A Zero Trust Access Authentication Scheme Based on 3GPP Networks for Edge Services

Aobo Duan, Ruhui Ma, Jin Cao, Shiyang He, Hui Li

***Abstract*—The evolution of 3GPP networks has revolutionized edge service accessibility, creating opportunities for applications requiring low latency and high reliability. However, traditional perimeter-based security models have become inadequate in the face of distributed environments and the proliferation of Bring Your Own Device (BYOD) policies. Current authentication schemes for edge services lack integration with 3GPP networks and fail to balance security with the efficiency requirements of edge computing. This manuscript proposes a novel zero trust access authentication scheme for edge services in 3GPP networks, implementing a hierarchical framework with adaptively selected authentication levels based on device trust values. The scheme introduces three authentication mechanisms—strong, moderate, and weak—that are triggered based on dynamic trust thresholds. Security analysis demonstrates that the proposed scheme satisfies critical security objectives including mutual authentication, key agreement, privacy preservation, and resistance against common attacks. Performance evaluation shows that while the strong authentication process introduces slightly higher overhead compared to existing schemes, the moderate and weak authentication processes significantly reduce computational and communication costs. This research contributes a comprehensive zero trust solution that effectively integrates with 3GPP network architecture while maintaining the strict security guarantees required for edge services.**

***Index Terms*—Zero Trust, access authentication, 3GPP networks**

I. Introduction

A. Background

The evolution of 3GPP networks has significantly transformed how devices access edge services, creating new opportunities for applications requiring low latency and high reliability. With the advancement of 5G/6G technologies, edge computing services have become essential for various applications including industrial automation, content delivery, and vehicle-to-everything (V2X) communications [1]. These services leverage the distributed nature of edge computing to process data closer to end users, reducing latency and bandwidth requirements [2]. The integration of edge computing with 3GPP networks has enabled seamless connectivity for diverse devices, allowing them to access computational resources and services without relying on distant cloud infrastructure [3].

However, the traditional security methods of trust models based on the perimeter have shown deficiencies [4]. The popularization of the "Bring Your Own Device" (BYOD) strategy [5] has changed the way users access organizational resources [6], triggering security vulnerabilities as employees can access sensitive resources from multiple locations using personal devices [7]. This leads to the failure of traditional boundary defense, especially for distributed edge services, which often involves resource-constrained devices frequently connecting to different network segments [8].In response to these challenges, Zero Trust Architecture (ZTA) has emerged as a vital security paradigm for 3GPP-based edge services [9]. As articulated by Kindervag and Balaouras, Zero Trust eliminates the concept of default trust, requiring continuous verification regardless of connection origin or previous authentication status [10] [11].

Recent advancements in zero-trust security for edge computing include two distinct approaches: Belal Ali et al. [12] have introduced a Zero-Trust Security (ZTS) maturity assessment framework specifically tailored for Multi-Access Edge Computing (MEC) ecosystems, while Hichem Sedjelmaci et al. [13] developed a robust Zero Trust Architecture (ZTA)-driven intrusion detection system aimed at safeguarding 6G-enabled edge computing infrastructures against sophisticated network threats. These works highlight the critical importance of implementing zero trust for edge services, as traditional trust boundaries no longer provide adequate protection against sophisticated attacks targeting the expanded attack surface of distributed edge environments [14].

Despite its promise, implementing Zero Trust for edge services in 3GPP networks presents several significant challenges. Traditional centralized authentication systems introduce considerable latency, as authentication requests must traverse the network to central servers unacceptable for real-time edge services requiring low-latency processing. Additionally, these centralized systems represent single points of failure and lack the scalability needed in dynamic edge environments. Resource constraints of edge devices further complicate the implementation of comprehensive security measures, as many traditional zero trust solutions require computational resources beyond what is available at the edge. These challenges underscore the limitations of traditional, often rigid, authentication mechanisms and necessitate a novel authentication scheme. Such a scheme, as will be detailed in this paper, moves beyond fixed authentication protocols by introducing a flexible and adaptive approach. This allows for varying levels of authentication based on dynamic conditions, thereby maintaining the strict security guarantees of Zero Trust while efficiently addressing the unique requirements of edge services in 3GPP networks.

Ruhui Ma and Shiyang He are with the State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, China. Ruhui Ma is the corresponding author, e-mail: rhma@xidian.edu.com.

Manuscript received April 19, 2021; revised August 16, 2021.

B. Related Work

Research on zero trust access authentication for edge services is evolving rapidly, addressing the challenges of securing modern distributed environments. Zero Trust Architecture (ZTA) has emerged as a critical security paradigm that operates on the principle that no entity—whether user, device, or system—should be inherently trusted, requiring continuous verification regardless of connection origin or previous authentication status. This approach is particularly relevant for edge computing environments where traditional perimeter-based security models have proven inadequate.

The 3GPP standards organization has addressed edge authentication concerns in its technical specifications. In TR 23.758 [16], authentication of clients for edge services is identified as a critical architectural requirement. The 3GPP edge computing framework includes mechanisms for user authentication, incorporating service provisioning and capability exposure needed for secure access. However, these standards focus primarily on traditional authentication models and have not fully integrated zero trust principles into edge service authentication.

For edge services in 3GPP networks, several authentication schemes have been proposed. Belal Ali et al. [25] developed a dual fuzzy methodology for trust-aware authentication and task offloading in MEC, which considers the resource constraints of edge servers while minimizing task completion time. Similarly, Shah et al. [15] introduced LCDA (Lightweight Continuous Device-to-Device Authentication), specifically designed for Zero Trust Architecture in edge environments, which provides continuous authentication mechanisms rather than relying on static, one-time verification.

Several researchers have explored blockchain-based approaches for zero trust access control. Lin et al [17] proposed BSeIn, a blockchain-based secure mutual authentication with a fine-grained access control system for Industry 4.0. Similarly, Huang et al [18] developed a blockchain-assisted Zero Trust security model for vehicular networks that enhances dependability in highly dynamic environments. For IoT environments, Huang et al [19]created ZT-access, combining zero trust access control with attribute-based encryption to protect against compromised devices.

For authentication between edge devices, various lightweight schemes have been proposed. A secure low-cost authentication scheme using SRAM-based Physical Unclonable Functions (PUFs) generates unique ”digital fingerprints” for device identification [20]. Another approach employs hypergraph hashing techniques to establish secure authentication between IoT edge devices, addressing the resource constraints typical in these environments [21].

Despite these advancements, there remains a significant gap in integrating zero trust principles directly into the 3GPP network architecture for edge services. Current solutions are either focused on specific application domains or fail to address the unique requirements of edge services in 3GPP networks. Our proposed scheme addresses this gap by introducing a hierarchical authentication framework that combines core and edge network components, providing a comprehensive zero trust solution tailored specifically for 3GPP-based edge services.

C. Contribution

Our research introduces a comprehensive zero trust access authentication scheme for edge services in 3GPP networks, featuring an initial authentication process and an adaptive re-authentication process. The adaptive re-authentication process dynamically selects among strong, moderate, and weak authentication levels, triggered by changes in the device's trust value after the initial authentication. The key contributions of our work are as follows:

- Proposed a novel hierarchical architecture that integrates NIST Zero Trust principles into 3GPP network edge services. This architecture enhances security and flexibility by enabling effective management of user identity legitimacy in dynamic edge environments through adaptively selected authentication levels.
- The initial authentication process establishes a robust security foundation for device access, ensuring strong mutual authentication, secure key agreement, and essential privacy preservation. This process is critical for initially verifying device legitimacy and setting up secure communication channels for subsequent interactions.
- The adaptive re-authentication process offers significant performance advantages and flexibility by dynamically selecting the authentication protocol based on the trust value of the device.
- Our scheme's security properties were formally verified using Tamarin, confirming that all authentication processes satisfy essential requirements including mutual authentication, key agreement, privacy preservation, and perfect forward/backward secrecy while demonstrating resilience against common attack vectors including replay, man-in-the-middle, impersonation, and passive attacks.

The rest of this paper is organized as follows: Section II presents the preliminaries and the trust evaluation mechanism. Section III introduces the system model and security objectives. Section IV details the proposed scheme. Section V provides security analysis and formal verification. Section VI evaluates the performance of our scheme. Finally, Section VII concludes the paper.

