Computer Communications 222 (2024) 220–229

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

3GPP Edge–Fog federation: Transparent 3rd-party authentication and application mobility

Asad Ali [a,*], Minhajul Islam [c], Tushin Mallick [c], Mohammad Sakibul Islam [c], Sadman Sakib [c], Md. Shohrab Hossain [c,1], Ying-Dar Lin [b,2]

[a] National Institute of Cyber Security, Ministry of Digital Affairs, Taiwan
[b] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[c] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh

ARTICLE INFO

Keywords:
3GPP edge computing
Fog computing
Authentication
Application mobility
Latency

ABSTRACT

3GPP edge and fog computing paradigms provide computational services to users at low latency. These paradigms alone are not enough to fulfill the users' requirements completely. Therefore, a federation among these computing paradigms is necessary. To realize such federation, there is a need of an authentication mechanism where subscribers of a fog, can access 3GPP edge's services, or vice versa, without buying new subscription, and an application mobility mechanism for continuous service during handover from 3GPP edge to fog, or vice versa, without re-authentication. In this work,[3] we propose: (1) a proxy-based state transfer and third-party authentication (PS3A), that uses a transparent proxy to transfer the authentication and application state information between 3GPP edge and fog, and (2) a token-based state transfer and proxy-based third-party authentication (TSP3A), that uses the proxy to transfer the authentication information and tokens to transfer application state information between 3GPP edge and fog. The proxy plays different roles, via virtual counterparts of entities involved in these protocols, to provide transparency. When the 3GPP edge, using EPS-AKA, receives an authentication request, the proxy relays and behaves as a virtual Home Subscriber Server (vHSS) for the 3GPP edge, and behaves as a virtual user for the fog, which is using OIDC. We applied PS3A and TSP3A to three federation scenarios among 3GPP edge and fog. Experimental results show that PS3A and TSP3A provide authentication within 0.345–2.858 s for a 0–100 Mbps proxy load. The results further show that TSP3A provides application mobility, while taking 40%–52% less time than PS3A, using state tokens. TSP3A and PS3A also reduce the service interruption latency by 82.4% and 84.6%, compared to the cloud-based service, via tokens and prefetching.

1. Introduction

Internet of things (IoT), which connects the daily life things to the rest of the internet and consists of various devices, sensors, and software, has seen a rapid development in recent years. IoT devices enable us to access information from anywhere at any time on any device and reduce the frequency of human intervention which improves the quality of life. As the number of IoT devices is increasing exponentially, the amount of data generated by them is also increasing. In order to handle the generated data, more storage resources and faster real-time data processing are required. To address this issue, computational offloading using fog computing and 3GPP edge computing have emerged as a promising solution.

1.1. 3GPP Edge and Fog computing

3GPP edge computing stems from the European Telecommunication Standards Institute's (ETSI) proposal of integrating the virtualization capabilities into the Mobile network operators (MNOs) within the RAN (Radio Access Network). 3GPP edge computing provides services such as computation and storage within the RAN. These services are similar to cloud but 3GPP edge provides these services with lesser computing power, lesser storage capacity, and at a reduced latency as compared to the cloud. This makes it suitable for the IoT devices and those traditional devices which require real-time data processing at reduced

* Corresponding author.
E-mail address: asad.ali@nics.nat.gov.tw (A. Ali).
[1] Member, IEEE.
[2] Fellow, IEEE.
[3] This work is an extension of [1] with inclusion of two new scenarios, extended testbed, and more results.

https://doi.org/10.1016/j.comcom.2024.05.002
Received 4 September 2023; Received in revised form 12 March 2024; Accepted 1 May 2024
Available online 6 May 2024

latency. 3GPP edge services are provided by the telecommunication companies via SIM card service.

Fog computing provides computational services like cloud computing, but it is decentralized as opposed to the cloud, and it provides computational services at locations closer to the end users. It is a concept developed by the researchers at Cisco, defined as a distributed platform which provides services such as data computation and storage to the end devices and lies between the cloud computing platforms and end users [2]. Fog computing improves the efficiency by placing resources closer to where they are needed. This reduces the distance that data needs to be transported on the network and minimizes the network traffic and latency.

1.2. 3GPP Edge–Fog federation

3GPP edge and fog computing are important to fulfill the requirements of heterogeneous IoT and traditional devices, but any one of them alone cannot fulfill all such requirements because of the capability, capacity, and coverage issues. The 3GPP edge servers deployed by the mobile network operators do not have complete coverage, and fog servers deployed by the fog service providers are usually local, and their service is area restricted. In order to solve the issue of coverage, there is a need for a federation among the fog and the 3GPP edge service providers. Federation is the collaboration among two separate and unconnected networks that have different infrastructures [3].

In federation, the parties which form the collaboration need security, privacy, and independence while sharing resources and capacity. The federation is beneficial for service providers in a sense that they are able to expand their capacity, capability, and coverage. Federation also brings advantages for mobile devices and subscribers as they can access services provided by multiple providers. It is also helpful for non-mobile devices in cases where one fog service provider goes out of service, as devices can fall back to 3GPP edge service providers or other fog services providers federated with the same 3GPP edge network.

1.3. Authentication and application mobility issues

The federation we are proposing, allows 3GPP subscribers to access the services provided by a fog without creating a new account and vice versa. However, a few issues arise as a result of such a federation. Subscribers of one service provider will need to authenticate themselves with the other service providers in order to access their services, as it is not feasible to create a new account, and have multiple subscriptions from multiple service providers. The solution to this issue is third-party authentication, where subscribers are able to authenticate themselves with a service provider via their stored credentials on another service provider. The major issue that arises here is that the 3GPP MEC and the fog, using OIDC, belong to different trust domains and use different protocols for user authentication, and the message flows of these protocols are different. This gives rise to the *third-party authentication* issue, and it becomes necessary to design a solution by which subscribers (the first party) of 3GPP could authenticate themselves with multiple fog service providers (the second party) using their 3GPP credentials (the third party) or vice versa.

Consider a scenario where a user moves out of the coverage of a 3GPP network, it needs to access another 3GPP network [4], or it can access the services provided by the federated fog service providers. If the user does not move instantaneously into the coverage of neighboring fog service providers, a discontinuation of service would occur, which would increase the latency and degrade the user's experience. Also, whenever a user moves from a 3GPP MEC services to the fog service providers, active application sessions must be retained so that the user does not have to start a new session at the fog service provider. The application state of active application sessions must be kept intact and transferred to the fog service providers with minimum latency so that user's experience is not degraded. The same is true for the users that move from the fog service providers to 3GPP edge. This leads us to an *application mobility issue*, and we need to design a solution that transfers the session state of users from the 3GPP MEC to the fog servers, and vice versa. In summary, we have identified two major issues that need to be resolved in order to realize a federation among the 3GPP MEC and fog service providers: third-party authentication and application mobility.

