This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

Transparent 3rd-Party Authentication with Application Mobility for 5G Mobile Edge Computing

Asad Ali, Ying-Dar Lin, *Fellow, IEEE,* Chi-Yu Li, *Member, IEEE,* and Yuan-Cheng Lai, *Member, IEEE*

***Abstract*—Mobile Edge Computing (MEC) is a key technology for supporting low latency applications close to the end user. Users can access application servers in MEC instead of routing to the internet by passing through a core cellular network. Few security challenges arise as the traffic does not traverse through the core network, and these can be solved by providing authentication services in the MEC. However, authentication and application mobility issues arise in the case of multiple MECs where a user is mobile and needs continuous service from application servers, without needing to establish a new session and providing authentication information repeatedly to every new MEC the user connects with. In this work[1], we propose two solutions, a TC3A (Token-based Cookie transfer & 3rd-party Authentication) and a TS3A (Token-based State transfer & 3rd-party Authentication) for resolution of authentication and application mobility issues while achieving low latency. We conducted experiments on a testbed that had MECs deployed in a real-time cellular network (emulated via OpenAirInterface) and performed user handover between two MECs. The experimental results show that TC3A and TS3A successfully re-authenticate the users, without provision of login credentials with target MEC, while reducing the latency by approximately 49.76–59.72% as compared to simple login method. The TC3A and TS3A also eliminate the need of keeping multiple accounts for applications at different MECs and most importantly provide application service continuity, through state transfer during cross-system handover, which is not provided by a simple login method. TC3A provides the application service continuity without any loss of session state, which is suitable for applications that cannot afford state loss, and TS3A provides the same while reducing the latency by 47.05–51.25% as compared to TC3A, which is suitable for applications that require low latency.**

***Index Terms*—Mobile Edge Computing, Multi-Access Edge Computing, Authentication, Mobility, Latency, 3GPP Cellular Networks.**

I. Introduction

MOBILE Edge Computing (MEC), also known as Multi-access Edge Computing, is a new computing paradigm which brings computing powers closer to the user. It can be considered as a cloud computing facility at the edge of the network. The major advantage of MEC is the reduction in latency compared to other computing paradigms. This reduction in latency is very attractive for those applications that need real time processing. Some of the applications that can take advantage of low latency are Vehicle to everything (V2X) communications and virtual reality (VR). MEC is deployed by cellular companies and is compliant with existing 3rd Generation Partnership Project (3GPP) networks, and runs above them. The existing cellular infrastructure supports the deployment and operation of MEC. ETSI [2] and 3GPP [3] are currently building standards for the MEC, so that low latency MEC services can be provided in existing 4G LTE, and in upcoming 5G cellular networks. They have defined use cases, deployment methods [4], architecture, issues, and challenges for the MEC.

A. Authentication and Application Mobility Issues

MEC avoids network traffic congestion and propagation delays by preventing the traffic from entering the core of a cellular network. This allows MEC traffic to bypass the security functions that exist in the core network, which can create security issues. MEC is deployed by a cellular company and is as secure as the core network. However, the applications in MEC can be deployed by third parties which creates further security issues such as extraction of user's radio analytics [5]. Such third-party applications can leverage the existing credentials information of a client in a core network to provide authentication.

Though users can access the applications deployed by MEC, a few issues arise where multiple MECs are deployed by the cellular network, and these MECs are connected to different base stations. In such cases, the issue of mobility arises where, if user equipment (UE) has established a session with an application server on MEC and moves to another place where it has to connect itself to another MEC for the same application, it will have to establish a new session with the application server at the new MEC. Hence, latency will be increased and user experience will be degraded. This can be clearly understood with the help of Figure 1, where two MECs are connected with a 4G LTE core network and a UE moves between them.

Apart from the mobility issue, as a UE moves from the application server on the source MEC (the MEC which already has a session established with UE) to the application server at target MEC (the MEC which needs to continue the service experience for UE), as seen in Figure 1, it will have to repeat the authentication process, in order to gain access to the services provided by the application server at the target MEC.

Asad Ali, Ying-Dar Lin, and Chi-Yu Li are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.

Yuan-Cheng Lai is with the Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan

[1]This work is an extension of [1], with improved survey, real-time cellular network testbed deployment, more results, and detailed security analysis.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

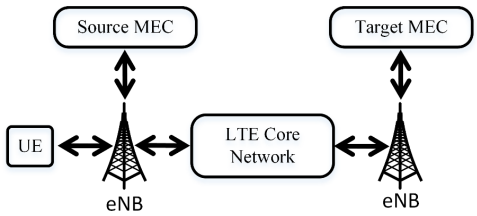


Fig. 1. Multiple MECs deployed in 4G LTE network.

This will introduce further latency in the application access. We therefore identify two major issue in this scenario. First, the need for the UE to authenticate itself with the application server at the target MEC goes against the main purpose of MEC which is to provide low latency services at the edge of the network. Second, the establishment of a new user session with the application server at the target MEC also introduces further latency, as it needs to transfer the session state from the source to the target MEC.

The issues that are identified here lead towards an increase in latency and hence prevent the MEC applications from achieving low latency. In order to achieve low latency, we need to provide a single sign-on (SSO) experience to the UE, in terms of authentication and a seamless transfer, from the application server at the source MEC to the application server at the target MEC. These issues lead us towards the questions to which we arrive at answers in this research. The questions that arise are: Is the seamless transition from the source to target MEC possible through using exchange of state information through UE? And, is it sufficiently secure to allow the UE to carry authentication and state related information?

In the literature there are different types of web tokens available that are responsible for providing a single sign-on experience for a user once they have been authenticated with the server, and do not need to provide their credentials every time, but rather just have to present a web token. There are several types of tokens available, such as JSON Web Tokens (JWT) [6], Simple Web Tokens (SWT) [7], and Security Assertion Markup Language Tokens (SAML) [8]. These tokens can be used and modified for a scenario where a UE has to move from the source MEC to the target MEC. The session state can be stored in the server or the client, depending upon the model, and can also be transferred from the source MEC to the target MEC. There have to be some security measures to ensure confidentiality and the integrity of the information transferred through the user.

B. Token based state and cookie transfer

In this paper, we propose third party authentication of a UE application as the UE moves from the application server at the source MEC to the application server at the target MEC. We apply web tokens as they have already been used in current web applications to provide the single sign on experience. We propose the application of an existing token to this MEC authentication problem, with a modification that includes the session state in the token. In that case, as the UE moves from source to the target MEC, it carries all the necessary information with it, in order to experience the seamless transfer between the two MECs. We name the solutions TC3A (Token-based Cookie transfer and 3rd-party Authentication) and TS3A (Token-based State transfer and 3rd-party Authentication).

The use of a token between the MECs and UE solves the problem of latency, but introduces few security concerns, as it places too much information in the hands of UE; the application at the UE can be malicious and can try to change the contents of a token on its own. Apart from that, UE can also obtain confidential information from the tokens and pretend to be someone else. A UE application can also send malicious packets to the core of the network. These are just a few of the security threats that can arise if we use TC3A and TS3A. In this paper, we will discuss these security threats in detail, and will provide the evidence that TC3A and TS3A are robustly resistant to these potential threats.

We implement TC3A and TS3A on a cellular network testbed and deploy real-time MEC servers inside that cellular network to test the results of TC3A and TS3A. The modules and architecture for this design will be explained in detail in the following sections. The TC3A and TS3A are implemented on a testbed and evaluated for MEC authentication latency. The time taken for the UE to resume the application service is also evaluated. The results show that TC3A and TS3A successfully re-authenticate users, without provision of login credentials with the target MEC, while reducing the latency by 49.76–59.72% as compared to a simple login method which requires provision of login credentials at every new MEC. TC3A and TS3A also eliminate the need to keep multiple accounts for applications at different MECs and, most importantly, provide application service continuity through state transfer during cross-system handover. TS3A provides the application service continuity while reducing the latency by 47.05–51.25% as compared to TC3A. TS3A is suitable for applications that require low latency while TC3A retains the state in its entirety and hence is suitable for applications that cannot afford even a slight loss of state during handover.