II. Preliminaries

This section introduces the signcryption algorithm that will be used subsequently.

A. Algorithm Description

The signcryption algorithms used in this paper are described as follows.

1). Initialization *(Initial)*

Given an elliptic curve E, consider a cycle group G on E with order q and its generator P. Moreover, a random number msk is chosen as the master secret key and computed as the master public key $mpk = msk \cdot P$. Then, choose three secure hash functions: $H_1 : \{0,1\}^* \times G \times G \times G \rightarrow Z_q{}^*$, $H_2 : G \times \{0,1\}^* \rightarrow Z_q{}^*$, $H_3 : G \times G \rightarrow \{0,1\}^*$. Finally, publish the system parameters $\{G, q, P, mpk, H_1, H_2, H_3\}$.

2). Pseudonym Generation *(PGen)*

Input the master secret key msk and the real identity RID, generate the expiration time of the pseudonymous identity EXP, compute the pseudonymous identity $PID = RID \oplus H(msk, EXP)||EXP$, output (PID).

3). Key Generation *(KGen)*

Input the master private key msk, the user's pseudonymous identity PID, and the master public key mpk. Select another random number $y \in Z_q$ and compute $Y = y \cdot P$. Then, compute $sk = y + msk \cdot H_1(ID, Y, mpk)$, and output the private key sk and public key $pk = Y$.

4). Signing *(Sign)*

Input the sender's private key sk and the message msg, select a random number $s \in Z_q{}^*$, compute $S_1 = s \cdot P$, $S_3 = H_2(S_1, msg)$, $S_2 = s/(sk + S_3)$, output the signature $\delta = (S_1, S_2)$.

5). Verifying *(Verify)*

Input the message msg, the sender's pseudonymous identity ID and public key pk and the signature $\delta = (S_1, S_2)$, compute $S_3{}' = H_2(S_1, msg)$, verify $S_1 = S_2 \cdot (pk + H_1(ID, pk, mpk) \cdot mpk + S_3{}' \cdot P)$. If it is, output true.

6). Signcryption *(Signcrypt)*

Input the sender's private key sk_s and the message msg, as well as the receiver's public key pk_r and pseudonymous identity ID_r, select a random number $r \in Z_q{}^*$, compute $S_1 = r \cdot P$, $v_1 = r \cdot (pk_r + H_1(ID_r, pk_r, mpk) \cdot mpk)$, $S_3 = H_2(S_1, msg) + H_2(v_1, ID_r)$, $S_2 = r/(sk_s + S_3)$, $c = H_3(v_1) \oplus msg$, $\delta = (S_1, S_2)$. Finally, output the signcryption result (δ, c).

7). Unsigncryption *(Unsigncrypt)*

Input the signcryption result $(\delta = (S_1, S_2), c)$, the receiver's private key sk_r, compute $v_1{}' = sk_r \cdot S_1$, $msg' = H_3(v_1{}') \oplus c$, $S_3{}' = H_2(S_1, msg') + H_2(v_1{}', ID_r)$, parse the sender's pseudonymous identity ID_s from msg' and search the corresponding public key pk_s, and verify $S_1 = S_2 \cdot (pk_s + H_1(ID_s, pk_s, mpk) \cdot mpk + S_3{}' \cdot P)$. If it is, output true.

B. Trust evaluation mechanism

In 2020, Yao et al [22] proposed a Trust-Based Access Control (TBAC), an access control model based on user behavioral trust. The model calculates the user's behavioral trust by analysing the deviation between the user's historical behavior and current behavior, and dynamically allocates access rights based on this trust. Permissions are assigned based on the user's trust level and the trust threshold of the resource. For example, when the user's behavioral trust level is higher than a certain threshold, the user can access the resource and perform the corresponding operation. Recently, Wang et al [28] extended and improved this idea and proposed a more complete DR-TBAC (Dynamic Rule TBAC) system. This system uses an LSTM model to calculate the user's trust level and integrates reinforcement learning, especially the Deep Q-Network (DQN) algorithm, to optimize the trust threshold and adjust the access control policy according to the dynamic changes of user behavior and environmental factors. We draw on their ideas to define T_1,T_2 according to the following steps:

1) ***Evaluate Historical Behavior:*** Define the trust thresholds based on historical data of user or device interactions. For example, how frequently the user has accessed services, whether they have exhibited any anomalous behavior, and how they have responded to previous authentication challenges.

2) ***Contextual Factors :*** Use contextual information such as the time of access, the sensitivity of the requested resources, and environmental variables (e.g., device type, location) to influence how T_1 and T_2 thresholds are applied. For instance, accessing high-risk services may require a higher trust threshold.

3) ***Continuous Monitoring:*** Trust should be dynamic and continuously updated based on real-time behavior and interactions. This allows for automatic adaptation of the trust thresholds as users or devices exhibit different behavior over time. This real-time adjustment should influence when users fall into the range of T_2 or T_1.

4) ***Behavioral Risk Assessment :*** Introduce risk-based models that calculate trust based on behavioral deviations from normal activity. If there are any sudden changes in behavior (e.g., accessing from an unusual location or at an odd time), the system might lower the trust threshold or require more rigorous authentication even for trusted users or devices.

5) ***Threshold Calibration:*** Regularly review and adjust T_1 and T_2 thresholds based on system performance and any evolving security risks or changes in usage patterns. For example, if a new type of threat is discovered, you might want to lower T_1 and T_2 temporarily until users' trustworthiness can be evaluated again.

In our proposed scheme, we define two trust thresholds, T_1 and T_2 (where $T_1 > T_2$), that determine the appropriate authentication level for devices. These thresholds are dynamically calculated based on historical device behavior, contextual access patterns, risk assessment, and current security posture. The trust value ranges from 0 to 100, with values below T_2 (typically set around 60) requiring strong authentication, values between T_2 and T_1 (typically 85) triggering moderate authentication, and values above T_1 permitting weak authentication. This adaptive approach balances security rigor with authentication efficiency, reducing overhead for consistently trustworthy devices while maintaining zero trust principles for potentially risky interactions. These trust thresholds are continuously recalibrated in response to evolving threat intelligence and network security conditions.

III. System Model and Security Objectives

In this section, we first present the system model, and then introduce the security objectives.

A. System Model

As shown in Fig. 1, we propose a zero-trust access authentication architecture for edge services based on the 3GPP network. Our approach adapts the NIST zero-trust architecture concepts to the 3GPP 4G/5G network environment by mapping the Policy Decision Point (PDP) and Policy Enforcement Point

(PEP) roles to appropriate 3GPP network functions. In our architecture, devices initially connect to network components acting as the PEP, which facilitate authentication through components serving as the PDP.

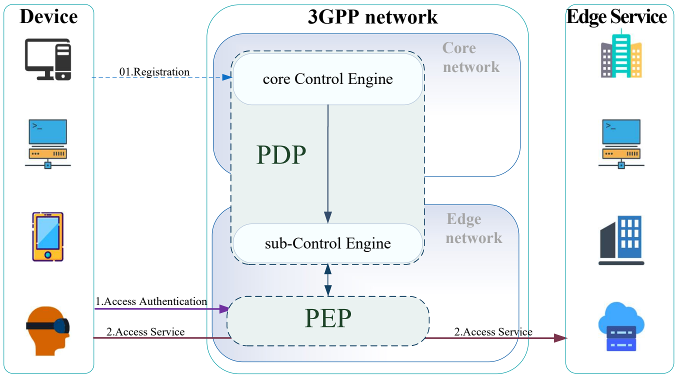


Fig. 1. System Model.

PDP is responsible for making access control decisions based on policy rules and contextual information. To more flexibly and quickly respond to device authentication requests, the PDP consists of a core Control Engine (cCE) and a sub-Control Engine (sCE).

- **A core Control Engine (cCE)**, operating at the network core level, responsible for performing strict and strong authentication of devices, maintaining identity management functions, and establishing security policies. For example, this corresponds to the Authentication Server Function (AUSF) in the 3GPP 5G network.
- **A sub-Control Engine (sCE)**, deployed within the edge network segment, handling localized authentication decisions and enabling efficient verification for time-sensitive edge applications, this corresponds to the Access and Mobility Management Function(AMF) in the 3GPP 5G network.

