of Things Journal*, no. 1, p. 1, 2022.

[6] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Secure computation offloading in blockchain based IoT networks with deep reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3192–3208, 2021.

[7] X. Zhang, W. Wu, S. Yang, and X. Wang, "Falcon: a blockchain-based edge service migration framework in mec," *Mobile Information Systems*, vol. 2020, pp. 1–17, 2020.

[8] S. Turner and C. Cunningham, "The Zero Trust Extended (ZTX) Ecosystem," 2019, https://www.forrester.com/report/The-Zero-Trust-eXtended-ZTX-Ecosystem/RES137210.

[9] N. MacDonald, L. Orans, and J. Skorupa, "Zero Trust Is an Initial Step on the Roadmap to CARTA," *Gartner Research*, 2018.

[10] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: design and philosophy," Technical Reports Series 10AOH08A-JC, The MITRE Corporation, McLean, Virginia, USA, 2018.

[11] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust archi-tecture," Tech. Rep. SP 800-207, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2020.

[12] A. Kerman, O. Borchert, S. Rose, and A. Tan, *Implementing a Zero Trust Architecture*, National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA, 2020.

[13] G. Fragkos, C. Minwalla, J. Plusquellic, and E. E. Tsiropoulou, "Artificially intelligent electronic money," *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 81–89, 2021.

[14] S. Almajali, H. B. Salameh, M. Ayyash, and H. Elgala, "A framework for efficient and secured mobility of IoT devices in mobile edge computing," in *Proceedings of the 2018 Third International Conference on Fog and mobile Edge Computing (FMEC)*, pp. 58–62, IEEE, Barcelona, Spain, April 2018.

[15] B. D. Deebak, F. Al-Turjman, and L. Mostarda, "Seamless secure anonymous authentication for cloud-based mobile edge computing," *Computers & Electrical Engineering*, vol. 87, Article ID 106782, 2020.

[16] K. Kaur, S. Garg, G. Kaddoum, M. Guizani, and D. N. K. Jayakody, "A lightweight and privacy-preserving authentication protocol for mobile edge computing," in *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Waikoloa, HI, USA, December 2019.

[17] P. Kumar and M. Liyanage, "Efficient and anonymous mutual authentication protocol in multi-access edge computing (MEC) environments," *IoT Security*, pp. 119–131, 2020.

[18] Y. Hou, S. Garg, L. Hui, D. N. K. Jayakody, R. Jin, and M. S. Hossain, "A data security enhanced access control mechanism in mobile edge computing," *IEEE Access*, vol. 8, Article ID 136119, 2020.

[19] J. Kindervag, *Build Security into Your Network's DNA: The Zero Trust Network Architecture*, Forrester Research Inc, Cambridge, MA, USA, 2010.

[20] D. Eidle, S. Y. Ni, C. DeCusatis, and A. Sager, "Autonomic security for zero trust networks," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 288–293, IEEE, New York, NY, USA, October 2017.

[21] S. Mehraj and M. T. Banday, "Establishing a zero trust strategy in cloud computing environment," in *Proceedings of the 2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, IEEE, Coimbatore, India, January 2020.

[22] A. Aref and T. Tran, "A hybrid trust model using reinforcement learning and fuzzy logic," *Computational Intelligence*, vol. 34, no. 2, pp. 515–541, 2018.

[23] B. Chen, S. Qiao, J. Zhao et al., "A Security Awareness and protection System for 5G Smart Healthcare Based on Zero-Trust Architecture," *IEEE Internet of Things Journal*, vol. 8, 2020.

[24] N. Papakonstantinou, D. L. Van Bossuyt, J. Linnosmaa, B. Hale, and B. O'Halloran, "A zero trust hybrid security and safety risk analysis method," *Journal of Computing and Information Science in Engineering*, vol. 21, no. 5, Article ID 050907, 2021.

[25] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, "Never trust, always verify: a multivocal literature review on current knowledge and research gaps of zero-trust," *Computers & Security*, vol. 110, Article ID 102436, 2021.