The remainder of this paper is organized as follows. Section II covers the background on existing MEC integrated 4G-LTE architecture, along with the threat model and related work. We give our problem statement in Section III with an example as explanation. Section IV documents the proposed design and architecture for TC3A and TS3A. The implementation of a prototype, modules, and testbed is presented in Section V, and Section VI gives the results and evaluation along with security analysis. Section VII concludes the paper along with providing an insight to the future work.

II. Background

This section explores the existing MEC architecture within 4G-LTE and our threat model along with related work.

A. 4G-LTE Exisiting Architecture with MEC

Multi-access Edge Computing (MEC) is an emerging technology to support the new low latency applications. MEC allows the deployment of application servers into a cellular

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

network in multiple ways. It is not limited to 4G or 5G technology but, there are also many new applications for which MEC is suitable, such as the Vehicle to Everything communication (V2X) [9], distributed computing, Virtual Reality, and real-time content delivery. The basic purpose of MEC deployment is to provide processing capabilities near a user and support the applications that have low latency requirements. MEC is normally considered as a feature that will be provided in 5G networks but, the rolling out of 5G across the whole world will take some time, and by that time 4G will still be in operation in many countries around the globe. Therefore, many companies in the industry have focused on the deployment of MEC in existing 4G networks. As 5G networks will be rolled- out around the world, the MEC deployed in 4G networks will be able to support 5G services as well, because of its reference architecture as defined by ETSI [10].

MEC is currently deployed in 4th generation LTE networks through one of four basic scenarios [4]. These scenarios differ from each other in terms of the point of deployment of the MEC host, and are termed as Bump in the Wire, Distributed EPC, Distributed S/PGW, and Distributed SGW with Local Breakout (SGW-LBO). These four scenarios are briefly explained as follows:

1) Bump in the Wire: MEC platform, in this scenario, is deployed at any point between the mobile core network and the base station. Low latency can be achieved in this case by deploying the MEC platform near the base station of the mobile network.

2) Distributed EPC: In this scenario, MEC applications are in the evolved packet core of the cellular network. This can help in saving the deployment cost for the cellular companies as they can make use of the components of EPC as virtual network functions for MEC.

3) Distributed S/PGW: This scenario is somewhat similar to the previous one with the difference that the serving gateway (S-GW) and the packet gateway (P-GW) are deployed at the edge site, while the home subscriber server (HSS) and the mobility management entity (MME) are located in the core of the operator's network.

4) Distributed SGW with Local Breakout (SGW-LBO): This scenario has a new architecture where operator wants to have a greater control over traffic granularity.

MEC can be deployed in core and also as "bump in the wire" to avoid traffic passage through core. We will be using the Bump in the Wire method in which an MEC platform is deployed at any point between the mobile core network and the base station. Low latency can be achieved in this case by deploying the MEC platform close to the base station of the mobile network. The architecture of MEC integrated cellular network applies one of the deployment methods [4] of MEC in a cellular network, and is shown in Figure 2. There are multiple components in this architecture which either belong to the cellular network or the MEC.

B. Threat Model

In this paper, the User Equipment (UE) was assumed to be malicious and could attack the MEC-integrated cellular network. The Core LTE Network was assumed to be authentic and not vulnerable to hackers. The MECs were deployed by the cellular network, and they were also assumed not to be vulnerable. It was also assumed that no MEC would pretend to be another MEC. The UE could be malicious as a result of some UE application. The downlink packets were considered to be safe while uplink packets could be malicious. The packets generated by UE could also be malicious in multiple ways, as it could add extra information or deduct some information. UE could also attempt to fabricate the packets, and malware on UE could send malicious packets to the MECs. There may also be cases where a malicious UE may obtain the information from a genuine UE and pose to be someone else. It is worth mentioning here that the attacks carried out by the malicious UE could easily be detected at the core of the LTE network but in this case the MEC traffic bypassed the core network, and hence there would be some functionality in our proposed design in order to mitigate these threats. If there was no mechanism available to prevent security threats, the whole system would become vulnerable as the UE would be able to penetrate the core network.

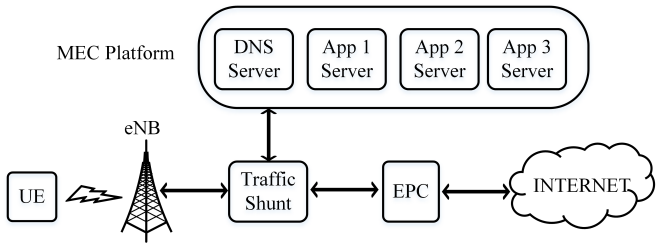


Fig. 2. MEC integrated cellular network.

C. Security Requirements

In light of the threats introduced in the previous subsection, there are some security requirements that must be met in order to ensure the system security. The basic security requirements are mutual authentication, confidentiality, and integrity for the communication between the involved entities i.e., UE, source MEC, and target MEC. Apart from that, we also incorporated the security requirements such as forward secrecy, replay resistance, and Man in the Middle (MITM) attack resistance.

1) Mutual Authentication: This is a very important security property where the involved parties need to authenticate each other to make sure they are about to start the communication with a legitimate party. In our case, UE should be able to authenticate the source and target MECs while, source and target MECs should be able to authenticate the UE before data transmission. The MECs should also be able to authenticate each other.

2) Confidentiality: This is the security property which limits the access to information. In our case, the flow of information between the UE and the source MEC, as well as the UE and the target MEC, must be confidential. Also, the confidentiality for information between different MECs must be ensured.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

3) Integrity: This security property ensures that the information is accurate and trustworthy. In our case, as we have assumed that the UE can be malicious, the proposed authentication protocol must be able to detect if the information provided by the UE to the MECs is authentic or not.

4) Forward Secrecy: This security property ensures an encryption system that frequently keeps changing the keys used for encryption. In our case, TC3A and TS3A must ensure fresh session keys so that if the most recent key is hacked, the rest of the sessions are not affected by it.

5) Replay Resistance: This security property makes sure that if an attacker captures a packet, it should not be able to forward it later to threaten the security. In our proposed security mechanism, we must make sure that replayed messages are detected and discarded by the MECs and UEs.

6) MITM Resistance: Man in the Middle attacks are usually carried out via active eavesdropping where a malicious UE can monitor the communication between legal UEs. Our proposed security mechanism must ensure that the malicious UEs are not able to modify the exchanged messages and send them to the MECs later.

D. Related Work

There are many studies in the literature which look into the mutual authentication between client and server. We have looked into the studies [11-19] which have provided user, handover, or mutual authentication in the MEC environment. We have also looked into a few other studies [20-24] that support application mobility in the MEC environments. These are closely related to our work where we propose authentication and application mobility for MEC. Some of these studies are listed in Table I. It can be seen from this table that, although some of these works focus on the authentication in MECs deployed by the cellular networks and some of them focus on the application mobility, none of them provides both.

Some studies [11-13] focus on the user authentication and mutual authentication with an objective of anonymity via anonymous identity-based scheme and identity-based anonymous authenticated key agreement protocol. Some studies [14,15] focus on handover authentication either by modifying the EAP-AKA protocol or by using SDN for handover of the resource-constrained edge nodes. There are also some studies [16,17] that consider user privacy during mutual authentication for 5G networks by proposing authentication frameworks. Apart from that, Kaur et al. [18] use the Elliptic Curve Cryptography (ECC) to provide lightweight mutual authentication and Zhang et al. [19] propose a blockchain based group signature scheme while addressing the attacks against underlying blockchain. Although these studies provide good solutions for user authentication, mutual authentication, and handover authentication but they do not consider application mobility.