PEP components enforce access control decisions made by the decision-making framework, typically allowing or denying access requests based on authentication outcomes and policy evaluation. In the 3GPP edge computing architecture, the roles of Policy Enforcement Point (PEP) are primarily undertaken by the User Plane Function (UPF) and, in the context of network access security, potentially the Session Management Function (SMF).

Device includes personal devices used by employees, BYOD, or systems belonging to visitors or contracted service providers who need network access. In a ZTA, no device is inherently trusted, regardless of its physical or network location. Every device must authenticate itself before connecting to an enterprise-owned resource.

Edge Service refers to various computing, storage, analysis, and management services provided in an edge computing environment, which support the operation of applications and data processing on edge devices or edge servers. Edge computing platforms and services are provided by cloud service providers such as Microsoft Azure.

This mapping allows us to implement zero-trust principles while leveraging the existing 3GPP network infrastructure. Upon successful authentication through this framework, the devices are granted access to edge services provided by service providers, such as Microsoft Azure, which offers edge computing services including content distribution and caching, game streaming, and vehicle-to-everything (V2X) sensing services.

B. Adversarial Model

In this section, the most adopted and accepted Dolev-Yao (DY) adversarial model is considered the basic adversary model used to analyze the security of the proposed scheme. In the DY adversarial model, the adversary A can control the entire communication network. Concretely, A can eavesdrop, tamper, or even replay communication data between the device and the PEP, A can impersonate a legitimate device accessing the network to gain unauthorized access. A can use a Man-In-The-Middle (MITM) attack to eavesdrop on the interactive data between devices and PEPs. A can deplete the resources of the 3GPP network, such as sending forged data, replaying data, sending useless data, etc., causing them to refuse to provide services. Additionally, since PEP, sCE, and cCE all reside within the 3GPP network where their inter-communication is considered secure, ensuring the security of communication data between devices and PEPs becomes critically vital.

C. Security Objectives

The proposed scheme should satisfy the following security objectives.

1. Authentication and key agreement

In the authentication process, authentication and key agreement mechanism is required between the device and the sCE. The authentication means the device and the sCE should be able to identify if the peer who communicates with him is a legitimate user. Meanwhile, the key agreement can be used to guarantee the security of subsequent communications.

2. Privacy preserving

In our system, user anonymity, viewed as an important security property should be guaranteed, where users' identities cannot be publicly transmitted on the communication channel and only the legitimate sCE can recover the users' identities.

3. Perfect Forward/Backward Secrecy

Perfect Forward/Backward Secrecy (PFS/PBS) should be ensured, so that the adversary who obtains the private key of the device or the sCE will have no way to know the following and preceding communications.

4. Data Confidentiality and Integrity

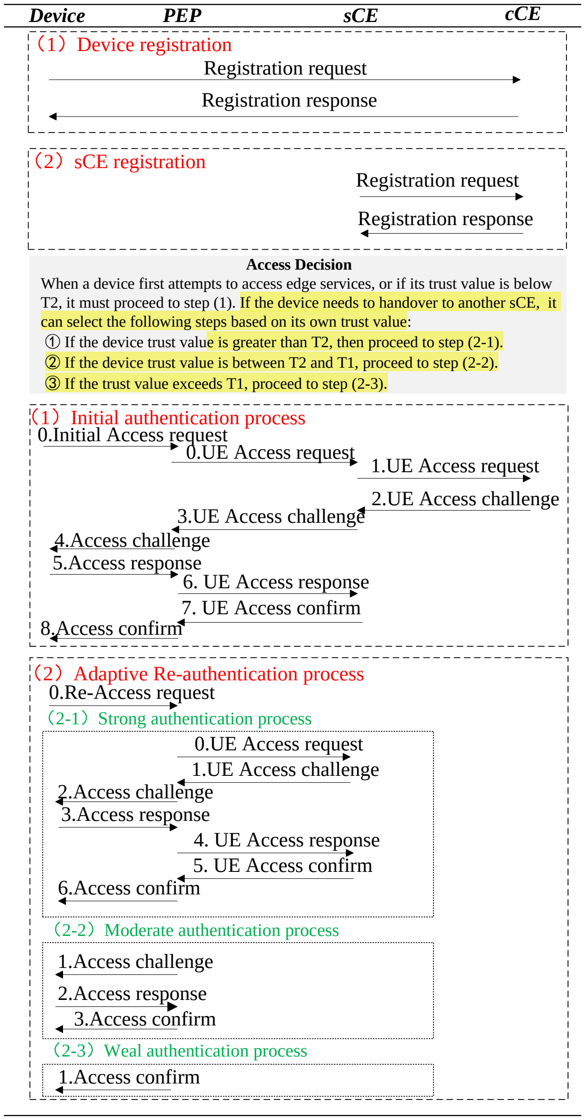
The proposed scheme should ensure that all sensitive information exchanged between network entities remains protected from unauthorized access and tampering. Cryptographic mechanisms must be employed to prevent eavesdropping and detect any modifications to messages during transmission.

5. Unforgeability, Undeniability and Untraceability

The scheme should guarantee that adversaries cannot generate valid authentication messages without legitimate keys; entities cannot deny their participation in the authentication process; and observers cannot track or correlate multiple sessions to the same device, thereby preserving user privacy and preventing tracking.

IV. The Proposed Scheme

In this section, we give a detailed description of the proposed scheme, which mainly consists of two processes: the initial authentication process and the adaptive re-authentication process. The adaptive re-authentication process includes three processes: strong authentication, moderate authentication, and weak authentication.



Access Decision

When a device first attempts to access edge services, or if its trust value is below T2, it must proceed to step (1). If the device needs to handover to another sCE, it can select the following steps based on its own trust value:

① If the device trust value is greater than T2, then proceed to step (2-1).

② If the device trust value is between T2 and T1, proceed to step (2-2).

③ If the trust value exceeds T1, proceed to step (2-3).

Fig. 2. The overview.

An overview of the zero-trust-based edge service access authentication architecture is shown in Fig. 2. Initially, each device and the sCE engage in a registration process to register the cCE for accessing edge services. When a device first attempts to access edge services, or if its trust value is below T_2, it must undergo an initial authentication process and obtain the anonymous identifiers, key parameters, etc., used for subsequent authentication. Subsequently, if the device needs to handover to another sCE and its trust value is more than T_2, the device undergoes a strong authentication process. If the device's trust value is between T_2 and T_1, a moderate authentication process is required. If the trust value exceeds T_1, the device undergoes a weak authentication process. Following the completion of the appropriate authentication process, the device gains access to the edge services.

A. Edge service registration Process

In this process, each device connects to the 3GPP network by executing the 5G-AKA process. For devices intending to access edge services, the 3GPP network assists in establishing a secure channel between the device and the cCE using the GBA/AKMA mechanism. Subsequently, the device and the cCE complete the registration process.

Initially, the cCE should perform the *Initial( )* algorithm to generate the master secret key msk and master public key mpk, and public these parameters $\{G, q, P, mpk, H_1, H_2, H_3, E_k(), D_k()\}$.

1) ***Device registration:***

a. The user of a device i chooses an identity ID_i and a password PW_i, and imprints biometrics B_i on the sensor of the device.

b. The device picks a random value $x_i \in Z_q$, computes $X_i = x_i \cdot P$, $A_i = H(ID_i, PW_i, B_i)$, $B_i = H(ID_i, PW_i, X_i)$, transmits an edge service registration request message (ID_i, X_i, A_i, B_i) to the cCE.

c. The cCE generates a 256 bits value K_i. Then, the cCE invokes the algorithm$PGen(ID_i)$ to generate Device's pseudonymous identityPID_i and computes $HA_i = A_i \oplus K_i$. Finally, the cCE transmits an edge service registration response message (PID_i, HA_i) to the device.

d. The device stores (ID_i, PID_i, HA_i).

2) ***sCE registration:***

The sCE transmits an edge service registration request message (ID_{sCE}) to the cCE. The cCE invokes the algorithm $KGen(msk, mpk, ID_{sCE})$ to get the sCE's private key sk_{sCE} and public key pk_{sCE}. Finally, the cCE transmits (sk_{sCE}, pk_{sCE}) to the sCE securely.