[26] S. Dhar and I. Bose, "Securing IoT devices using zero trust and blockchain," *Journal of Organizational Computing & Electronic Commerce*, vol. 31, no. 1, pp. 18–34, 2021.
[27] R. Latif, M. U. Ahmed, S. Tahir, S. Latif, W. Iqbal, and A. Ahmad, "A Novel Trust Management Model for Edge Computing," *Complex & Intelligent Systems*, pp. 1–17, 2021.
[28] M. Liyanage, P. Porambage, A. Y. Ding, and A. Kalla, "Driving forces for multi-access edge computing (MEC) IoT integration in 5G," *ICT Express*, vol. 7, no. 2, pp. 127–137, 2021.
[29] S. Teerakanok, T. Uehara, and A. Inomata, "Migrating to zero trust architecture: Reviews and challenges," *Security and Communication Networks*, vol. 2021, Article ID 9947347, 2021.
[30] Palo-Alto-Networks, "What is SASE?," 2021, https://www.paloaltonetworks.com/cyberpedia/what-is-sase.
[31] R. Chatterjee and R. Chakraborty, "A modified lightweight PRESENT cipher for IoT security," in *Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp. 1–6, Gunupur, India, March 2020.
[32] Gsma, "Operator platform telco edge require-ments," Tech. Rep. OPG.02, GSM Association, London, England, UK, 2021, https://www.gsma.com/futurenetworks/wp-content/uploads/2021/06/OPG-Telco-Edge-Requirements-2021.pdf.
[33] A. Sophia, "M. E. C. security: Status of Standards Support and Future Evolutions," 2021, https://www.etsi.org/newsroom/press-releases/1933-etsi-releases-a-white-paper-on-edge-computing-security.
[34] D. Soldani, "6G fundamentals: vision and enabling technologies," *Journal of Telecommunications and the Digital Economy*, vol. 9, no. 3, 2021.
[35] L. Tomaszewski, S. Kukliński, and R. Kołakowski, "A new approach to 5G and MEC integration," in *Artificial Intelligence Applications And Innovations. AIAI 2020 IFIP WG 12.5 International Workshops*, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds., Springer International Publishing, Manhattan, NY, USA, 2020.
[36] P. Ranaweera, A. Jurcut, and M. Liyanage, "MEC-enabled 5G use cases: a survey on security vulnerabilities and countermeasures," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–37, 2022.
[37] D. Greenwood, "Applying the principles of zero-trust architecture to protect sensitive and critical data," *Network Security*, vol. 2021, no. 6, pp. 7–9, 2021.
[38] L. Rikhtechi, V. Rafeh, and A. Rezakhani, "Secured access control in security information and event management systems," *Journal of Information Systems and Telecommunication*, vol. 9, no. 33, pp. 67–78, 2021.
[39] Palo-Alto-Networks, "Zero Trust enterprise," 2021, https://www.paloaltonetworks.com/resources/guides/zero-trust-overview.
[40] Cisco, "Securing the 5G Core (5GC and Evolved Packet Core (EPC) with cisco Security," 2019, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/service-provider-security-solutions/white-paper-c11-742166.html.
[41] Palo-Alto-Networks, "What is SOAR?," 2021, https://www.paloaltonetworks.com/cyberpedia/what-is-soar.
[42] Y. K. Teoh, S. S. Gill, and A. K. Parlikad, "IoT and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning," *IEEE Internet of Things Journal*, vol. 1–8, p. 1, 2021.
[43] I. H. Abdulqadder and S. Zhou, "SliceBlock: context-aware authentication handover and secure network slicing using DAG-blockchain in edge-assisted SDN/NFV-6G environment," *IEEE Internet of Things Journal*, vol. 1–18, p. 1, 2022.
[44] Citrix-Systems, "Tech Brief: Citrix Endpoint Management," 2021, https://docs.citrix.com/en-us/tech-zone/learn/tech-briefs/citrix-endpoint-management.html.
[45] Palo-Alto-Networks, "What is XDR? — understanding extended detection and response (XDR)," 2019, https://www.paloaltonetworks.com/cyberpedia/what-is-xdr.
[46] B. Cao, W. Liu, and M. Peng, "Integration of MEC and Blockchain," *Wireless Blockchain: Principles, Technologies And Applications*, Hoboken, NJ, USA, 2021.
[47] D. Xiao, M. Li, and H. Zheng, "Smart privacy protection for big video data storage based on hierarchical edge computing," *Sensors*, vol. 20, no. 5, p. 1517, 2020.
[48] W.-Z. Zhang, I. A. Elgendy, M. Hammad et al., "Secure and optimized load balancing for multitier IoT and edge-cloud computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8119–8132, 2021.
[49] J. Chen, Y. Yan, S. Guo, Y. Ren, and F. Qi, "A system for trusted recovery of data based on blockchain and coding techniques," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–12, 2022.
[50] M. Yahuza, M. Y. I. B. Idris, A. W. B. A. Wahab et al., "Systematic review on security and privacy requirements in edge computing: state of the art and future research opportunities," *IEEE Access*, vol. 8, Article ID 76541, 2020.
[51] S. Ahmad, S. Mehfuz, and J. Beg, "Enhancing security of cloud platform with cloud access security broker," in *Information and Communication Technology for Competitive Strategies (ICTCS 2020),* Springer, Singapore, 2021.
[52] A. Moradi Vartouni, M. Teshnehlab, and S. Sedighian Kashi, "Leveraging deep neural networks for anomaly-based web application firewall," *IET Information Security*, vol. 13, no. 4, pp. 352–361, 2019.