We also looked into the studies that provide application mobility in the MEC environment such as [20] which uses the Open MEC to realize a use case by using Recursive InterNetwork Architecture (RINA) to support the mobility and slicing. Zhang et al. [21] propose application mobility with an objective of privacy preservation while Subramanya et al. [22] improve MEC metrics such as latency. Stauffer et al. [23] provide user application migration between edge servers for the 5G new radio and Ibrahiem et al. [24] provide user application migration for the mobile users in large-scale WiFi networks. The above mentioned studies use different approaches like EPS-AKA, Service Agent, Mobile-IP handoff, unified authentication frameworks, AKE, group signature schemes, SDN, RINA, lightMEC, and container migration for the user authentication, mutual authentication, handover authentication, or application mobility in the MEC environments but none of the studies collectively considers transferring the authentication credentials along with the application state for third-party applications installed on MEC while focusing on transparency.

Our proposed solution is better because it provides a comprehensive solution for the handover authentication and application mobility for third party applications deployed in MEC while considering the transparency. The previous works proposed to address secure handover for authentication in mobile networks mainly focus only on handover authentication with an objective of privacy preservation and user anonymity but do not consider application mobility. Some of the solutions do not take transparency into account as well. Our proposed TC3A and TS3A provide transparency, authentication and application mobility. A transparent solution is really important for deployment, as it does not require any modifications to the existing underlying LTE architecture. The application mobility ensures continuity of the user's application session between multiple MECs, and the user does not have to restart the session. Third-party authentication allows a user to use multiple applications deployed on MEC by third-party service providers, without having to create multiple accounts and providing login information at every MEC server.

III. Problem Statement

The issues that we address here are authentication and the support for application mobility in the MEC-integrated 4G/5G cellular networks. In this section, we will explain the problem, and the assumptions that we have considered for this problem. We will then explain the issues with the help of a hypothetical scenario.

A. Problem Description

We assume we have a source and a target MEC. The source MEC is the one which already provides services to the UE. We assume that the UE is mobile and will become out of the range of the source MEC at some point, and will have to connect itself to another MEC. We call this the target MEC, and is the one which has to provide a service to the UE. The UE already had a session established with the source MEC and we have to transfer the authentication and application information from the application server in the source MEC to the application server in the target MEC, to provide a seamless handover to the UE application client, while achieving low latency.

We assume that the two MECs are connected via an existing 3GPP network and the UE was initially authenticated with the

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

TABLE I
RELATED WORK

Author	Method	Authentication	Transparency	Application Mobility
Li [11]	Anonymous identity-based AKE	✓	X	X
Jia [12]	Mutual authentication with anonymity	✓	X	X
Lee [13]	Anonymous mutual authentication	✓	✓	X
Han [14]	EAP-AKA handover authentication	✓	X	X
Wang [15]	SDN based handover authentication	✓	✓	X
J. Zhang [16]	Privacy-preserving authentication framework	✓	✓	X
Cui [17]	Unified authentication and trust	✓	✓	X
Kaur [18]	lightweight mutual authentication	✓	X	X
S. Zhang [19]	Group signature and authentication	✓	✓	X
Grasa [20]	Mobility, slicing, multi-homing via RINA	X	X	✓
Zhang [21]	MEC enhanced mobility support for IoT	X	X	✓
Tejas [22]	low latency lightMEC	X	✓	✓
Stauffer [23]	5G-NR mobility management	X	✓	✓
Ibrahiem [24]	ARNAB	X	✓	✓
Ours	TC3A, TS3A	✓	✓	✓

source MEC. We also assume that both the MECs belong to the same cellular service provider and the application servers deployed in these MECs can be from different third parties. It is important to note here that we have assumed that the UE will use the same application in both the source and target MEC. As the UE moves from the source MEC, it becomes detached from the source eNB, and thus with the source MEC. Then, it becomes connected with the target eNB, which provides connection to the target MEC. We call this change of eNBs the link layer handover, and we assume that the link layer handover triggers the MEC handover. The UE must access the same application seamlessly from target MEC, and authentication information must be transferred from the source MEC to target MEC without the need of a login. The session state information stored at the source MEC must also be transferred to the target MEC. This all must be done while achieving low latency.

B. Issues

There are a few issues that had to be faced while trying to achieve low latency. These are: How to authenticate the UE with target MEC? How to transfer state information from source to target MEC App server? We will explain these issues with the help of an example. The issues of authentication and application mobility can be understood with the help of the scenario shown in Figure 3, where two MECs are connected to an LTE core network via their respective eNBs. One of the MECs is connected with the source eNB, which we call the source MEC, because we assume that the user is authenticated with this MEC and has established a connection with this MEC in order to access application 1.

This application can be any generic web service, like video streaming or data processing. We assume that the client of the user equipment has established a session with the application server in the source MEC. Now, the scenario is that the user is mobile and moves out of the range of the source MEC, and approaches the target MEC. We call this the target MEC

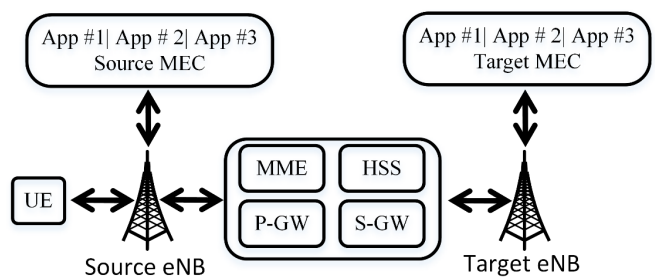


Fig. 3. Example scenario.

because the user now wishes to access the same application in the application server 1 installed in the target MEC. It is worth mentioning here that both the applications in source and target MEC are exactly the same. The first issue that arises here is that of authentication data transfer. If the UE needs to be authenticated again with the application server in the target MEC, it will introduce latency while accessing the application, which is not good for latency sensitive applications.

The second issue is the transfer of session state. Whenever a client establishes a session with a server, the server stores some information related to that session and sends the session cookie to the client, so that the client does not have to send its information every time. The server keeps record of the client through the session state and session cookie. Now, as the client has moved from the source MEC to target MEC, it will establish a new session with the target MEC, and all the information stored in the source MEC application server will not be present in the target MEC application server. We need to transfer this information from the source to the target MEC, so that the user can resume the session with the target MEC application from the exact same position where it left the session with the source MEC application server. We have identified two issues, with this example, which are authentication information transfer and session state information transfer from source to target MEC.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

TABLE II
TC3A AND TS3A COMPARISON

Parameters	TC3A	TS3A
3-p Authentication	✓	✓
Cookie Transfer	✓	X
Session State Transfer	X	✓
Number of Tokens	1	N
Inter-MEC Connectivity	X	✓
App Server Modification	X	X

IV. PROPOSED DESIGN ARCHITECTURE

In this section, we propose a solution to the two problems identified in the previous section. We used the token passing approach for transferring the authentication and state information from the source MEC to target MEC. A token is a small packet that contains some information about the origin and use of the token. A token passing approach is used because it is a safe way to transfer credentials from the server to client. We have modified this token passing approach to solve the problem of transferring authentication and state information between two MECs. The proposed approaches are named TC3A (Token-based Cookie transfer & 3rd-party Authentication) and TS3A (Token-based State transfer & 3rd-party Authentication). Both these approaches make use of the token, and transfer the authentication information and session state information from the source MEC to the target MEC. The token generated in both the approaches consists of the user's information, token expiry time, and the source which generated the token.

This token-based solution solves the problem of authentication and application mobility in the MEC transparently, as it does not involve any changes in the underlying LTE architecture for MEC deployment. If we propose a solution that requires changes in the LTE network architecture, the solution would not be feasible for deployment and will be rejected by the mobile network operators. On the other hand, we can easily tweak the client applications in UE so that they could process the tokens. The differences between these two solution approaches, TC3A and TS3A, are shown in Table II. Third party authentication is provided by both of them while TC3A transfers the cookie from source to target MEC and TS3A transfers the session state from source to target MEC. The number of tokens sent to the client in TC3A is only one while in TS3A, multiple tokens are sent from the source MEC to the client. Target MEC needs to contact source MEC in case of TC3A, while it is not the case with the TS3A. Both TC3A and TS3A do not require any modifications to the application server. The next section outlines the proposed architecture for TC3A and TS3A before we cover them in detail.