Additionally, the cCE generates a group key GK_{PEP} for all PEPs. Finally, the cCE transmits GK_{PEP} to the PEPs securely.

B. Initial Authentication Process

When first attempts to access edge services, or if its trust value is below T_2, each device performs the initial access authentication process with cCE to securely obtain the relevant keys for subsequent adaptive authentication. The specific details are as follows:

a. The device i generates a random number $r_i \in Z_p$ and computes $R_i = r_i{*}P$, $enckey_i || mackey_i = h(r_i{*}mpk)$, ciphertext $c_i = E_{enckey_i}(ID_i)$, message authentication code $mac_i = h(mackey_i, c_i)$. Finally, each device transmits an initial request message including (R_i, c_i, mac_i) to the PEP.

b. The PEP verifies the message type and forwards (R_i, c_i, mac_i) to the sCE.

c. The sCE forwards (R_i, c_i, mac_i) together with its identity ID_{sCE} to the cCE.

d. The cCE computes $enckey_i||mackey_i = h(R_i * msk)$, verifies message authentication code $mac_i = h(mackey_i, c_i)$, and decrypts $ID_i = D_{enckey_i}(c_i)$. Then, the cCE searches the long-term secret key K_i based on ID_i and generates random value $rand_i$, computes $MAC_i = h(K_i, rand_i, R_i)$, $K_{sCE_i} = KDF(K_i, rand_i, ID_{sCE})$, $RES_i = h(K_i, rand_i)$, and $RES_i^* = h(h(K_i, rand_i), ID_{sCE})$. Additionally, the cCE invokes the algorithm $PGen(msk, ID_i)$ to get the pseudonymous identity PID_i, invokes the algorithm $KGen(msk, mpk, ID_i)$ to get the user's private key sk_i and public key pk_i. Finally, the cCE computes $TGK_i = h(GK_{PEP}, PID_i)$, $c_k = E_{enckey_i}(PID_i, sk_i, pk_i, TGK_i)$ and transmits $(rand_i, MAC_i, K_{sCE_i}, RES_i^*, c_k)$ to the sCE securely.

e. The sCE stores $(K_{sCE_i}, RES_i^*, c_k)$ and transmits $(rand_i, MAC_i)$ to the PEP.

f. The PEP forwards an access challenge including $(rand_i, MAC_i)$ to the device.

g. Upon receiving the challenge message, the device inputs its identity ID_i, password PW_i, and biometries B_i , computes $A_i = H(ID_i, PW_i, B_i)$ and $K_i = A_i \oplus HA_i$, verifies MAC_i and computes $RES_i = h(K_i, rand_i)$, transmits RES_i to the sCE. Finally, the device computes the keys $K_{sCE_i} = KDF(K_i, rand_i, ID_{sCE})$ and $K_{PEP_i} = KDF(K_i, rand_i, ID_{PEP})$, which serve as the shared secret keys for secure communication between the device and the sCE, and between the device and the PEP.

h. Concurrently, the cCE computes K_{PEP_i} and securely shares (K_{PEP_i}, TGK_i) with the PEP.

i. The PEP stores (K_{PEP_i}, TGK_i) for future communications with the device.

j. The sCE verifies $RES_i^* = h(RES_i, ID_{sCE})$. If it is, the sCE forwards RES_i to the cCE and transmits an access confirm message c_k to the device through the PEP.

Following the aforementioned procedures, the device can establish secure communication channels with both the sCE and PEP using the respective cryptographic keys,K_{sCE_i} and K_{PEP_i}. Subsequently, the device can employ the key parameters $(PID_i, sk_i, pk_i, TGK_i)$ for following authentication.

C. Adaptive Re-Authentication Process

When re-access edge services and the device's trust value exceeds T_2, each device performs the following steps.

The device i computes $mac_i = h(TGK_i, PID_i, ts_i)$ and transmits a re-access request message including (PID_i, mac_i, ts_i) to the PEP, where ts_i denotes the current timestamp. The PEP verifies ts_i and mac_i, and checks the trust value of the device. If ts_i and mac_i is correct, the PEP performs the different processes according to the trust value of the device and the request type (handover or not). Both PEP and sCE respectively broadcast their own information, comprising PEP's identity identifier ID_{PEP}, public key pk_{PEP}, and sCE's identity identifier ID_{sCE} with public key pk_{sCE}.

1) *Strong Authentication Process:*

If the device needs to handover to another sCE and its trust value is more than T_2, the device should perform the following steps.

a. The PEP forwards PID_i to the sCE.

b. The sCE invokes the algorithms $Sign(sk_{sCE}, msg)$ to output the signature (f_1, f_2), which msg denotes the sCE's necessary information for device access. Finally, the sCE sends (f_1, f_2, msg) to the PEP.

c. The PEP forwards (f_1, f_2, msg) to the device.

d. After receiving the message, the device invokes the algorithms $Verify(ID_{sCE}, msg, (f_1, f_2), pk_{sCE})$. And if the output is true, the device invokes the $signcrypt(ID_{sCE}, sk_i, MSG, pk_{sCE})$ to generate the signcryption result (f_1', f_2', c).

e. The PEP forwards (f_1', f_2', c) to the sCE.

f. After all these, the sCE invokes the algorithms $Unsigncrypt(sk_{sCE}, (f_1', f_2', c))$. If the output is true, the sCE transmits a UE access confirm message to the PEP.

g. The PEP forwards it to the device.

h. The device derives the shared cryptographic keys $K_{sCE_i}' = KDF(K_i, ts_i, ID_{sCE})$ and $K_{PEP_i}' = KDF(K_i, ts_i, ID_{PEP})$, which establish secure communication channels between the device and the newly provisioned sCE, and between the device and the PEP, respectively.

2) *Moderate Authentication Process:*

If its trust value is greater than T_2 and less than T_1, the device undergoes the following steps.

a. The PEP generates a random number $nouce$ and computes $res = h(K_{PEP}, nouce, ts_i, 1)$ and transmits an access confirm message $(res, nouce)$ to the device.

b. The device verifies res. If it is correct, the device computes $res_2 = h(K_{PEP}, nouce, ts_i, 2)$ and transmits res_2 directly accesses the edge services.

d. The PEP verifies res_2. If it is, the PEP allows the device to access edge services. Additionally, the device and the PEP compute the next temporary authentication key $TK_{PEP_i} = h(K_{PEP}, nouce, ts_i, 3)$.

3) *Weak Authentication Process:*

If the trust value exceeds T_1, the device undergoes the following steps.

a. The PEP transmits an access confirm message to the device.

b. The device directly accesses the edge services.

V. Security Analysis

In this section, we have adopted the qualitative security analysis to prove that it can satisfy various security properties and compared the security properties with the previous schemes. In addition to the qualitative safety analysis, the formal validation of Tamarin is also used.

A. Security Analysis

In this section, we analyze how our proposed zero trust access authentication scheme addresses key security requirements across the initial, strong, and moderate authentication processes. For each security property, we demonstrate how our hierarchical authentication framework and adaptive trust-based approach provide comprehensive protection while maintaining efficiency for edge services in 3GPP networks. It's important to note that the weak authentication process consists only of an access confirmation message without cryptographic operations, as it's designed for devices with trust values above the threshold T_1. Due to its simplified nature, we focus our security analysis on the initial, strong, and moderate authentication processes where cryptographic security mechanisms are employed.

1) ***Mutual authentication:*** In the initial authentication process, the device authenticates the cCE by verifying $MAC_i = h(K_i, rand_i, R_i)$, while the cCE authenticates the device by verifying $RES_i = h(K_i, rand_i)$. The sCE further verifies the device's identity by checking $RES_i^* = h(RES_i, ID_{sCE})$. During the strong authentication process, the device authenticates the sCE by verifying the sCE's signature (f_1, f_2) using the $Verify(ID_{sCE}, msg_i, (f_1, f_2), pk_{sCE})$ algorithm, confirming that only the legitimate sCE with the private key sk_{sCE} could have generated this signature. Conversely, the sCE authenticates the device by verifying the signcryption result (f_1', f_2', c) using the $Unsigncrypt(sk_{sCE}, (f_1', f_2', c))$ algorithm, which confirms the device possesses the correct private key sk_i. In the moderate authentication process, mutual authentication is achieved through the challenge-response mechanism: the PEP verifies the device by checking $res_2 = h(K_{PEP}, nonce, 2)$, while the device authenticates the PEP by verifying $res = h(K_{PEP}, nonce, 1)$. These cryptographic verification processes ensure that each entity can authenticate its communication partner in all processes, preventing impersonation attacks.