1.4. Proxy and token based solution

In order to solve the third-party authentication and application mobility issues, we propose two solutions to end users, namely: (1) Proxy-based state transfer and third-party authentication (PS3A), and (2) Token-based state transfer and Proxy-based third-party authentication (TSP3A). The reason behind proposing these two solutions is to test the efficacy of both the token-based and proxy-based approaches to see which solution is useful under what conditions. PS3A and TSP3A both make use of a transparent proxy to transfer authentication information of between 3GPP edge and fog servers. The basic design idea behind the proxy is transparency by using virtual counterparts to avoid any changes to the message flows of authentication protocols and existing MEC and fog servers' infrastructures. The two solutions differ from each other in terms of state transfer method. In PS3A, application state transfer is carried out through proxy and, in TSP3A, application state is transferred through a state token. We deployed a testbed to check if these proposals achieve transparent third-party authentication and application mobility, which are the major objectives of this work. We also ran experiments to calculate the latency introduced by these proposals. The essence of this paper is summarized as follows:

- We propose two solutions that allow a user to access a fog's services with a 3GPP subscription, and vice versa, by using a transparent proxy and token based approach.
- The proposed proxy provides translation among multiple authentication protocols and transfers a user's authentication information across different trust domains.
- The proposed token-based method transfers the application state information across different domains at much reduced latency.

The rest of the paper is organized as follows. In Section 2, we describe the background to the OIDC and EPS-AKA that are used for authentication along with the related work. We present the problem scenarios and formulate the problem in Section 3. Section 4 explains the proposed proxy design, architecture, and message flows for two federation scenarios. In Section 5, we present the implementation and testbed. Results and their evaluations are presented in Section 6. Section 7 concludes the paper along with some research directions for future work.

2. Background and related work

In this section, we discuss 3GPP edge computing, fog computing, the authentication protocols used in this work, and related work.

2.1. 3GPP edge computing

3GPP edge computing or multi-access edge computing is presented by ETSI [5], and is a distributed computing paradigm that brings computation at the network's edge to deliver low latency and save bandwidth [6]. 3GPP edge servers can be deployed in the existing cellular networks (4G-LTE or 5G) to converge the cellular and IT services [7]. The subscribers of 4G-LTE are able to access the service of 3GPP edge servers once they are authenticated with the underlying 4G-LTE network. 4G-LTE architecture offers reduced latency, higher capacity, higher speed, and flexible bandwidth usage. The components of 4G architecture are: User Equipment (UE), evolved NodeB

(eNodeB), and Evolved Packet Core (EPC). EPC consists of different entities among which Mobility Management Entity (MME) and Home Subscriber Server (HSS) play role in the authentication.

These entities perform multiple functions like authentication, session management, and mobility management etc. The main control entity in the EPC is the MME which communicates with the HSS for authentication of the users. MME offers several functionalities like roaming management, mobility management, radio resource management, and load balancing between S-GWs. S-GW acts as a mobility anchor, and it routes and forwards user data packets. HSS is the central database which provides user information to MME for user authentication. The 3GPP edge servers are usually deployed between the eNB and EPC [8] and 4G-LTE subscribers are able to use their services.

2.2. Fog computing

Centralized cloud computing faces some challenges like high latency and less security. In 2011, fog computing was introduced to tackle the huge amounts of data along with providing real-time processing for low latency applications [2–9]. In 2015, the OpenFog Consortium was founded to promote the public's interests and to advance the development of fog computing [10]. Fog computing has a layer based architecture [11] that provides low latency and more secure system than the cloud due to its distributed architecture. The major difference between cloud and fog computing is the decentralized architecture and location of the services. In cloud computing, data is processed far away from the end users, and it is suitable for such applications that need more computational power, better storage, and in-depth analysis of the data.

In fog computing, data is stored and processed closer to the information source, which makes it the first choice for the latency sensitive applications. Fog computing provides distributed storage and computing resources and provides better security due to its decentralized nature as it becomes difficult for the attackers to manipulate the data. Therefore, fog offers low latency and more secure system as compared to the cloud and allows collaboration of different physical environments among multiple services and provides flexibility to users. Fog computing will be more and more necessary as the number of IoT devices increase and real time cities emerge [12].

2.3. 3GPP edge and fog authentication process

EPS-AKA is used for authentication in the 4G-LTE networks. The authentication procedure is defined by the 3GPP group for mobile users' authentication when they access the EPS network [13]. The authentication process starts when a UE sends the "attach request" containing the UE's IMSI to the MME which forwards the received IMSI to HSS that generates an authentication vector (AV) using the UE's IMSI as an authentication response and sends it back to the MME which stores XRES and sends RAND and authentication token (AUTN) to the UE. The UE computes RES using these values, and sends it to the MME which compares the value of RES and XRES and authenticates the UE. Once the UE is authenticated with the EPC of a cellular network, it can use the services of 3GPP edge servers deployed in the cellular network using MECsec design [14].

Fog authenticates the users via user credentials (i.e., Username and password). The users are authenticated by fog using authentication token generated from trusted sources. OpenID Connect (OIDC), a simple identity layer, extends OAuth 2.0 protocol and provides an effective technique to identify individuals to service providers, assists to get user profile information, and offers the user's identity verification. Hence, OIDC is the most probable choice for third-party authentication in fog service providers. OIDC is a protocol for third party authentication and consists of three components: User, OpenID Provider (OP or IdP), and Relying Party (RP). OIDC provides a mechanism to authenticate a user to the RP by using account information details of the user stored in the IdP.

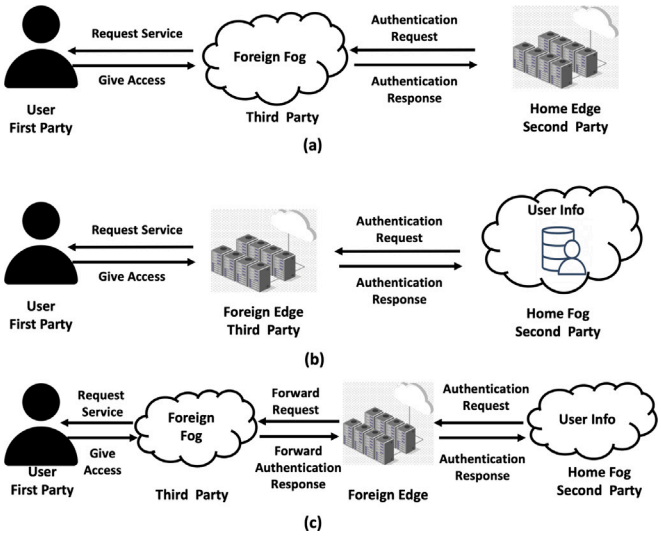

Fig. 1. Problem scenarios (a) Edge–Fog federation (b) Fog–Edge federation (c) Fog–Edge–Fog federation.