A. Architecture

The proposed architecture makes use of Bump in the Wire deployment method of MEC in cellular networks and is shown in Figure 4. There are multiple components in this architecture, which either belong to the cellular network or the MEC. We are concerned with the components which are included in the MEC platform because we have to design them. The most important component is the MEC manager in the MEC. An MEC manager is multi-purpose and communicates with the core cellular network and with the user equipment through the eNB. In the original 4G LTE architecture, eNB is directly connected to the EPC while here in Bump in the Wire method, an MEC platform is deployed between the eNB and EPC. The MEC manager takes care of the traffic and directs it to the MEC App server or the EPC accordingly.

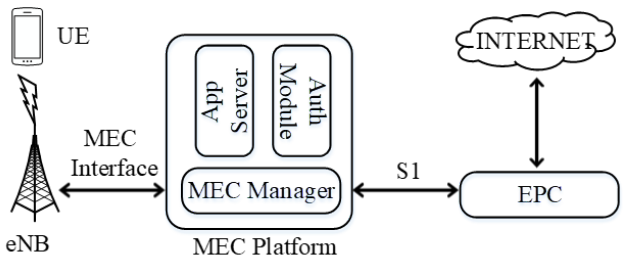


Fig. 4. MEC integrated cellular system architecture for TC3A and TS3A.

The MEC manager is responsible for checking the traffic that goes in and out of the MEC platform. It will also obtain authentication information about the user from the core network and send it to the authentication module, which will store the information. The authentication information can either be generated by data prefetching for the UE from the Internet servers if the UE has already been authenticated with the HSS [25] or it can be obtained from the UE via sign up at an application deployed at the MEC. The MEC manager also communicates with the application server in order to obtain the cookie and state information for both the proposed solutions which are TC3A and TS3A. The MEC manager will then generate the token that will be presented to the user equipment. It is also responsible for the security of the token and for providing the necessary measurements to keep the token secure. The messages exchanged between these modules are explained in next sub-section and are shown in Figures 5 and 6.

B. Protocol

We propose two solutions for the transfer of authentication information and session state information between a source and a target MEC. We only explain the message flows of these solutions here and do not provide the cryptographic details for the ease of understanding. Those details are provided in the section IV.C.

1) TC3A (Token-based Cookie transfer & 3rd-party Authentication): This is represented in Figure 5, and it can be divided into three stages: a source MEC stage, a handover stage and a target MEC stage. In the source MEC stage, the UE interacts with the Source MEC (MEC A) and establishes a session with the application server in the source MEC. We term this the source MEC because we assume that the UE has been authenticated with this MEC via mutual authentication that will be explained later, and has already an established session with this MEC. As soon as the session is established, the application server forwards the session cookie to the UE,

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

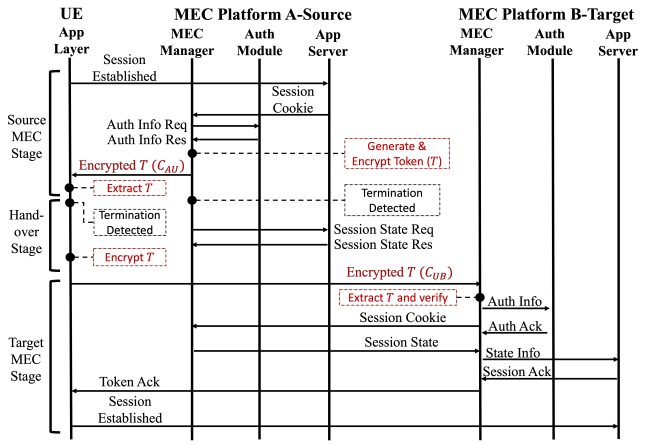


Fig. 5. Token-based Cookie transfer and 3rd-party Authentication message flow.

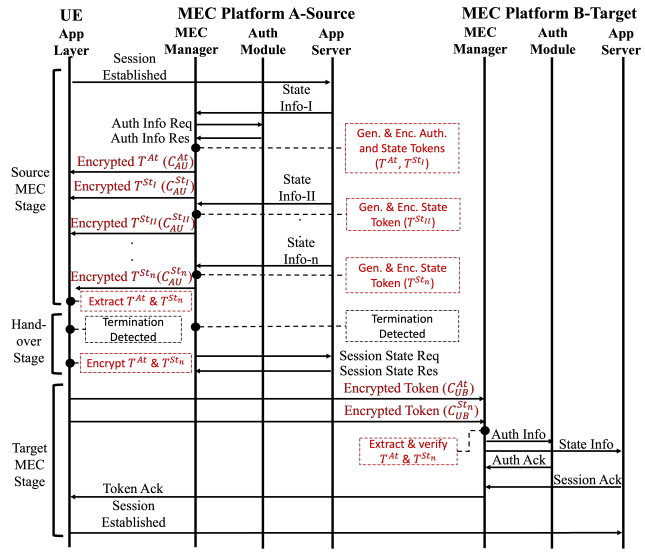


Fig. 6. Token-based State transfer and 3rd-party Authentication message flow.

which is then captured by the MEC manager. As explained in the architecture section, all the information between the UE application client and MEC application server goes through the MEC manager which handles the process of information exchange. As soon as the session cookie has been captured by the MEC manager, it sends a request to the authentication module to obtain the user's authentication information.

When the MEC manager receives the authentication information from the authentication module, it creates a token "T" on the basis of this information. This token contains the issuer, audience, timestamp, cookie and a verification field ("*vrfy*"). The MEC manager encrypts this token "T" via symmetric key encryption to create the encrypted message C_{AU}. The message C_{AU} represents a ciphertext sent from MEC A to the UE U. The MEC manager presents this encrypted message C_{AU} to the UE, which decrypts it via the symmetric key and extracts the token "T". The details of the encryption process, token "T", and message C_{AU} will be explained in the next section. As soon as the user moves away from the vicinity of the source MEC, both the application client and application server will detect session termination and will react accordingly. The MEC manager of source MEC then requests the session state information from the application server by presenting it with a cookie, and the application server sends the information of the session to the MEC manager, where it is saved. This is the handover stage. As the UE is handed over from source MEC (MEC A) to target MEC (MEC B), the application client in UE exchanges key information with the MEC manager of MEC B, and prepares the message C_{UB} by encrypting the token via symmetric key. The details of the encryption process and message C_{UB} will be explained in the next section.

For the target MEC stage, the user equipment presents the message C_{UB}, which contains the token "T", to target MEC, to be received by the MEC manager. The MEC manager then decrypts C_{UB} to extract token "T", and verifies it. After verification, it extracts the authentication information and session cookie from the token, and sends the authentication information to the authentication module of the target MEC, which then authenticates the user on the basis of the information. The MEC manager also sends the session cookie to the MEC manager of the source MEC. The source MEC manager already has the state information and when it receives the session cookie, it sends the session state information to the MEC manager of the target MEC. As the session state information is received, the user has already been authenticated with the target MEC, so this information is then passed on by the MEC manager to the application server in the target MEC, which responds with a session ACK message. After that, the MEC manager informs the application client in UE about the authentication and session ACK via token ACK, and hence a session is established between the application client and application server in the target MEC.

2) TS3A (Token-based State transfer & 3rd-party Authentication): This is represented in Figure 6. In the source MEC stage, the application client establishes a session with the application server in the source MEC (MEC A), so termed because we assume that the UE has been authenticated with this MEC and already has an established session with this MEC. As soon as the session is established, the application server sends the session information to the MEC manager. As soon as the session information is received by the MEC manager, it sends a request to the authentication module to obtain the authentication information about the user. The authentication module has the authentication details and responds to the MEC manager with the authentication information.