2) ***Key agreement:*** In the initial authentication process, the device and the sCE establish a shared key K_{sCE_i} while the device and PEP establish K_{PEP_i}. These keys are derived from the long-term secret K_i, the random challenge $rand_i$, and the respective identity information, creating unique session keys for secure communication. During the strong authentication process, when the device needs to handover to another sCE, the device and the new sCE compute a fresh session key K'_{sCE_i}, and the device establishes a shared key $K'_{PEP_i}$with the PEP, incorporating the current timestamp ts_i to ensure key freshness and prevent replay attacks. In the moderate authentication process, the device and PEP establish TK_{PEP_i} as a new shared key derived from the previous key and session-specific random values. This hierarchical key agreement approach optimizes authentication overhead while maintaining security across all processes. All key derivation procedures are protected by the underlying cryptographic operations, making it computationally infeasible for adversaries to derive the session keys without possessing the private keys of the legitimate entities.

3) ***Data Confidentiality*** *:* After the initial authentication process is over, the long-term shared keys K_{SCE_i} and K_{PEP_i} protect all subsequent communications between the device and sCE and PEP, respectively. After strong authentication, we have a newly calculated key K'_{SCE_i} to maintain confidentiality between the device and the new sCE. Similarly, after moderate authentication, the newly derived key TK_{PEP_i} ensures continued confidentiality between the device and PEP. These cryptographic mechanisms ensure that messages remain difficult for adversaries to decrypt.

4) ***Data Integrity*** *:* After the initial authentication process, the Device uses its private key sk_i to sign messages sent to the sCE, and the sCE uses the Device's public key pk_i to verify the signature, ensuring integrity. When the sCE sends messages to the Device, hash functions using the shared keys K_{SCE_i} and K_{PEP_i} protect message integrity. During strong authentication, the signcryption algorithm provides both confidentiality and integrity through signatures (f_1', f_2'). In moderate authentication, the integrity of communications is protected by hash values res and res_2. If any of these messages are tampered with during transmission, the verification will fail, allowing immediate detection of integrity violations.

5) ***Unforgeability*** *:* In the strong authentication process, the device's request message includes a signature generated using the device's private key sk_i. The sCE uses the device's public key pk_i to verify this signature. Since the private key is computationally unfeasible to derive from the public key, only authenticated devices can generate the correct signature. This prevents adversaries from forging legitimate device signatures. In the response message, the message content encrypted in ciphertext c is used to generate hash values for verification. During moderate authentication, the values res and res_2 are derived using the shared keyK_{PEP} and a random nonce, which adversaries cannot forge without knowledge of K_{PEP}. This comprehensive approach ensures that all communications within our scheme are protected against forgery attempts.

6) ***Undeniability:*** The proposed scheme ensures undeniability through multiple mechanisms across authentication processes. During the initial authentication process, the device's identity information is securely bound to its cryptographic parameters (PID_i, sk_i, pk_i). In the strong authentication process, the signcryption algorithm inherently provides non-repudiation since it uses the sender's private key sk_i to generate signatures that only the legitimate device could create. These signed messages cannot be denied later because they are cryptographically linked to the device's private key. Additionally, all authentication sessions incorporate timestamps (ts_i) and random numbers $(rand_i, nouce)$ that uniquely bind messages to specific sessions, preventing replay-based repudiation attempts. For moderate authentication, the response values res_2 provide cryptographic proof of the device's participation. This comprehensive approach ensures that neither the device nor the network entities can plausibly deny their participation in the authentication process.

7) ***Untraceability:*** In the initial authentication process, the device's identity ID_i is protected through encryption as $c_i = Eenckey_i(ID_i)$, making it impossible for adversaries to track user identities. During the strong authentication process, when the device generates the signcryption result (f_1', f_2', c), the random number r creates unique values for each session, preventing correlation between sessions. In the moderate authentication process, the use of challenge-response with random $nonce$ further ensures that each authentication attempt produces different values. Throughout all authentication processes, session-specific random elements (r_i in initial authentication, r in strong authentication, and nonce in moderate authentication) ensure that adversaries cannot link multiple sessions to the same device. As a result, our scheme achieves untraceability across all communication processes.

*8) **Privacy preserving**:* In the initial authentication process, the device's real identity ID_i is encrypted as $c_i = Eenckey_i(ID_i)$ using the derived $enckey_i$, ensuring that only the cCE with the master secret key msk can decrypt and obtain the device's identity. During the strong authentication process, the device's pseudonymous identity PID_i is used instead of its real identity, and sensitive messages are protected using signcryption (f_1', f_2', c), where only the intended sCE with the private key sksCE can decrypt the content. In the moderate authentication process, the device is identified only by its previously established shared key with the PEP, without revealing any identity information over the communication channel. The hierarchical trust-based architecture ensures that identity information is only shared on a need-to-know basis, with the cCE functioning as the trusted identity manager while the sCE and PEP operate with pseudonymous identities. Therefore, even if adversaries intercept the communication, they cannot obtain the device's private information across any authentication process.

*9) **Perfect Forward/Backward Secrecy (PFS/PBS)**:* In the initial authentication process, the device generates a random number $r \in Z_q{}^*$ which is used to compute $R_i = r_i * P$ and derive the encryption key $enckey_i || mackey_i = h(r_i * mpk)$. Even if an attacker compromises the long-term key K_i in the future, they cannot recover the session keys K_{sCE_i} and K_{PEP_i} without knowing the random value r_i used in that specific session. During the strong authentication process, the device uses a fresh random number r when generating the signcryption result (f_1', f_2', c), and the newly derived key K'_{sCE_i} incorporates the current timestamp ts_i, ensuring independence between sessions. In the moderate authentication process, the newly established key TK_{PEP_i} is derived using a session-specific random nonce, maintaining forward and backward secrecy. The usage of ephemeral random values $(r_i, r, nonce)$ across all authentication processes ensures that even if an adversary compromises the current session key, previous and future session keys remain secure. Additionally, the underlying ECC-based operations provide perfect forward secrecy, ensuring that even if the long-term keys are leaked, the adversary cannot recover previously established session keys.

Withstanding several protocol attacks : The proposed scheme can withstand several protocol attacks as follows.