2.4. Related work

There are multiple studies in literature that focus on the authentication for 3GPP edge and fog. We surveyed such studies to find out the existing solutions to the third-party authentication problem in federated 3GPP edge and fog. We also analyzed the transparency of the proposed solutions as per existing standards. Our findings are summarized in Table 1. Some studies [15,19] propose mutual authentication in the edge–fog environment by using blockchain or secret splitting but do not provide transparency. There are also studies [16,18] that provide roaming for the WLAN/Cellular architecture via using AAA broker, but they do not provide authentication services for the fog users in 3GPP edge services.

A few studies [20,21] focus on group roaming and privacy preservation by using aggregate signatures and Pseudonym Based Cryptography (PBC) but do not provide transparent solutions for the fog user's authentication in the 3GPP edge. There are also studies [22,23] that provide 3GPP re-authentication or 3G-WLAN inter-working, but they modify the existing EAP-AKA mechanism. Apart from these, some studies focus on Cloud–Fog [17] or Cloud–Edge [24] federation as well. Cloud–Edge federation [24] is closely related to our work, but the difference lies in the involved computing paradigms and federation scenarios. An earlier study [25] from some of us solves the problem of third-party authentication for mobile users while they move between the MECs deployed by the 4G-LTE networks but does not provide a federation between fog and 3GPP edge and does not consider multiple federation scenarios. In this work, we propose a federation among 3GPP edge and fog while considering multiple scenarios and transparency. To the best of our knowledge, none of the mentioned studies has solved the issue of providing 3GPP edge services to the fog users, or vice versa, while considering transparency and multiple federation scenarios.

3. Problem formulation

3.1. Problem scenarios

Consider an MEC deployed in a 3GPP MNO which is in federation with a fog service provider. We consider two scenarios which are Edge–Fog and Fog–Edge.

Table 1
Related work.

Author	Objective	Scenario	Approach	Transparency
Imine [15]	Mutual Authentication	Edge–Fog	Blockchain and Secret Sharing	X
Minghui [16]	Roaming & Authentication	Edge–Fog	AAA Broker with RADIUS	✓
Sarang [17]	Security	Cloud–Fog	SDN	X
Minghui [18]	WLAN/Cellular Roaming	Edge–Fog	Agent-based integrated service	X
Yixin [19]	Mutual Authentication & Key exchange	Edge–Fog	Secret Splitting & Self certification	X
Chengzhe [20]	Group Roaming	Edge–Fog	certificate-less aggregate signatures	X
Amor [21]	Privacy-preserving Authentication	Edge–Fog	Pseudonym Based Cryptography	X
Shidhani [22]	3GPP Re-authentication	Edge–Fog	Modified EAP-AKA	X
Hyeran [23]	3G-WLAN Mutual Authentication	Edge–Fog	Modified EAP-AKA	X
Ours	Federated Authentication, Application Mobility	Fog–Edge Edge–Fog	Transparent Proxy, Tokens	✓

3.1.1. Edge–Fog

In the first scenario, we assume that a user is the subscriber of the 3GPP MNO and uses the services provided by the MEC deployed by the 3GPP MNO. The home subscriber server (HSS) in the 3GPP MNO contains the subscription and authentication information of the user; the fog service provider does not have any information of the user. In this scenario, as shown in Fig. 1(a), we assume that the user has moved to the fog service provider while using the MEC services in the 3GPP MNO. In this scenario, the fog service provider needs to get the authentication information from the 3GPP MEC, along with the application state information, in order to obtain the service continuity.

3.1.2. Fog–Edge

In the second scenario, we assume that a user is the subscriber of a fog service provider and uses the services provided by the fog. The fog service provider is the home for the user and the 3GPP MNO is the foreign service provider for the user as it does not have any information of the user. In this scenario, as shown in Fig. 1(b), we assume that the user has moved from using the fog service provider towards the MEC services provided by the 3GPP MNO. In this scenario, the fog service provider needs to provide the authentication information to the 3GPP MEC, along with the application state information so that the user could continue the session.

In the Fog–Edge scenario, we also consider another case which is Fog–Edge–Fog, as shown in Fig. 1(c), where we assume that the user has disconnected from the home fog and moved to the foreign fog. In this case, the foreign fog needs to derive the authentication information from the home fog. The issue here is that foreign fog is not federated directly with the home fog, and they are indirectly federated via a foreign 3GPP edge. In this work, we will only consider these two cases where the user is a subscriber of a fog and needs the services of a 3GPP edge federated with fog or needs the services of another indirectly federated fog.

We consider both the scenarios where either the UE moves from the 3GPP MNO to the fog service provider or the UE moves from the fog service provider to the 3GPP MNO, because our objective is to form a federation between the fog service providers and the 3GPP MNOs by performing transparent 3rd-party authentication along with seamless application mobility with minimal latency. An application can be stateful or stateless [26]. In stateful applications, user data, also called an application state, denotes application usage, such as the number of seconds watching a video. In such applications, when a user switches the service providers, the application state has to be migrated to resume the application from the same position while keeping service interruption delay to a minimum.

3.2. Problem statement

We have an MEC framework in a 3GPP cellular network that is federated with a fog network. A 3GPP subscriber authenticated with the 3GPP cellular network, may or may not be using certain applications in the 3GPP MEC, moves to a fog network and wants to access applications in the fog server or a fog subscriber authenticated with the fog service provider, may or may not be using certain applications in the fog, moves to a 3GPP MEC network and wants to access applications in the 3GPP MEC. In both the cases, either fog or 3GPP edge can be the home or foreign service providers. If the user has a subscription with the 3GPP edge and moves to the fog service provider, the 3GPP edge will be considered as home and the fog service provider will be considered as the foreign service provider.

The objective is to provide the services of the foreign service provider to the subscriber of the home service provider without creating another account. We assume that the fog is using OpenID Connect (OIDC), which is a popular third-party authentication mechanism that allows a client to authenticate an end-user based on authentication with an authorization server and obtain information about user [27]. It is predicted that in the coming years, OIDC will have widespread adoption in fog computing and IoT applications [28]. The subscriber may also start using a particular application in the foreign network from the same state it had left off in the home network. This objective must be achieved at low latency while maintaining the transparency of existing 3GPP and fog architecture and protocols.

4. Proposed design architecture

We propose Proxy-based state transfer and third-party authentication (PS3A), and Token-based state transfer and Proxy-based third-party authentication (TSP3A) for solving the authentication and application mobility problems. PS3A and TSP3A make use of a transparent proxy to transfer the user's information from the 3GPP MEC to the fog. The major design idea behind the proxy is transparency, to avoid any modifications in the existing 3GPP cellular network, MEC, and fog infrastructure. We provide transparency by proposing virtual counterparts inside the proxy to communicate the MEC and fog entities with their virtual counterparts. PS3A and TSP3A share a common third-party authentication solution, via proxy, and differ in the state transfer method for application mobility via the proxy and via the token, respectively.