The MEC manager now has the session state and the authentication information, and creates an authentication token T^{At} on the basis of the authentication information, encrypts this token via symmetric key to create the message C_{AU}^{At}, and presents it to the UE. Note that the MEC in this case does not include the state information in the authentication token as it is not the complete information. The application server will periodically keep sending the state information. The MEC manager will keep generating the state tokens (T^{St_I}, $T^{St_{II}}$,.., T^{St_n}) periodically until the UE becomes detached from the application server in the source MEC. All the state tokens are also encrypted like the authentication token to create the

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

messages ($C_{AU}^{St_I}$, $C_{AU}^{St_{II}}$,.., $C_{AU}^{St_n}$). These encrypted messages are sent to the UE which extracts the authentication and state tokens after decryption. The details of token creation and encryption/decryption will be explained in the next section. The application client and application server will detect the session termination and the protocol will enter the handover stage. The application server will remove the state information and the user will move away from the vicinity of the source MEC. As the UE is handed over from source MEC (MEC A) to target MEC (MEC B), the application client in UE exchanges key information with the MEC manager of MEC B, and prepares the messages C_{UB}^{At} and $C_{UB}^{St_n}$ by encrypting the authentication and state tokens via symmetric key.

The next stage is the target MEC stage where user equipment presents the messages C_{UB}^{At} and $C_{UB}^{St_n}$ to the target MEC, which is received by the MEC manager of the target MEC. The manager of the target MEC decrypts these messages, extracts and verifies the authentication and state tokens. The MEC manager sends the authentication information to the authentication module of the target MEC, and the authentication module of the target MEC will authenticate the user on the basis of that information, and send an ACK to the MEC manager. The MEC manager then sends the state token to the application server which responds with an ACK. After that, the MEC manager informs the application client in UE about the authentication and session state information ACK via token ACK and hence a session is established between the application client and application server in the target MEC.

C. Cryptographic Details of TC3A and TS3A.

1) Security Notions: Figures 5 and 6 show the message flows of TC3A and TS3A respectively, but do not provide the complete insight on the cryptographic details of the protocols, tokens generation, C_{AU}, C_{UB}, C_{UB}^{At}, and $C_{UB}^{St_n}$ generation due to complexity. In this section, we provide all these details and explain the concrete construction of TC3A and TS3A by using two security notions in this work. i) The first security notion is a security primitive known as the discrete logarithm problem (DLP) where we have a group G. In the group G, given g, p, and x such that there exists n such that $x = g^n (mod\ p)$, if an adversary knows g, p, and x, it is hard to guess n. ii) The second security notion is a security scheme known as the symmetric key encryption (SKE) which consists of key generation, encryption, and decryption. We have two key generation algorithms as $KeyGen_{MEC-UE}^{SKE}(1^n)$ and $KeyGen_{MEC-MEC}^{SKE}(1^n)$ that generate symmetric keys for MEC-UE communication and MEC-MEC communication respectively. Each MEC has its own key (private), represented by α, β, and γ etc. Each UE also has its own key (private) which we represent as u. These keys are private to each MEC and UE, and they are used to generate the public intermediate keys and common keys for the MEC-UE communication. The generation of the public intermediate keys and common keys for the MEC-UE communication will be explained later. The common secret key *K*, which is a symmetric secret key, is used for the encryption and decryption of the messages exchanged between the MECs. The private keys (α, β, γ, and u) and public intermediate keys (g^α, g^β, g^γ, and g^u) are generated by each MEC and UE. We deploy a trusted Certificate Authority (CA) to ensure the authentication of public keys securely. Each MEC and UE create a Certificate Signing Request (CSR), the CA verifies the identity of each MEC and UE, and signs their CSR, generates and distributes digital certificates to each respective MEC and UE. These certificates bind the public intermediate keys of each MEC and UE to their verified identities and are signed using the CA's private key. Later, when MECs and UE exchange public intermediate keys (g^α, g^β, g^γ, and g^u) with each other, they can authenticate these keys through CA's public key. The common secret key *K* (key for MEC-MEC communication) is distributed among all the MECs by the MEC orchestrator. The public intermediate keys generate common keys for the MEC-UE communication to encrypt and decrypt all the messages exchanged between MECs and UEs. The private keys α, β, γ etc., and u are a composite of 8 variables of 32 bits each. Hence, each of them is 256 bits long and for the symmetric key encryption, we use AES-256 in CBC mode with IV (Initialization Vector).

TABLE III
DATABASE SAVED WITH EACH MEC MANAGER

MEC	Private Key	Public Intermediate Key
A	α	g^α
B	β	g^β
C	γ	g^γ
D	δ	g^δ
.	.	.
Z	ω	g^ω

2) Explanation: We explain the cryptographic details of TC3A in this subsection. In our model, we have considered all MECs to be safe as they are deployed by the 3GPP MNO. The details which are private to each MEC and UE are the private keys α, β, γ, u etc. The details which are public for all the MECs and UEs are the group $G = 1, ..., p-1$ where P is a Prime Number. The generator $g \in G$ is used along with the keys α, β, γ to generate the public intermediate keys. The generator is such that $g = g^0 \neq g^1 \neq ... \neq g^{p-2}$ and $\forall x \in G$, $\exists\ n \ni g^n = x$. The keys $(\alpha, \beta, \gamma, .., \omega)$ are used with the generator $g \in G$, to create the public intermediate keys (g^α, g^β, g^γ,..,g^ω), that are later used for two purposes. One is to generate the common keys for encrypting the MEC-UE communication and the other is to serve as the identity of the MEC and UE. The common secret key (*K*) is used for encrypting the MEC-MEC communication and it is only known among MECs and not known to any UE. The public intermediate keys g^α used by MEC A, g^β by MEC B, and g^u by the UE are all public, and they are used to generate the common keys for each pair of MEC-UE communication. Each MEC represented by alphabets, as shown in Table III, has its own private key (α, β, γ etc.), and a public intermediate key (g^α, g^β, g^γ,..,g^ω).

In TC3A, the MEC manager of MEC *A* receives the session cookie from the application server, and asks the authentication module about the authentication information via Auth Info

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

Req. The authentication module responds with the Auth Info Res, that contains the identity of the issuer which is the public intermediate key g^{α} of MEC *A*, and the identity of the audience which is the public intermediate key g^{u} of UE (represented by U). The MEC manager generates a token "T" which contains the identity of issuer (g^{α}), the audience (g^{u}), the cookie, a timestamp, and a field which we call as $vrfy$. The field $vrfy$ contains the identity of the issuer g^{α}, the identity of UE (g^{u}), the cookie, and the timestamp, encrypted with the common secret key (*K*). This field can only be decrypted by the entity which has the common secret key (*K*) (i.e. all other MECs as UEs do not have this common secret key). Also, we have assumed the MECs to be non-malicious, and they do not pretend to be other MECs. Hence, every MEC encrypts the field $vrfy$ with the common secret key. The field $vrfy$ has the following information:

$$1.\ vrfy = ENC_{K}^{SKE}(g^{\alpha}||g^{u}||timestamp||cookie)$$

Hence, the token generated by the MEC *A* for the UE *U* have the following information:

$$2.\ T = (g^{\alpha}, g^{u}, timestamp, cookie, vrfy)$$

The MEC *A* sends its public intermediate key g^{α} to the UE and the UE sends its public intermediate key g^{u} to the MEC *A*. They both calculate the common key $g^{\alpha u}$. The MEC *A* encrypts the token *T* with the common key $g^{\alpha u}$ as follows:

$$3.\ C_{AU} = ENC_{g^{\alpha u}}^{SKE}(T)$$

This message C_{AU} is sent from MEC *A* to the UE, which decrypts it using the common key $g^{\alpha u}$, and obtains the token which has the following information:

$$4.\ T = (g^{\alpha}, g^{u}, timestamp, cookie, vrfy)$$

where $vrfy = ENC_{K}^{SKE}(g^{\alpha}||g^{u}||timestamp||cookie)$

The UE then moves from MEC *A* to MEC *B*. During this handover, the MEC manager of MEC *A* requests the session state information from the application server by presenting it with a cookie, and the application server sends the information of the session to the MEC manager, where it is saved. As soon as the UE connects with the new MEC *B*, they exchange their public intermediate keys. The MEC *B* sends its public intermediate key g^{β} to the UE and the UE sends its public intermediate key g^{u} to the MEC *B*. They both calculate the common key $g^{u\beta}$. The UE creates the message C_{UB} as follows and sends it to the MEC *B*.