1) **Replay attack**: Our proposed scheme resists replay attacks across all authentication processes through the use of unique session parameters. In the initial authentication process, the cCE generates a random challenge $rand_i$ that uniquely binds authentication messages to a specific session. During strong authentication, the device includes timestamps (ts_i) when computing $mac_i = h(TGK_i, PID_i, ts_i)$, allowing the PEP to verify message freshness. In moderate authentication, the PEP generates a random $nonce$ value that prevents the reuse of authentication messages. The verification procedures in each process reject any replayed messages because timestamps would be outdated, and random challenges wouldn't match the expected values for the current session. Additionally, the unique key derivation process ensures that session keys are specific to each authentication attempt, further mitigating the risk of replay attacks.
2) **MitM attack**: Our scheme provides comprehensive protection against Man-in-the-Middle (MitM) attacks across all authentication processes. In the initial authentication process, the device and the cCE establish shared keys K_{sCE_i} and K_{PEP_i} using ECC-based operations similar to ECDH, where the secret key K_i is securely derived and never transmitted in plaintext. During strong authentication, the device verifies the sCE's signature (f_1, f_2) generated with the sCE's private key sk_{sCE}, while the sCE authenticates the device through the signcryption result (f_1', f_2', c), which can only be generated with knowledge of the device's private key ski. In moderate authentication, the challenge-response mechanism using values $res = h(K_{PEP}, nonce, 1)$ and $res_2 = h(K_{PEP}, nonce, 2)$ ensures both entities possess the shared secret K_{PEP}. Since these cryptographic operations require possession of the respective private keys and pre-established secrets, a MitM attacker cannot intercept and modify communications without detection, as they cannot generate valid signatures or responses without the corresponding private keys.
3) **Impersonation attack**: Our scheme provides robust protection against impersonation attacks across all authentication processes. In the initial authentication process, the device authenticates using its long-term secret K_i, which is never transmitted in clear and is verified through $RES_i = h(K_i, rand_i)$. An attacker cannot impersonate the device without knowing K_i, which is securely stored and used to derive the authentication parameters. During strong authentication, impersonating the sCE is prevented as the adversary would need the private key sk_{sCE} to generate valid signatures (f_1, f_2). Similarly, device impersonation is thwarted as the attacker would require the device's private key sk_i to produce valid $Signcryption$ results (f_1', f_2', c). The sCE verifies these values through the $Unsigncrypt$ algorithm, which would fail with forged credentials. In moderate authentication, the challenge-response mechanism using the shared key K_{PEP} ensures that only legitimate entities possessing the correct key can generate valid responses $res_2 = h(K_{PEP}, nonce, 2)$. The cryptographic binding of identities to their respective keys throughout all authentication processes makes successful impersonation computationally infeasible without compromising the secure key storage of legitimate entities.
4) **Passive attack**: Our scheme provides comprehensive protection against passive attacks across all authentication processes through strong encryption and secure key management. In the initial authentication process, sensitive information including the device's real identity

ID_i is encrypted as $c_i = Eenckey_i(ID_i)$, preventing eavesdroppers from obtaining authentication credentials or identity information. During strong authentication, all communications between the device and the sCE utilize signcryption, where messages are secured via (f_1', f_2', c) which provides both confidentiality and integrity. The authentication data is encrypted using the sCE's public key, ensuring only the intended recipient with the corresponding private key sk_{sCE} can decrypt the contents. In moderate authentication, the challenge-response values res and res_2 are generated using the shared key K_{PEP}, which is never transmitted in plaintext across the network. After authentication, all subsequent communications are protected by the established session keys K_{sCE_i}, K_{sCE_i}', or TK_{PEP_i} depending on the authentication process, ensuring that passive attackers cannot extract sensitive information even through prolonged network monitoring. This layered cryptographic approach ensures no valuable information leaks to passive attackers throughout the entire authentication process.

B. Formal Verification: Tamarin

In this section, we formally analyze the proposed schemes by using the automatic verification tool named Tamarin, which can precisely analyze the secrecy and complex authentication properties of various protocols. Tamarin tool supports the equational specification of some cryptographic operators, such as Diffie-Hellman exponentiation and bilinear pairings. The Tamarin simulation results of the proposed schemes are shown as follows.

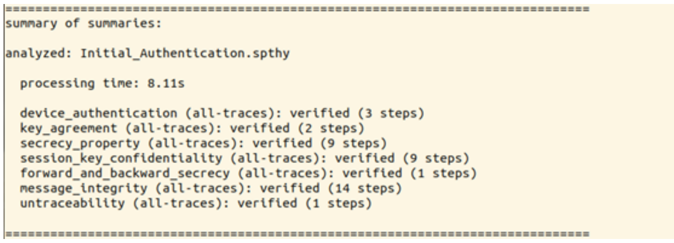
The running results of all proposed protocols are shown in Figs. 3, 4, and 5.

Fig. 3 shows the Tamarin execution result of the initial authentication. From Fig. 3, the output of all lemmas is displayed as $verified$. It means that this scheme achieves device authentication, key agreement, secrecy property, session key confidentiality, forward and backward secrecy, message integrity, and untraceability.

Fig. 4 shows the Tamarin execution result of the strong authentication. From Fig. 4, the output of all lemmas is displayed as $verified$. It means that this scheme achieves confidentiality, mutual authentication, protocol correctness, perfect forward and backward secrecy, message integrity, accountability, privacy preservation, and secure key establishment.

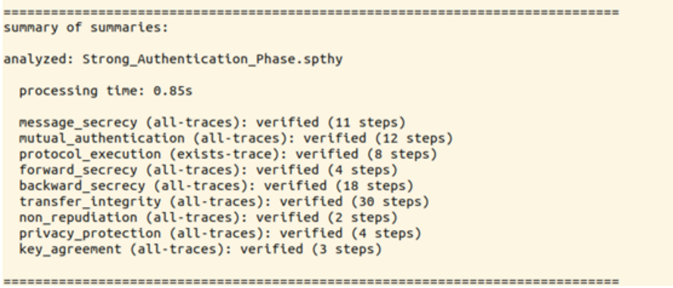
Fig. 5 shows the Tamarin execution result of the strong authentication. From Fig. 5, the output of all lemmas is displayed as $verified$. The scheme achieves mutual authentication, key agreement, message integrity, session key security, privacy protection, and replay protection. These properties were verified in varying steps, ranging from 2 to 8, providing formal proof of the scheme's security claims. Collectively, these attributes ensure mutual authentication between the device and the PEP, secure key establishment, message integrity, confidentiality of session keys, privacy preservation, and protection against replay attacks.

In addition, there is no validation here since weak authentication is just a confirm message, and since by the time the device performs weak authentication, it has already completed the strong and moderate authentication processes.



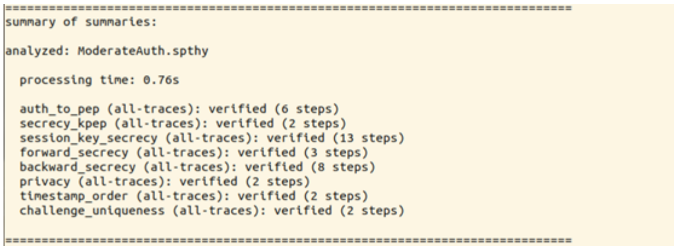
```
==============================================================================
summary of summaries:

analyzed: Initial_Authentication.spthy

  processing time: 8.11s

  device_authentication (all-traces): verified (3 steps)
  key_agreement (all-traces): verified (2 steps)
  secrecy_property (all-traces): verified (9 steps)
  session_key_confidentiality (all-traces): verified (9 steps)
  forward_and_backward_secrecy (all-traces): verified (1 steps)
  message_integrity (all-traces): verified (14 steps)
  untraceability (all-traces): verified (1 steps)

==============================================================================
```

Fig. 3. Tamarin result of the Initial Authentication.



```
==============================================================================
summary of summaries:

analyzed: Strong_Authentication_Phase.spthy

  processing time: 0.85s

  message_secrecy (all-traces): verified (11 steps)
  mutual_authentication (all-traces): verified (12 steps)
  protocol_execution (exists-trace): verified (8 steps)
  forward_secrecy (all-traces): verified (4 steps)
  backward_secrecy (all-traces): verified (18 steps)
  transfer_integrity (all-traces): verified (30 steps)
  non_repudiation (all-traces): verified (2 steps)
  privacy_protection (all-traces): verified (4 steps)
  key_agreement (all-traces): verified (3 steps)

==============================================================================
```

Fig. 4. Tamarin result of the Strong Authentication.



```
==============================================================================
summary of summaries:

analyzed: ModerateAuth.spthy

  processing time: 0.76s

  auth_to_pep (all-traces): verified (6 steps)
  secrecy_kpep (all-traces): verified (2 steps)
  session_key_secrecy (all-traces): verified (13 steps)
  forward_secrecy (all-traces): verified (3 steps)
  backward_secrecy (all-traces): verified (8 steps)
  privacy (all-traces): verified (2 steps)
  timestamp_order (all-traces): verified (2 steps)
  challenge_uniqueness (all-traces): verified (2 steps)

==============================================================================
```

Fig. 5. Tamarin result of the Moderate Authentication.