4.1. Architecture

The proxy needs to be deployed between the 3GPP MEC and fog network, as shown in Fig. 2. The MEC platform is deployed in a 3GPP cellular network that contains the necessary infrastructure to run MEC applications [29]. The MEC platform manager takes care of the application requirements and a system level entity, which is the MEC controller, coordinates all MEC platforms within the 3GPP MEC network via a system orchestrator. In the fog network, an authentication module handles the authentication related tasks. Application Mobility Module in both fog and MEC controllers handles the tasks related to the application state transfer. The proposed proxy connects the fog and MEC network at system level using different virtual counterparts and MEC and fog controllers. We assume that OIDC is available as an authentication mechanism in the fog network. Therefore, the proxy acts

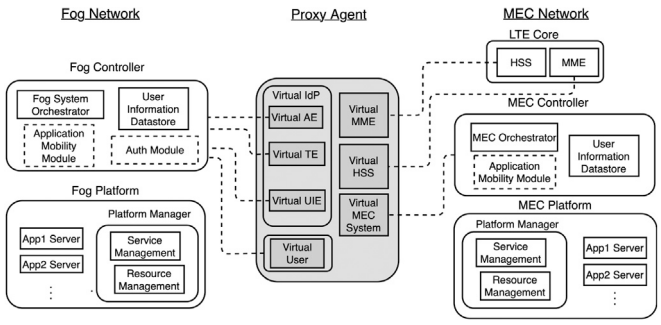

Fig. 2. PS3A and TSP3A architecture (Deployment of proxy agent between 3GPP Edge and Fog).

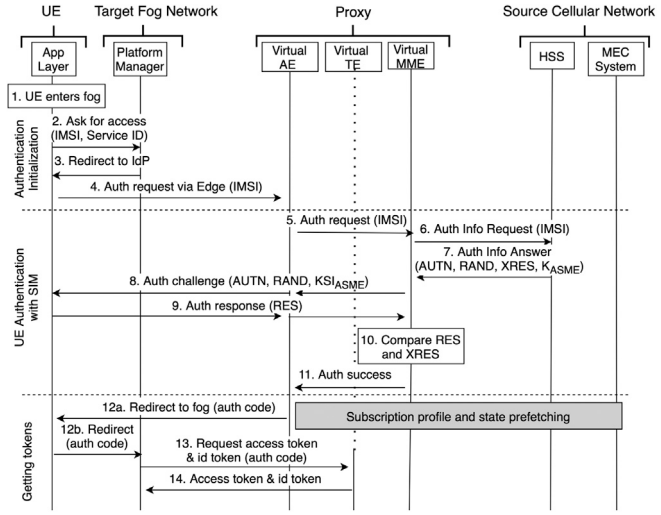

Fig. 3. Authentication message flow (UE authenticates itself with a Fog using 3GPP Edge).

as a virtual Identity Provider (vIdP) while communicating with the fog so that the RP component in the fog can communicate with vIdP. The vIdP consists of a virtual Authorization Endpoint (vAE), a virtual Token Endpoint (vTE) and a virtual Userinfo Endpoint (UIE). The proxy acts as a virtual MME (vMME) and a virtual MEC system for the 3GPP MEC in order to be transparent.

In order to provide application mobility, there is a need of transferring the session state information from the 3GPP MEC to the fog server. Therefore, in PS3A, proxy acts as a virtual UIE so that fog can request user's session state from the vUIE. The proxy then collects the session state from the 3GPP MEC by acting as the Virtual MEC System. On the other hand, for TSP3A, no additional component in the proxy is required. The virtual components inside the proxy adhere to the specification of the system they are responsible for and ensure a standardized interface to the outside world. Here, by virtual we mean that these components are implemented in software modules rather than being implemented on separate hardware. With such implementation, proxy achieves fast, secure, and reliable internal communication.

4.2. Message flows

The different scenarios in our problem statement are Edge–Fog, Fog–Edge, and Fog–Edge–Fog scenarios. For the sake of simplicity, we explain the message flow for the Edge–Fog scenario where a subscriber of 3GPP edge needs to access the service of a foreign fog. In order to provide third-party authentication and application mobility to the subscriber, we identify four stages:

(1) *Registration*. A fog and an MEC network make an agreement for federation and register their corresponding components with a proxy agent. (2) *Third-party authentication*: When the user moves from the MEC to the fog network, it is authenticated to the fog network with its 3GPP cellular credentials. (3) *Subscription profile collection*: Fog network collects the subscription profile of the user from the MEC and verifies the service access. (4) *Application state transfer*: Application state is transferred from MEC to the fog network and the user resumes using service from the fog network.

After describing these four steps, we discuss how proxy can fetch information early from source MEC to reduce service resumption delay.

4.2.1. Registration

The fog and MEC network connect with a proxy agent in this stage. A Diameter connection is set up between the vMME in proxy and HSS in 3GPP cellular network, which is secured via TLS. As per OIDC standards [27], the fog platform manager (FPM) in the fog network registers with the vIdP component in the proxy as the Relying Party (RP) in OIDC terms, and receives the client ID.

4.2.2. Third-party authentication

PS3A and TSP3A authenticate a 3GPP subscriber with the fog network via its 3GPP cellular network credentials in the following 3 stages, *authentication initialization, UE Authentication with SIM Credentials,* and *Obtaining Tokens*, as shown in Fig. 3. The FPM identifies the user and redirects it to vAE which has it authenticated via vMME and HSS in the 3GPP network by using 3GPP EPS-AKA protocol. After successful authentication, FPM receives an access token and an ID token from vTE for the authenticated user.

The user needs to authenticate with OIDC in fog network via using its 3GPP cellular network credentials. Therefore, we combine the OIDC in fog network and EPS-AKA in the cellular network. The message flow, as shown in Fig. 3, consists of 3 stages namely: *authentication initialization, UE Authentication with SIM Credentials*, and *Obtaining Tokens*. PS3A and TSP3A authenticate a 3GPP subscriber with the fog network.

Authentication Initialization: When the user enters a fog network, it requests access to a particular service by third-party authentication with cellular credentials. Fog Platform Manager (FPM) identifies the user as the subscriber of the federated 3GPP MEC network and redirects the user to the virtual AE. The UE presents its IMSI to Virtual AE.

UE Authentication with SIM Credentials: The user is then authenticated via its SIM credentials. The vIdP sends IMSI internally to the vMME which authenticates the user by using EPS-AKA. The messages between vMME and the user are exchanged via vAE. If the user authentication is successful, vMME informs the vAE about successful authentication.

Obtaining Tokens: When the vAE confirms successful authentication of the user, it redirects the user to FPM with the authorization code. In the meantime, a new record for the user containing the IMSI is created in vIdP in proxy. After receiving the authorization code, FPM requests an access token and ID token from vTE which generates an access token and an ID token and returns these to FPM.

4.2.3. Subscription profile collection

After authentication, FPM needs to obtain the user's subscription profile, stored in the MEC controller, to perform authorization, accounting, and ensure QoS. FPM fetches this information from the MEC network and verifies subscription before initializing the service application instance for the user as shown in Fig. 4. Then, FPM sends a subscription profile request to vUIE with the access token which then verifies the token and looks up IMSI for this token. Proxy, while acting as the virtual MEC system, sends this IMSI to the MEC controller and collects subscription profile and returns this information to the FPM which verifies the subscription and informs the user whether service access is accepted.