$$5.\ C_{UB} = ENC_{g^{u\beta}}^{SKE}(g^{\alpha}, g^{u}, timestamp, cookie, vrfy)$$

where $vrfy = ENC_{K}^{SKE}(g^{\alpha}||g^{u}||timestamp||cookie)$

MEC *B* decrypts the message C_{UB} using the common key $g^{u\beta}$. MEC *B* extracts the token and sees the issuer and audience along with the $vrfy$.

$$6.\ T = (g^{\alpha}, g^{u}, timestamp, cookie, vrfy)$$

The g^{α} tells the MEC *B* about the issuer of this token. The g^{u} tells the MEC *B* about the intended audience of this token. The "vrfy" can only be decrypted by using the common secret key (*K*) shared among MECs. Only MECs can decrypt this, and it cannot be decrypted by the UEs. Once MEC *B* decrypts "vrfy", it verifies that this message originated from MEC *A* and was intended for the UE that has presented the token. The MEC manager of MEC *B* sends the "Auth Info", i.e. issuer and audience, to the authentication module and receives the "Auth Ack" in return. The MEC manager of MEC *B* sends the cookie to the MEC manager of MEC *A* and obtains the session state information for the UE from MEC *A*. The MEC manager of MEC *B* forwards the "State Info" to the application server which responds with the "Session Ack", and the session is established between the UE and MEC *B*. The same cryptographic notions are used for the TS3A as well. The only difference between the two is that in TS3A, we need to create a separate authentication token T^{At} which contains the issuer and audience via using the "Auth Info Req" and "Auth Info Res" message exchange. We also need multiple state tokens (T^{St_I}, $T^{St_{II}}$, .., T^{St_n}) which are generated after the MEC manager of MEC A receives the State Information from the application server periodically (State Info-I, State Info-II, .., State Info-n). The rest of the steps and the process of encryption/decryption and verification is the same as TC3A.

V. Implementation

In this section, we explain the prototype architecture, modules, and the testbed on which experiments were performed.

A. Prototype Architecture

The implementation was carried out with the deployment of a cellular network with the help of Open Air Interface. Two MECs were deployed in the cellular network by using Bump in the Wire approach as explained above. One of the MECs was set up as the source MEC, and the other as the target MEC server. We deployed the UE on a laptop and connected it to the source MEC, and then moved it to the target MEC. Once the authentication was complete, the UE obtained access to the resource sever. The resource server, which we chose for implementation, was an open source and cross-platform video streaming server called VideoLAN Client (VLC). We assumed that the user was mobile and needed to access the video streaming server (VLC) in order to have video content delivered to the device.

We deployed the same VLC video streaming server in both the source and target MECs. In the experiment, the deployed UE established a connection with the source MEC and accessed the video streaming server in the source MEC, and then established a session with the video streaming server which stored the state of the session and provided token to the UE. Once the video streaming started, the UE was moved from the source MEC to the target MEC. The underlying cellular network let the UE know that the eNB has been changed now and hence the source MEC is no longer available. The UE then established a connection with the target MEC which had the same video streaming server, but it needed authentication first. The authentication was provided through the token that was granted to the UE by the Auth module of the source MEC. The session state was transferred after the authentication had been completed, and hence the session with the target MEC was established from the same place where it got disconnected with the source MEC.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

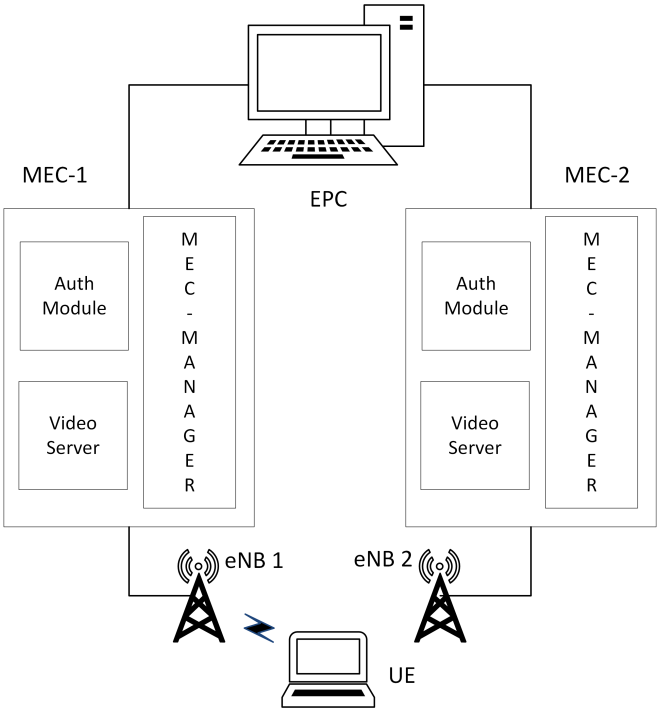


Fig. 7. Experiment Architecture.

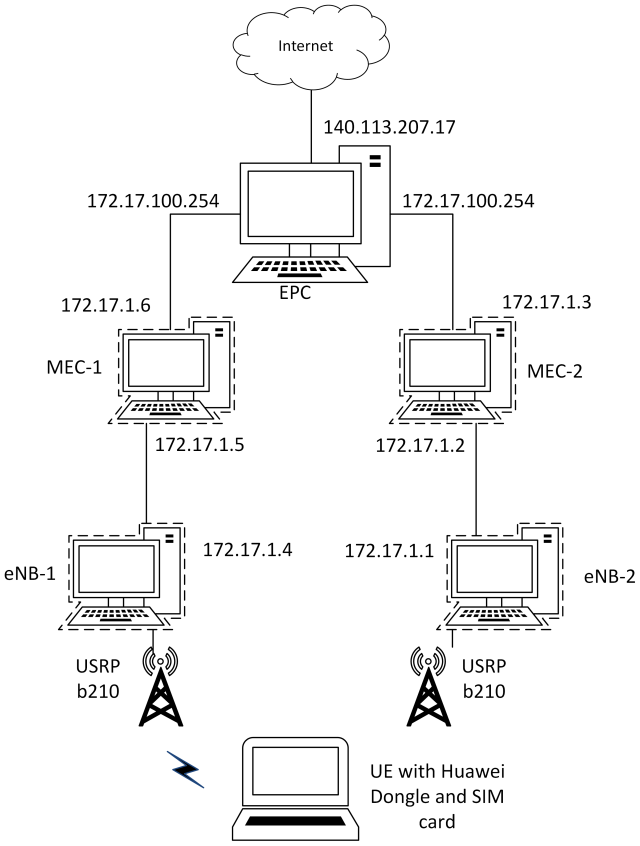


Fig. 8. Experimental Testbed.

B. Modules

The modules designed for the solution implementation are an authentication module and App server together with an MEC manager. The MEC manager has to manage the other two modules and the authentication module handles all the authentication information in both the source and target MECs. The MEC manager is responsible for generating the auth token which can be used to access the video streaming server. Figure 7 also shows the auth module and the video server module which are managed by the MEC manager.

C. Testbed

The testbed for the experiment consisted of 5 PCs which worked as Evolved Packet Core (EPC), eNodeB 1, eNodeB 2, source MEC, and target MEC and 1 laptop which worked as the UE. These 5 PCs together created a cellular network with two eNodeBs along with their respective MECs. OpenAirInterface (OAI) implementation was used to deploy the cellular network. The EPC consists of the Home Subscriber Server(HSS), Mobility Management Entity(MME), and SPGW. The SPGW interface was connected with the internet so that the cellular system could provide internet access to the connected users. The eNodeBs were deployed with the help of USRP b210 so that they could communicate wirelessly with the UEs. The machines in which MECs were deployed were provided with two network interfaces, one connected to the EPC and the other connected to the eNodeB.