VI. Performance Evaluation

Communication overhead and computational overhead are considered as the most important metrics of efficiency in the authentication process. In this section, we compare the communication and computational overheads of our scheme with the other several similar schemes: Dawei Li's Blockchain Scheme [24], Belal Ali's Dual Fuzzy Scheme [25] and BSeIn Blockchain Scheme [26]. Considering that the initial authentication is generally done less frequently, we only compare the strong, moderate and weak authentication process in the adaptive authentication process. To achieve the same security level with AES 128 bits, we suppose that the key size for algorithms based on ECC is 256 bits, the key size for algorithms based on integer-factorization cryptography such as RSA is 3072 bits, and for the parameters based on the finite-field cryptography, the size of the public key is 3072 bits and the size of the private key is 256 bits [23]. In addition, the output length of the hash is 128 bits, the size of the random number is 128 bits, and the size of the timestamp is 32 bits [27], respectively. Since all existing schemes need to transfer the necessary information, such as the identity information, we assume that the length of the necessary information is 320 bits.

A. Communication overhead

In this part, comparisons between schemes are made based on the size of each message. In our strong authentication process, four messages need to be exchanged between device, PEP, and sCE. In our moderate authentication process, they only need to exchange two messages: res and res_2. In our weak authentication process, there is only one acknowledgment message. Under this circumstance, we can analyze the communication overhead caused by these existing schemes by calculating the total size of the messages. For the sake of fairness, for Li et al's scheme [24], we assume that the communication overhead of Msg4 includes the communication overhead of identity update and revocation. For Belal Ali et al's scheme [25], we summarize the authentication requests, multi-factor authentication and authentication result messages as Msg1, and combine the edge server trust evaluation requests, trust value calculation results and blockchain records as Msg2. For Lin et al's scheme [26], Msg1 represents the initialization stage of the entire authentication process. On basis of the sizes of all the messages, we make a comparison of the communication overheads of these existing schemes as shown in Table I and Fig. 6.

TABLE I
THE OVERHEAD OF COMMUNICATION.

(byte)	MSG_1	MSG_2	MSG_3	MSG_4	Total
Li et al's scheme [24]	52	32	48	48	180
Ali et al's scheme [25]	64	40	32	16	152
Lin et al's scheme [26]	160	102	64	64	390
Strong-Authentication	32	104	104	4	244
Mod-Authentication	32	32	16	0	80
Weak-Authentication	32	0	0	4	36

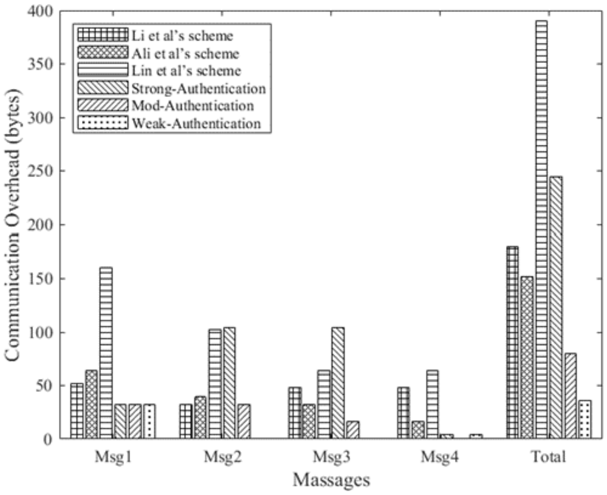


Fig. 6. The overhead of communication.

According to the Fig. 6, in the scheme we proposed, the overhead of strong authentication is slightly larger than that of other schemes, while the calculation results of moderate authentication and weak authentication are much smaller than those of other schemes.

B. Computational overhead

In this part, comparisons will be made based on the encryption operations carried out in each scheme. The computational overhead of the primitive cryptography operations is measured by using C/C++ OPENSSL library [30]tested on an Intel(R) Core(TM) m3-6Y30 CPU 0.9 GHz processor as a device and an Intel(R) Core(TM) i5-7500 CPU 3.40 GHz as an sCE as shown in Table II. Since the elliptic curve point multiplication, symmetric encryption/decryption and hash operations dominate the computational overhead in the proposed scheme, here we mainly consider these operations.

Table III shows the analysis results on the computational overheads of different entities in our scheme, where T_{Dev} represents the computational overhead of the Device, T_{PEP} represents the computational overhead of the PEP, T_{cCE} represents the computational overhead of the cCE, and T_{sCE} represents the computational overhead of the sCE.

TABLE II
COMPUTATIONAL OVERHEAD OF THE CRYPTOGRAPHY OPERATIONS.

(ms)	Symbol	Device	sCE
Point Multiplication	TM	0.960	0.366
Symmetric Encryption	TE	0.386	0.125
Hash Operation	TH	0.032	0.018

According to the numerical results, it can be clearly seen that among all the schemes, the computational overhead of device is much higher than that of sCE, cCE and PEP, because it performs the most point multiplication operations, which dominates the total cost. Due to their key management and verification responsibilities, cCE and sCE have moderate computing requirements. The PEP mainly processes message forwarding with the minimum computational burden. Due to the dominant position of device in the total cost, we compared the overhead of device in each scheme un-

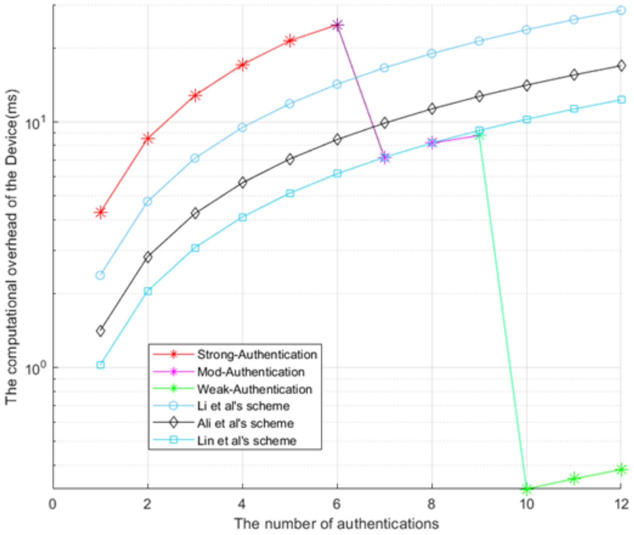


Fig. 7. The overhead of Device

TABLE III
COMPUTATIONAL OVERHEAD.

(ms)	T_{Dev}	T_{PEP}	T_{cCE}	T_{sCE}
Li et al's scheme [24]	2TM + 1TE + 2TH = 2.370	1TM + 3TH=0.420	0	1TM + 3TH = 0.420
Ali et al's scheme [25]	1TM + 1TE + 2TH = 1.410	1TM + 3TH=0.420	0	1TM + 3TH = 0.420
Lin et al's scheme [26]	TM+2TH= 1.024	TM + 2TH =0.430	TM + TH =0.398	TM + 2TH =0.430
Strong-Authentication	4TM+TE+3TH=4.274	2TH=0.036	3TM+2TE+4TH=1.834	2TM+2TH=0.768
Mod-Authentication	1TM + 2TH =1.024	2TH=0.036	0	0
Weak-Authentication	1TH = 0.032	0	0	0

der different authentication times, as shown in Fig. 7. Fig. 7 illustrates the computational overhead advantage of our adaptive authentication approach over multiple authentication cycles. Initially, all schemes exhibit high overhead due to strong authentication requirements. However, as the number of authentication attempts increases, our proposed scheme demonstrates significant overhead reduction when devices transition from strong authentication (trust value is more than T_2) to moderate authentication (trust value is between T_2 and T_1), and eventually to weak authentication (trust value exceeds T_1). Compared with the existing static schemes, this adaptive behavior can significantly reduce the computational overhead while maintaining an appropriate security level based on the credibility of the device. The results validate that our trust-based approach effectively optimizes resource utilization in edge computing environments without compromising security.

The analysis demonstrates that our scheme realizes secure authentication and key agreement with acceptable computational overhead for resource-constrained devices in edge computing scenarios.

C. Performance with malicious Devices

Although we have demonstrated that our scheme can withstand several known attacks through security analysis, there may still be unknown or uncertain attacks that we cannot determine if our proposed scheme can resist. In the authentication process, if malicious devices send multiple malicious request messages to the sCE, it will consume considerable computational overhead for them to handle these malicious requests. In this section, we analyze how much computational overhead the sCE or PEP need to consume to identify a malicious device. Table IV shows the comparison results of the computational cost on sCE under malicious devices, where T'_{mal} represents the required computational overhead for sCE to detect a malicious device.