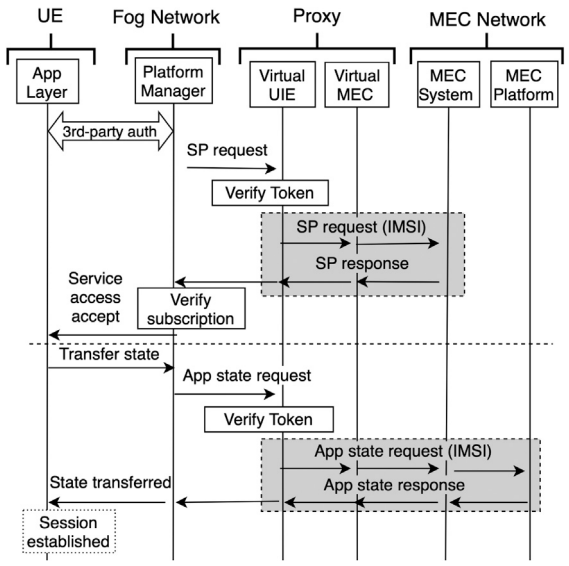

Fig. 4. Subscription profile collection and state transfer message flow (state transferred from 3GPP Edge to Fog).

4.2.4. Application state transfer

We use two different methods to transfer the application state from the MEC to the fog platform.

PS3A. In PS3A, the application state (JSON format) is transferred via the proxy which acts as vUIE to the fog network. The user requests the FPM to transfer the state from the MEC which, in turn, requests vUIE for the application state as a claim with the access token obtained in the authentication step. The proxy acts as a virtual MEC system and sends the application state request, with the saved IMSI, to the MEC network which locates the user's MEC platform via its IMSI and forwards the application state request to that MEC platform. The MEC platform returns the application state to the MEC controller, which then returns it to the proxy. The proxy then provides the state to the FPM which initializes the application with this state and initiates a session with the user.

TSP3A. In TSP3A, the UE receives a state token every time it updates its application state while accessing MEC applications. The state token contains the state information and the validity of the token. After disconnecting with the MEC network, the user provides the fog network with the state token after authentication with the fog network. The fog checks the validity of the token and updates the application state, based on the state information within the state token.

Comparison between PS3A and TSP3A. In PS3A, a state is transferred through proxy which adds network delay. TSP3A transfers the state via a token and incurs less delay for the application state transfer. The PS3A transfers the state through backhaul, with the necessary security measures, and TSP3A ensures encryption and integrity protection for the token for secure transfer. In PS3A, fog network always receives the latest state from the MEC while in TSP3A, most up-to-date state may not be sent and some state information may become lost. Furthermore, a periodic state update in TSP3A adds extra overhead to the system. Hence, TSP3A is suitable for applications that need low latency and PS3A is suitable for applications that require most up-to-date state.

Prefetching. After successful authentication, in both PS3A and TSP3A, proxy needs to collect subscription profile from source MEC. Moreover, proxy needs to transfer state from source MEC in PS3A. Proxy can fetch this information early even before the information is requested. This information can be stored in proxy in temporarily and returned to fog network when requested. These two prefetching steps are shown inside rectangles in Fig. 4. To avoid unnecessary caching, proxy performs prefetching only after third-party authentication is confirmed in Virtual AE as shown in Fig. 3.

5. Implementation

5.1. Prototype architecture

We used Open Air Interface (OAI) [30] for deploying the 4G-LTE cellular network components and deployed the MEC platform inside the 3GPP cellular network using Node.js which simply acted as a backend server for the proxy. We also set up an additional application server in the MEC platform and a state manager, using Node.js, in the fog for handling application state. The User Equipment (UE) and Edge components (including HSS, MME, and eNB) were implemented using OAI. The fog components were implemented in Python. The proxy's vHSS used OAI provided HSS and the proxy's vUser portion was implemented in Python. The fogs were implemented using Python and Django web framework. The UE, MME of the foreign edge, and virtual HSS were run in different docker networks within the same machine.

5.2. Testbed

The three scenarios, Edge–Fog, Fog–Edge, and Fog–Edge–Fog were implemented using 2 machines, both having different specifications and hardware. In the Edge–Fog, and Fog–Edge scenario, the first machine was used for the UE, edge, and proxy and the second machine was used for the fog. The UE was implemented using OAI provided Radio Access Network (RAN) codebase and core components were implemented using OAI provided Core Network (CN) codebase. In the Fog–Edge–Fog scenario, we needed two fog networks and two proxies. The first machine was used for the UE, foreign fog, and the home fog. The second machine was used for the edge components and two proxies. The proxy between the foreign fog and foreign edge (proxy 1) had 2 modules where the vIdP module was implemented using Python and Django OIDC provider. The vUE module was also implemented in Python. The proxy between the foreign edge and home fog (proxy 2) also had 2 modules which were virtual HSS and virtual User.

6. Results and evaluation

For evaluation of our proposed methods, we measured third-party authentication latency for 3 scenarios, Edge–Fog, Fog–Edge, and Fog–Edge–Fog. We also calculated the state transfer latency for two methods, PS3A and TSP3A for 2 scenarios, Edge–Fog and Fog–Edge. We also compared the service interruption time taken by our proposed solutions against the authentication and state transfer time taken in the absence of our solutions (i.e., via cloud).

6.1. Authentication latency

6.1.1. Edge-Fog

First, we measured latency due to third-party authentication by applying different loads on proxy as shown in Fig. 5. We divided the authentication latency into three parts: (1) proxy latency, (2) OIDC latency, and (3) Edge and network communication latency. The authentication latency was 345 ms without any load on the proxy. We created network load on proxy by opening hundreds of sockets in proxy and sending network traffic to those sockets (base load). The authentication latency significantly increased when network load was increased on the proxy. Authentication latency increased to 2858 ms, about 8 times the delay without any load, with 100 MB/s load. The OIDC latency increased with proxy load as the proxy acted as a vIdP for the OIDC client i.e., the fog and took increasingly more time to serve the OIDC client and thus the increase in OIDC latency. The proxy latency also increased, but the increase in OIDC latency is greater than the proxy latency as the proxy prioritizes its own workload over serving the OIDC client.

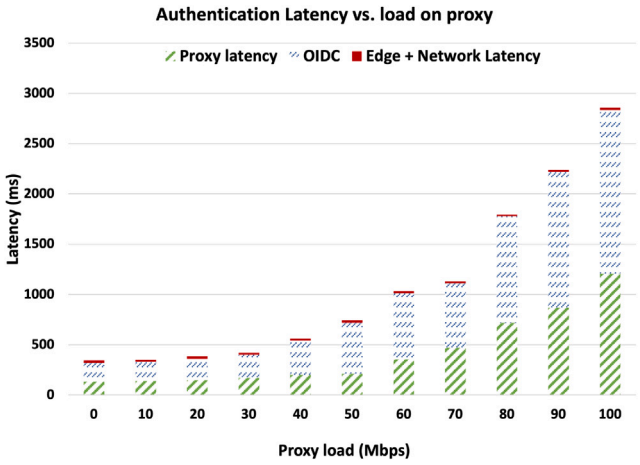

Fig. 5. Edge–Fog authentication latency (Increase in Proxy load increases latency).