The UE was created by using a laptop, a Huawei Dongle E3372, and a programmable SIM card. Huawei Dongle E3372 is a high-speed 4G dongle which can be connected to any laptop in order to provide cellular network services. Huawei Dongle E3372 needs a SIM card in order to convert the laptop into a UE. Therefore, a SIM card was programmed with the details of the user provided in the database of the HSS. The UE was successfully able to access the internet via connection to eNodeB, to MEC, to EPC, to internet. The testbed is shown in Figure 8, and it consists of the 5 PCs which worked as UE, eNodeB 1, eNodeB 2, source MEC and target MEC. The details of machines that were used as the EPC, eNBs, source MEC, target MEC and UE are given in the Table IV.

TABLE IV
TESTBED DETAILS

Entity	System	Components
EPC	CPU: Intel Core i7-8700 3.20 GHz RAM: 16 GB	OAI HSS, MME, SPGW
eNB-1	CPU: Intel Core i5-3470 3.20 GHz RAM: 32 GB	OAI eNB USRP b210
eNB-2	CPU: Intel Core i7-8700 3.20 GHz RAM: 16 GB	OAI eNB USRP b210
MEC-1	CPU: Intel Core i7-8700 3.20 GHz RAM: 16 GB	Auth Module App Server-1
MEC-2	CPU: Intel Core i7-8700 3.20 GHz RAM: 16 GB	Auth Module App Server-2
UE	CPU: Intel Core i5-6200 2.4GHz RAM: 8 GB	Huawei Dongle SIM card

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

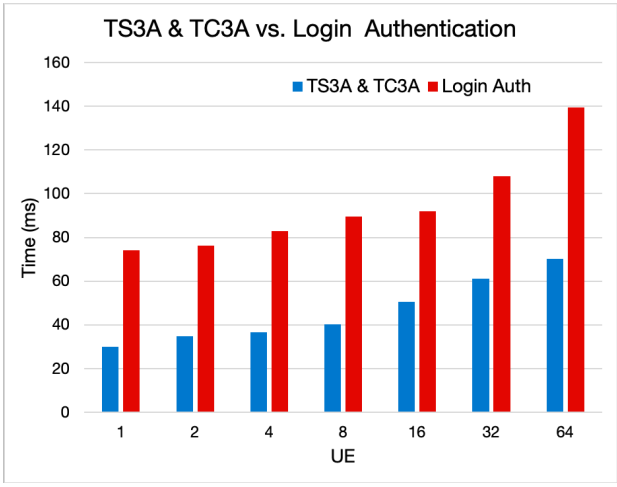


Fig. 9. Authentication Latency at target MEC.

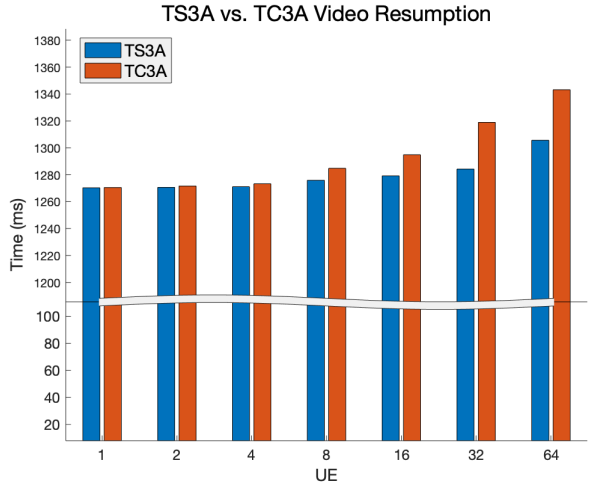


Fig. 10. Video Resumption Latency.

VI. Results and Evaluation

We performed our experiment on the testbed, as shown in Figure 8. The proposed TC3A and TS3A provided authentication to users without the need of keeping multiple accounts, while providing state transfer during cross-system handover at reduced latency as compared to a simple login method. The major purpose is to provide seamless handover to the UE, not to reduce the latency. We obtained several results related to the latency to prove that our proposed method does not incur any extra latency. The results are documented in the next subsections.

A. Authentication Latency at Target MEC

We obtained five sets of results from the experiment. The first result was the comparison between the Authentication time taken by the target MEC for login method and our proposed TC3A and TS3A. When a user authenticates with the source MEC for the first time, it needs to provide its credentials. The server provides a token to the user upon authentication, and the user sends this token to the target MEC, and hence achieves low latency. The graph shown in Figure 9 does not include the handing over of user from source to target MEC as it only compares the difference between the authentication time at target MEC with login credentials and authentication time at target MEC with TC3A and TS3A. TC3A and TS3A take the same time for authentication at the target MEC and the difference between them will be explored below. The results in Figure 9 show that the authentication time taken by the target MEC while using TC3A and TS3A is 59.72% less than the login authentication for 1 user and is 49.76% less for 64 users.

B. Video Resumption Latency

We were also interested in finding out the time difference between the two solution approaches, TS3A and TC3A, for resuming a video under different number of UEs, once authentication had been completed. Note that this result only shows the video resumption time at the target MEC and does not include the authentication time at the target server, which will be included in the results in next subsection. Figure 10 shows that TS3A takes less time than TC3A in resuming the video for the user, because in TC3A the target MEC needs to contact the source MEC in order to get the user state, while there is no need to contact the source MEC in the case of TS3A. The difference between TS3A and TC3A is not considerable when number of UEs is less. This difference gradually increases as the number of UEs increase. The reason is that for each new UE, the target MEC has to fetch its state from the source MEC. As the number of UEs increases, the number of connections to the source MEC for fetching the UE state also increase, and hence the difference between TC3A and TS3A increases. Note that the y-axis has been broken between 100-1200 ms in order to clearly show the increase in time as the number of UEs increases.

TS3A reduces the latency by 2.70% compared to TC3A when there are 64 users. This does not seem like a considerable difference but, the interesting thing to note here is that we are using VLC for video streaming and the VLC takes approximately 97–99% of the time shown in the bars of Figure 10 and hence the difference between TS3A and TC3A seems negligible. This time is taken by the VLC to deliver the video after the request from the UE, and we cannot reduce this time as it depends upon the type of service provided by a service provider. Therefore, we present another result to emphasize the difference between TC3A and TS3A. Figure 11 shows the state fetching time for TS3A and TC3A, the time taken by TS3A and TC3A before they send a request to the VLC media server for a video stream. The results show that TS3A reduces the latency by 47.05–51.25% as compared to TC3A which makes it suitable for applications that require lower latency. TC3A is suitable for those applications which do not have strict latency requirements but cannot afford the loss of state as TC3A retains the state in its entirety.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

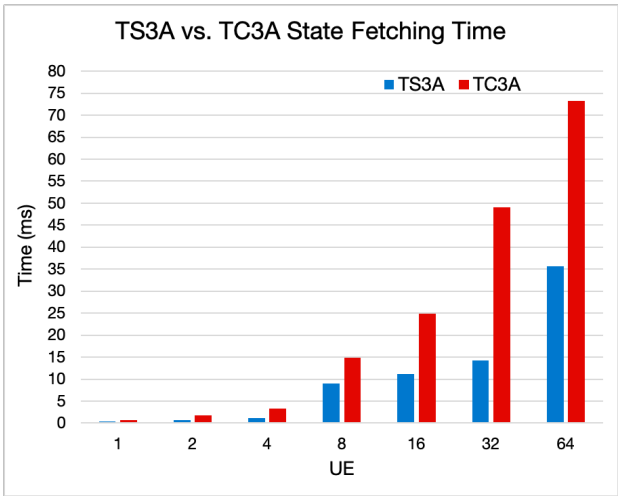


Fig. 11. State Fetching Latency.

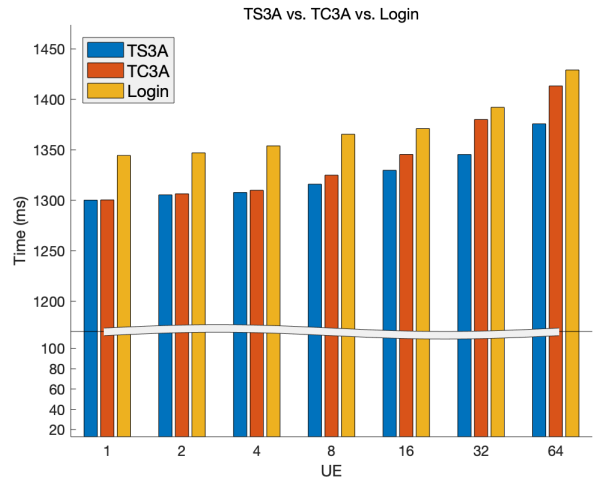


Fig. 12. TS3A v TC3A v Login.