TABLE IV
COMPARISON OF SCE'S COMPUTATIONAL OVERHEAD UNDER MALICIOUS DEVICES.

(ms)	T'_{mal}
Li et al's scheme [24]	1TH + 2TM =0.750
Ali et al's scheme [25]	3TH + 1TM =0.420
Lin et al's scheme [26]	2TM+3TH=0.786
Strong-Authentication	TH + 3TM= 1.099
Mod-Authentication	2TH =0.036

The results show that our scheme has reasonable computational overhead on sCE or PEP for identifying malicious devices. Although this overhead is slightly higher than some existing schemes that do not provide privacy preservation, it is an acceptable tradeoff for achieving better security properties, including privacy protection. Furthermore, the verification can be performed simultaneously to improve efficiency when handling multiple malicious requests.

VII. CONCLUSION

This paper presented a novel zero trust access authentication scheme for edge services in 3GPP networks that addresses the limitations of traditional perimeter-based security approaches. Our hierarchical framework offers multiple authentication methods to achieve a flexible authentication mechanism. By dynamically adjusting the device access authentication protocol based on the trust value, it is possible to effectively reduce the computing and communication overhead while ensuring the security of the device. This adaptive authentication strategy not only enhances the security of the system, but also optimizes the utilization of resources, adapts to the demands of different scenarios, and strengthens the overall performance and reliability of edge services. Security analysis and formal verification using Tamarin demonstrate that the proposed scheme satisfies critical security requirements including mutual authentication, key agreement, privacy preservation, and perfect forward/backward secrecy. Performance evaluations show that our adaptive approach significantly reduces communication and computational overhead for trustworthy devices while maintaining robust security. The scheme's three-tiered authentication levels (strong, moderate, and weak) enable efficient resource allocation without compromising security. Future research should focus on optimizing trust evaluation mechanisms, enhancing resilience against emerging threats, and extending the framework to other edge computing paradigms beyond 3GPP networks.

REFERENCES

[1] Mao Y, You C, Zhang J, et al. A survey on mobile edge computing: The communication perspective[J]. IEEE communications surveys & tutorials, 2017, 19(4): 2322-2358.

[2] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges[J]. IEEE internet of things journal, 2016, 3(5): 637-646.

[3] Taleb T, Samdanis K, Mada B, et al. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1657-1681.

[4] Stafford V. Zero trust architecture[J]. NIST special publication, 2020, 800(207): 800-207.

[5] Eslahi M, Naseri M V, Hashim H, et al. BYOD: Current state and security challenges[C]//2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE). IEEE, 2014: 189-192.

[6] Ghosh A, Gajar P K, Rai S. Bring your own device (BYOD): Security risks and mitigating strategies[J]. Journal of Global Research in Computer Science, 2013, 4(4): 62-70.

[7] Souppaya M, Scarfone K. Guidelines for managing the security of mobile devices in the enterprise[J]. NIST special publication, 2013, 800(124): 124-800.
[8] Roman R, Lopez J, Mambo M. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges[J]. Future Generation Computer Systems, 2018, 78: 680-698.
[9] Gilman E, Barth D. Zero trust networks[M]. O'Reilly Media, Incorporated, 2017.
[10] Kindervag J, Balaouras S. No more chewy centers: Introducing the zero trust model of information security[J]. Forrester Research, 2010, 3(1): 1-16.
[11] Kindervag J. Build security into your network's dna: The zero trust network architecture[J]. Forrester Research Inc, 2010, 27: 1-16.
[12] Ali B, Gregory M A, Li S. Multi-access edge computing architecture, data security and privacy: A review[J]. Ieee Access, 2021, 9: 18706-18721.
[13] Sedjelmaci H, Ansari N. Zero trust architecture empowered attack detection framework to secure 6G edge computing[J]. IEEE Network, 2023, 38(1): 196-202.
[14] Sharma S, Gupta G, Laxmi P R. A survey on cloud security issues and techniques[J]. arXiv preprint arXiv:1403.5627, 2014.
[15] Shah S W, Syed N F, Shaghaghi A, et al. LCDA: Lightweight continuous device-to-device authentication for a zero trust architecture (ZTA)[J]. Computers & Security, 2021, 108: 102351.
[16] Gupta N. Study on Application Architecture for Enabling EDGE Applications[C]//S6-190111, 3GPP TSG-SA WG6 Meeting. 28: 21-25.
[17] Lin J, Shen Z, Miao C. Using blockchain technology to build trust in sharing LoRaWAN IoT[C]//Proceedings of the 2nd International Conference on Crowd Science and Engineering. 2017: 38-43.
[18] Huang Y, Tang J, Cheng Y, et al. Real-time detection of false data injection in smart grid networks: An adaptive CUSUM method and analysis[J]. IEEE Systems Journal, 2014, 10(2): 532-543.
[19] Huang X, Yu R, Kang J, et al. Distributed reputation management for secure and efficient vehicular edge computing and networks[J]. IEEE Access, 2017, 5: 25408-25420.
[20] Gao Y, Ma H, Al-Sarawi S F, et al. PUF-FSM: a controlled strong PUF[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017, 37(5): 1104-1108.
[21] Zhang J, Xue N, Huang X. A secure system for pervasive social network-based healthcare[J]. Ieee Access, 2016, 4: 9239-9250.
[22] Yao Q, Wang Q, Zhang X, et al. Dynamic access control and authorization system based on zero-trust architecture[C]//Proceedings of the 2020 1st international conference on control, robotics and intelligent system. 2020: 123-127.
[23] Barker E B, Barker W C, Burr W E, et al. Sp 800-57. recommendation for key management, part 1: General (revised)[J]. 2007.
[24] Li D, Zhang E, Lei M, et al. Zero trust in edge computing environment: a blockchain based practical scheme[J]. Mathematical Biosciences and Engineering, 2022, 19(4): 4196-4216.
[25] Ali B, Gregory M A, Li S, et al. Implementing zero trust security with dual fuzzy methodology for trust-aware authentication and task offloading in multi-access edge computing[J]. Computer Networks, 2024, 241: 110197.
[26] Lin C, He D, Huang X, et al. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0[J]. Journal of network and computer applications, 2018, 116: 42-52.
[27] Barker E, Chen L, Keller S, et al. Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography[R]. National Institute of Standards and Technology, 2017.
[28] Wang R, Li C, Zhang K, et al. Zero-trust based dynamic access control for cloud computing[J]. Cybersecurity, 2025, 8(1): 12.
[29] Bradatsch L, Miroshkin O, Kargl F. ZTSFC: a service function chaining-enabled zero trust architecture[J]. IEEE Access, 2023, 11: 125307-125327.
[30] OPENSSL,(http://www.openssl.org/).

VIII. Biography Section

Aobo Duan received the B.E. degree from Xi'an University of Posts & Telecommunications,China,in 2024.He is currently pursuing the M.Sc. degree with Xidian University, Xi'an, China. He research interests include satellite and 5G networks security.

Ruhui Ma received the Ph.D. degree in Cyber Security from Xidian University, in 2020. She is currently an associate professor at Xidian University, Xi'an, Shaanxi, China. Her research interests include wireless communication and LTE/LTE-A/5G/6G networks.

Jin Cao received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2008 and 2015, respectively. He has been a Professor with the School of Cyber Engineering, Xidian University since July 2020. He has published over 60 papers on the topics of wireless network security. His research interests include wireless network security and 5G/6G networks.

Shiyang He received the M.S. degree in Telecommunications Engineering from Xidian University, China, in 2016. He is currently working toward the Ph.D. degree at the school of Cyber Engineering, Xidian University, Xian Shaanxi, China. His research interests include cryptographic algorithm, hardware speedup and field-programmable gate array architectures and applications.

Hui Li received the MA.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 1993 and 1998, respectively. He has been a Professor with the School of Cyber Engineering, Xidian University since June 2005. He has published over 170 international academic research papers on information security and privacy preservation. His current research interests include cryptography, information theory, and network coding. Prof. Li is the Chair of ACM SIGSAC China. He served as a Technique Committee Chair or a Co-Chair for several conferences.