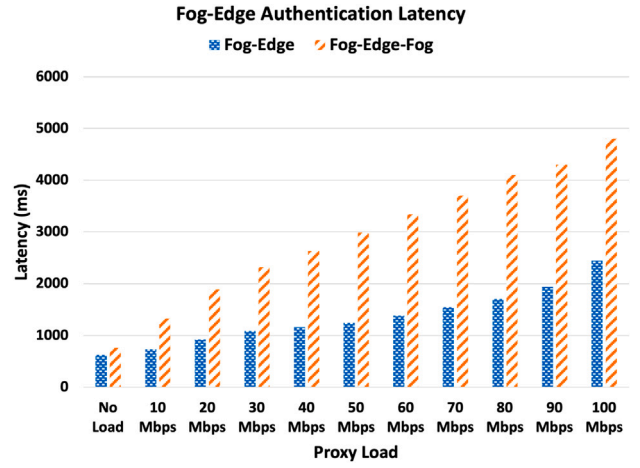

Fig. 6. Fog–Edge authentication latency (Scenario 1: Fog–Edge, Scenario 2: Fog–Edge–Fog (increased latency due to involvement of 5 entities)).

6.1.2. Fog–Edge

We also evaluated the authentication latency for Fog–Edge scenario under different proxy loads. The results can be seen in Fig. 6. The authentication latency varied from 600–2446 ms for no load–100 Mbps load on proxy. The increase in the authentication latency was 70%, for a 400% increase in proxy load. This clearly shows that the proposed solution is capable of handling multiple third-party authentication requests without incurring much latency. We also broke down the authentication latency into 3 entities involved in the authentication process.

We found that the proposed proxy took the least percentage of time among involved entities. Proxy took 0.7% of the total authentication time which was 78% and 99% less than the time taken by the fog and the 3GPP edge components. This shows that the proposed proxy does not cause the bottleneck and the bottleneck is rather created by the 3GPP edge components. In order to analyze the time taken by 3GPP edge components, we broke down the time taken by 3GPP edge into individual messages and found that the *"Attach Request"* and *"200 OK"* messages take the most amount of time which are the part of the standard EPS-AKA protocol and hence cannot be modified.

6.1.3. Fog–Edge–Fog

We also evaluated the authentication latency for the Fog–Edge–Fog scenario under different proxy loads. The results can be seen in Fig. 6. The authentication latency varied from 760–4800 ms for no load–100 Mbps load on the proxy. The increase in the authentication latency was 125%, for a 400% increase in proxy load. This clearly shows that the proposed solution is capable of handling multiple third-party authentication requests without incurring much latency. It can be seen that the authentication latency for the Fog–Edge–Fog scenario is greater than the Fog–Edge scenario because the Fog–Edge–Fog scenario involves two proxies and multiple virtual components. There are five entities involved in the Fog–Edge–Fog scenario, as opposed to the three entities involved in the Fog–Edge scenario. We broke down the authentication latency into five entities involved in the authentication process and found that the proposed proxies took the least percentage of time among involved entities. The proxy-1 and proxy-2 took 0.1% and 0.6% of the total authentication time.

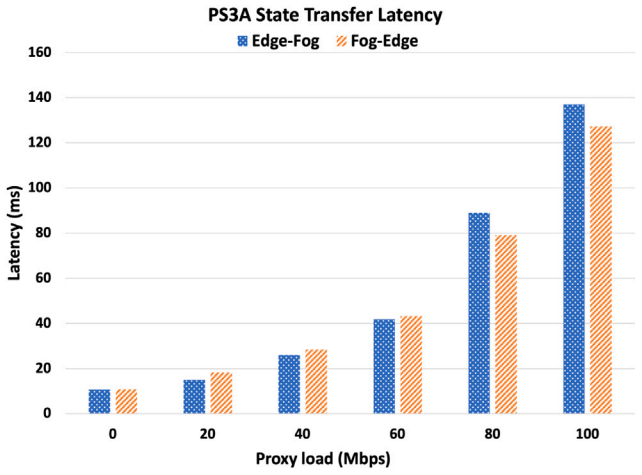

Fig. 7. PS3A State transfer latency (PS3A transfers state from Edge to Fog and Fog to Edge with almost similar latencies).

6.2. State transfer latency

We also calculated the state transfer latency for PS3A and TSP3A. For each method, we calculated the latency for Edge–Fog and Fog–Edge scenarios.

6.2.1. PS3A

We calculated the state transfer latency for Edge–Fog and Fog–Edge scenarios using the PS3A method whilst increasing the network load on the proxy as shown in Fig. 7. We used a small state (<100B), which the proxy was able to pre-fetch from the MEC before it was requested by the fog platform. Therefore, the PS3A state transfer latency was only because of the data transmission from the proxy to the fog platform. Fig. 7 shows that as the network load is increased on the proxy, PS3A state transfer latency also increases and ranges between 10.6–137 ms for 0–100 Mbps load. Fig. 7 also shows the PS3A state transfer latency for the Fog–Edge scenario, which increases as the network load is increased on the proxy and ranges between 10.7–127 ms for 0–100 Mbps load. This is quite similar to the PS3A state transfer latency for the Edge–Fog scenario. It can be seen that PS3A does not incur much state transfer latency under proxy load.

6.2.2. TSP3A

We also analyzed the TSP3A state transfer latency for Edge–Fog and Fog–Edge scenarios, as shown in Fig. 8. In TSP3A, the proxy is not involved in the state transfer and therefore, we increased the number of UEs that send simultaneous state update requests to the fog. In the Edge–Fog scenario, TSP3A state transfer latency increases as the number of UEs connected to the fog increases and ranges between 6.5–746.5 ms for 1–100 UEs. The increase in state transfer latency is

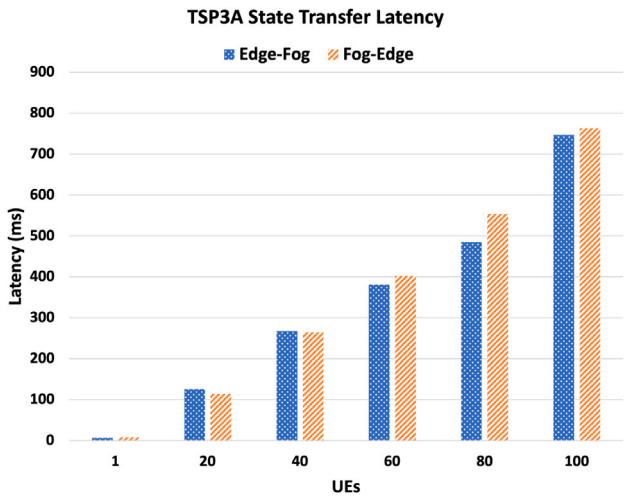

Fig. 8. TSP3A State transfer latency (Both scenarios see a similar latency increase trend with an increase in the number of UEs).

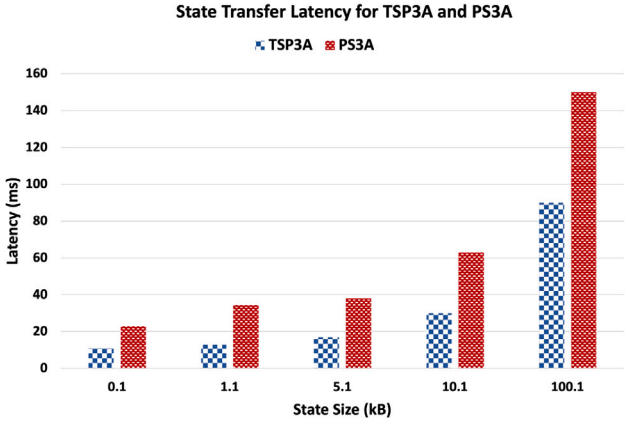

Fig. 9. State transfer latency vs. State size (TSP3A takes 40%–52% less time than PS3A).

linear and when the number of UEs reaches 100, the state transfer latency increases by a greater amount, most probably as a consequence of the network traffic collision. Fig. 8 also shows the TSP3A state transfer latency for the Fog–Edge scenario, which ranges between 7.92–762.8 ms for 1–100 UEs. The increase in state transfer latency for the Fog–Edge scenario is similar to the Edge–Fog scenario.

6.3. State transfer comparison

In order to compare PS3A and TSP3A, we used different state sizes to see how these methods behave for different state sizes. It can be seen from Fig. 9 that TSP3A state transfer latency ranges between 11–90 ms and PS3A state transfer latency ranges between 23–150 ms for the state size of 0.1–100.1 KB. It should also be noted that, for different state sizes, TSP3A takes 40–52% less time than PS3A because PS3A retrieves the state from an entity (MEC) located farther away whereas, TS3 A retrieves the state via the UE which takes less time. Besides, TSP3A state transfer took 94–98% less time compared to state transfer from cloud.

6.4. Service interruption latency

We also compared the service interruption latency of PS3A and TSP3A with the state transfer via cloud. The state size was 100 KB in all three methods. Total service interruption time in TSP3A, PS3A, and cloud-based approach was 435 ms, 495 ms, and 2817 ms, respectively. TSP3A took the least amount of time which is 12.1% and 84.6% less than the PS3A and the cloud. PS3A and TSP3A took 82.4% and 84.6% less time compared to cloud, respectively.

7. Conclusion and future work

A federation among 3GPP edge and fog is useful for both subscribers and providers as subscribers can access services of different providers with one account and providers can enhance their capacity, coverage, and capability. In order to realize federation, third-party authentication and application mobility are necessary, which are challenging because fog and 3GPP edge belong to different trust domains and use different authentication protocols. In this work, we proposed PS3A and TSP3A methods that use a proxy for transferring the authentication information of subscribers from 3GPP MEC to the fog and vice versa, and use proxy and tokens respectively, for the application state transfer. We implemented the proxy on a testbed and the results show that PS3A and TSP3A provide authentication within 345–2858 ms, 600–2446 ms, and 760–4800 ms, for Edge–Fog, Fog–Edge, and Fog–Edge–Fog scenarios respectively, when the proxy load is increased from 0–100 Mbps. The PS3A used the proxy to provide application mobility within 10.6–137 ms, and 10.7–127 ms, for Edge–Fog and Fog–Edge scenarios respectively, when the proxy load is increased from 0–100 Mbps.

The TSP3A used tokens to provide the application mobility within 6.5–746 ms, and 7.92–763 ms, for Edge–Fog and Fog–Edge scenarios respectively, for 0–100 UEs. The results further show that TSP3A provides application mobility while taking 40–52% less time than PS3A via using state token. The advantage of proposed methods over existing methods is that TSP3A and PS3A reduce the service interruption latency by 82.4% and 84.6%, compared to the cloud-based service, via tokens and prefetching. In the future, we will extend this work to provide solutions for horizontal, vertical, and hybrid federation scenarios involving cloud, 3GPP edge, and fog. This work can also be extended by providing a comprehensive security analysis of the proposed proxy while considering different threats. Also, other federation issues, such as resource sharing, traffic offloading, and load balancing among cloud, 3GPP edge, and fog can be addressed in the future.

CRediT authorship contribution statement

Asad Ali: Conceptualization, Investigation, Writing – original draft, Writing – review & editing, Methodology. **Minhajul Islam:** Data curation. **Tushin Mallick:** Data curation, Methodology. **Mohammad Sakibul Islam:** Validation. **Sadman Sakib:** Methodology. **Md. Shohrab Hossain:** Methodology, Resources, Software. **Ying-Dar Lin:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Asad Ali, Tushin Mallick, Sadman Sakib, Md Shohrab Hossain, Ying-Dar Lin, Provisioning fog services to 3GPP subscribers: Authentication and application mobility, in: ICC 2022-IEEE International Conference on Communications, IEEE, 2022, pp. 4926–4931.

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13–16.

[3] Martín Serrano, Steven Davy, Martin Johnsson, Willie Donnelly, Alex Galis, Review and designs of federated management in future internet architectures, in: The Future Internet Assembly, Springer, Berlin, Heidelberg, 2011, pp. 51–66.

[4] Asad Ali, Samin Rahman Khan, Sadman Sakib, Md Shohrab Hossain, Ying-Dar Lin, Federated 3GPP mobile edge computing systems: A transparent proxy for third party authentication with application mobility support, IEEE Access 10 (2022) 35106–35119.

[5] Bin Liang, Mark A. Gregory, Shuo Li, Multi-access edge computing fundamentals, services, enablers and challenges: A complete survey, J. Netw. Comput. Appl. (2021) 103308.

[6] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, Lanyu Xu, Edge computing: Vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.

[7] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, Dario Sabella, On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1657–1681.

[8] Fabio Giust, Gianluca Verin, Kiril Antevski, Joey Chou, Yonggang Fang, Walter Featherstone, Francisco Fontes, Danny Frydman, Alice Li, Antonio Manzalini, et al., MEC deployments in 4G and evolution towards 5G, ETSI White Paper 24 (2018) 1–24.

[9] Amir Vahid Dastjerdi, Harshit Gupta, Rodrigo N. Calheiros, Soumya K. Ghosh, Rajkumar Buyya, Fog computing: Principles, architectures, and applications, in: Internet of Things, Elsevier, 2016, pp. 61–75.

[10] M.S.V. Janakiram, Is fog computing the next big thing in internet of things, Forbes Magazine (2016).

[11] Mohammad Aazam, Sherali Zeadally, Khaled A. Harras, Fog computing architecture, evaluation, and future research directions, IEEE Commun. Mag. 56 (5) (2018) 46–52.

[12] Rob Kitchin, The real-time city? Big data and smart urbanism, GeoJournal 79 (1) (2014) 1–14.

[13] Mourad Abdeljebbar, Rachid El Kouch, Security improvements of EPS-AKA protocol, Int. J. Netw. Secur. 20 (4) (2018) 636–644.

[14] Chi-Yu Li, Ying-Dar Lin, Yuan-Cheng Lai, Hsu-Tung Chien, Yu-Sheng Huang, Po-Hao Huang, Hsueh-Yang Liu, Transparent aaa security design for low-latency mec-integrated cellular networks, IEEE Trans. Veh. Technol. 69 (3) (2020) 3231–3243.

[15] Youcef Imine, Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, Lounis Ahmed, MASFOG: An efficient mutual authentication scheme for fog computing architecture, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE, IEEE, 2018, pp. 608–613.

[16] Minghui Shi, Xuemin Shen, Jon W. Mark, IEEE 802.11 roaming and authentication in wireless LAN/cellular mobile networks, IEEE Wirel. Commun. 11 (4) (2004) 66–75.

[17] Sarang Kahvazadeh, Vitor B. Souza, Xavi Masip-Bruin, Eva Marn-Tordera, Jordi Garcia, Rodrigo Diaz, Securing combined fog-to-cloud system through SDN approach, in: Proceedings of the 4th Workshop on CrossCloud Infrastructures & Platforms, 2017, pp. 1–6.

[18] Minghui Shi, Humphrey Rutagemwa, Xuemin Shen, Jon W. Mark, Aladdin Saleh, A service-agent-based roaming architecture for WLAN/cellular integrated networks, IEEE Trans. Veh. Technol. 56 (5) (2007) 3168–3181.

[19] Yixin Jiang, Chuang Lin, Xuemin Shen, Minghui Shi, Mutual authentication and key exchange protocols for roaming services in wireless mobile networks, IEEE Trans. Wireless Commun. 5 (9) (2006) 2569–2577.

[20] Chengzhe Lai, Hui Li, Rongxing Lu, Rong Jiang, Xuemin Shen, SEGR: A secure and efficient group roaming scheme for machine to machine communications between 3GPP and WiMAX networks, in: 2014 IEEE International Conference on Communications, ICC, IEEE, 2014, pp. 1011–1016.

[21] Arij Ben Amor, Mohamed Abid, Aref Meddeb, A privacy-preserving authentication scheme in an edge-fog environment, in: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications, AICCSA, IEEE, 2017, pp. 1225–1231.

[22] Ali A. Al Shidhani, Victor C.M. Leung, Fast and secure reauthentications for 3GPP subscribers during WiMAX-WLAN handovers, IEEE Trans. Dependable Secur. Comput. 8 (5) (2010) 699–713.

[23] Hyeran Mun, Kyusuk Han, Kwangjo Kim, 3G-WLAN interworking: security analysis and new authentication and key agreement based on EAP-AKA, in: 2009 Wireless Telecommunications Symposium, IEEE, 2009, pp. 1–8.

[24] Ying-Dar Lin, Duc-Tai Truong, Asad Ali, Chi-Yu Li, Yuan-Cheng Lai, Thai-Mai Thi Dinh, Proxy-based federated authentication: A transparent third-party solution for cloud-edge federation, IEEE Netw. 34 (6) (2020) 220–227.

[25] Asad Ali, Ying-Dar Lin, Chi-Yu Li, Yuan-Cheng Lai, Transparent 3rd-party authentication with application mobility for 5G mobile edge computing, in: 2020 European Conference on Networks and Communications, EuCNC, IEEE, 2020, pp. 219–224.

[26] Andrew Machen, Shiqiang Wang, Kin K. Leung, Bong Jun Ko, Theodoros Salonidis, Live service migration in mobile edge clouds, IEEE Wirel. Commun. 25 (1) (2017) 140–147.

[27] Natsuhiko Sakimura, John Bradley, Mike Jones, Breno De Medeiros, Chuck Mortimore, Openid connect core 1.0, OpenID Found. (2014) S3.

[28] Jorge Navas, Marta Beltrán, Understanding and mitigating OpenID connect threats, Comput. Secur. 84 (2019) 1–16.

[29] MECISG ETSI, Multi-access edge computing (MEC) framework and reference architecture, ETSI GS MEC 3 (2019) V2.

[30] Openairinterface, 2021, https://openairinterface.org/. (Online; Accessed 13 August 2021).

Asad Ali is currently working as a researcher at the National Institute of Cyber Security, Taiwan. He received his Ph.D. degree from National Yang Ming Chiao Tung University (NYCU), Taiwan in 2022. He received his Master degree in Electrical Engineering from National University of Science & Technology (NUST), Pakistan. His research interests are threat intelligence, network security, wireless communications, network design and optimization.

Minhajul Islam received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in the year 2022. He has stepped into research, through working on multi-access edge computing and mobile networking technologies. His research interests include computer networking, security, and applied machine learning.

Tushin Mallick received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in the year 2022. He has stepped into research, through working on multi-access edge computing and mobile networking technologies. His research interests include computer networking, security, and applied machine learning.

Mohammad Sakibul Islam received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in the year 2022. He has stepped into research, through working on multi-access edge computing and mobile networking technologies. His research interests include computer networking, natural language processing and applied machine learning.

Sadman Sakib received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in the year 2022. He has stepped into research, through working on multi-access edge computing and mobile networking technologies. His research interests include edge computing, IoT security, computer networks, and applied machine learning.

Md. Shohrab Hossain is a Professor of Computer Science and Engineering at Bangladesh University of Engineering and Technology (BUET), Bangladesh. He received his Ph.D. degree in Computer Science from the University of Oklahoma, USA in 2012. His research interests include Mobile malware detections, cybersecurity, Software defined networking (SDN), security of mobile and ad hoc networks, and Internet of Things. He has published more than 75 technical research papers in leading journals and conferences. He has been serving as the TPC member of IEEE GLOBECOM, IEEE ICC, IEEE VTC, Wireless Personal Communication, (Springer), Journal of Network and Computer Applications (Elsevier), IEEE Wireless Communications.

Ying-Dar Lin is a Chair Professor of computer science at National Yang Ming Chiao Tung University (NYCU), Taiwan. He received his Ph.D. in computer science from the University of California at Los Angeles (UCLA) in 1993. His research interests include network softwarization, cybersecurity, and wireless communications. His work on multi-hop cellular was the first along this line, and has been cited over 1000 times. He is an IEEE Fellow and IEEE Distinguished Lecturer. He has served or is serving on the editorial boards of several IEEE journals and magazines, and was the Editor-in-Chief of IEEE Communications Surveys and Tutorials (COMST) during 2016–2020.