C. TS3A vs. TC3A vs. Login Latency at Target MEC

Another interesting result is the time taken by the Target MEC to authenticate the user and resume the video for the user from where the user left off watching it in the source MEC. This comparison is shown in Figure 12 where it can be clearly seen that the TS3A authenticates the user with the target MEC and resumes the video with the minimum latency under different number of UEs. TC3A takes longer than TS3A, but the advantage of TC3A is that the video is resumed from the exact same place where it was terminated in the source MEC. We conducted the experiment for the transfer of video state but this feature of TC3A retains the state in its entirety and hence TC3A is useful for such applications that cannot afford the state loss. Although, there was not much difference between the latency of TC3A and the simple login method, it should be noted that TC3A also provides application mobility which enables the video to be resumed from the exact same place in target MEC where it was terminated in the source MEC. This service is not available in the simple login method. Note that the Figure 12 includes both the authentication time and video resumption time at the target MEC under different number of UEs. Also, the y-axis has been broken between 100–1200 ms in order to clearly show the increase in time as the number of UEs increases.

D. TS3A vs. TC3A vs. Login Latency Profile

We also obtained the latency profile for the three methods and assessed the time taken by the different segments. The results are shown in Figure 13, where it can be seen that initially all three methods take similar time in the source MEC. This is because of the reason that the UE initially attaches to the eNB and there is no difference in these 3 methods for UE attach procedure. Hence, the time for UE Attach is the same in all three cases. In all three cases, the authentication with source MEC takes place through login, and hence the time for this stage is also same for all three methods. However, as they move from the source MEC towards the target MEC, the login method needs to provide credentials again, and hence the

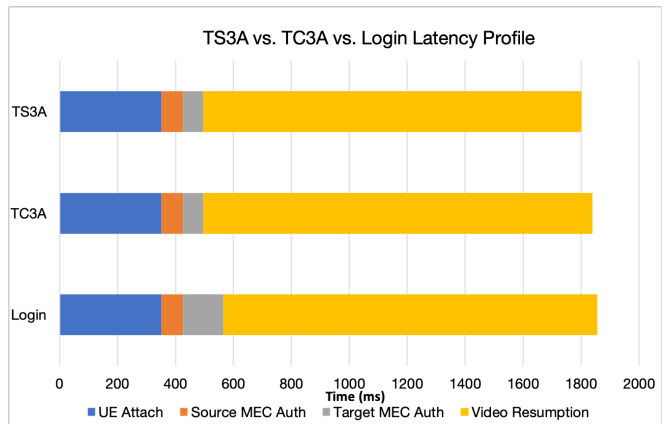


Fig. 13. Latency Profile.

third segment takes more time compared to the other segments. Figure 13 shows that TS3A and TC3A reduce the latency compared to a simple login method. It should be noted that application handover via TC3A and TS3A takes more time than a simple login method. This is so because Login method does not provide application service continuity and hence it does not need to perform the state fetching like TC3A and TS3A which fetch the state from source MEC via two different methods to provide seamless application handover to the user.

E. Latency Breakdown

We furthermore obtained a detailed message breakdown for latency. The results obtained for TC3A are shown in Figure 14, from which it can be seen that first 4 messages complete the UE authentication with the HSS and among these 4, second step introduces the latency and is the bottleneck for this procedure. This step is a standard EPS-AKA protocol message and we cannot do anything to reduce this latency. After the first 4 steps, source MEC authentication is the fifth step which does not take a considerable time. Figure 14 also shows that more time is taken by the video server to start streaming compared to the other steps. We cannot reduce this

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

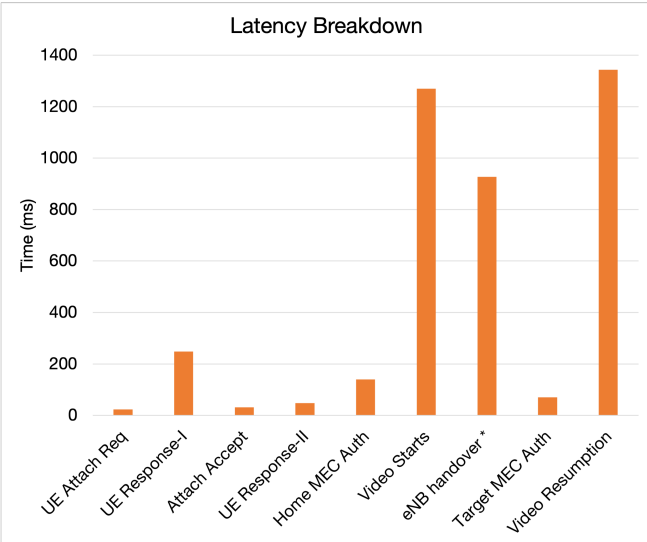


Fig. 14. Latency Breakdown.

time as it is the standard time taken by the VLC media server to start streaming the video over HTTP/TCP using android low quality encoding. The next step is eNB handover* which takes approximately 926 ms [27] for X2 handover. We could not calculate the X2 handover in this experiment as OAI EPC does not currently support X2 handover. After the eNB handover, the UE authenticates itself with the target MEC and this is where we have reduced the time so that UE would not have to provide credentials again. After authentication with target MEC, the video streaming service is resumed from the exact same place where the UE left off watching video in the source MEC. Note that video resumption takes more time than video start in the source MEC as it has to fetch the state for video prior to sending request to the VLC media server in target MEC.

VII. Security Analysis

In this section, we examine how TC3A and TS3A provide guarantees for the previously identified security requirements and explain how they defend against possible attacks.

A. Security Guarantees

1) Mutual Authentication: Mutual authentication is an important security requirement, and it is done via public intermediate keys. The MEC A and the UE exchange their public intermediate keys g^{α} and g^{u} with each other. Only the legitimate UE can calculate $g^{\alpha u}$ because u is private. Once the UE decrypts via using $g^{\alpha u}$, it will find out g^{α} and g^{u} inside the message. This information is publicly available and hence, both the MEC and the UE can mutually authenticate each other. No malicious UE can calculate $g^{\alpha u}$ even if it knows g^{α}, g^{u}, or both. Now if a malicious UE sits in the middle, and poses as MEC A, it will receive g^{α} from MEC A and g^{u} from the valid UE. It will not be able to calculate $g^{\alpha u}$ due to the discrete logarithm problem. Hence, it cannot generate a valid C_{AU}, that can be decrypted by the valid UE using $g^{\alpha u}$, which contains valid information about two entities in the communication.

2) Confidentiality: Confidentiality is one of the most important security requirements [28]. The confidentiality of messages in our proposed solution is guaranteed if the session key exchange security is guaranteed. We provide security proof of the key exchange as shown in Theorem VII.1.

Theorem VII.1. *Any instance of our solution is as secure as the Discrete Logarithm Problem (DLP), except with negligible probability.*

Proof. The security game of our solution is given by Table V.

Challenge		**Adv**
$\alpha, g^{\alpha}(\text{mod } p)$		
$u, g^{u}(\text{mod } p)$		
	$\xrightarrow{g^{\alpha}(\text{mod } p),\ g^{u}(\text{mod } p)}$	
	$\xleftarrow{\gamma}$	
Adv wins if $g^{\alpha u} \equiv g^{\gamma}(\text{mod } p)$		

TABLE V
CONSTRUCTION: **ADV** WINS IF $g^{\alpha u} \equiv g^{\gamma}(\text{MOD } p)$.

Now construct the adversary **Adv**' as shown in Table VI.

Challenge		**Adv′**		**Adv**
$\alpha, g^{\alpha}(\text{mod } p)$				
	$\xrightarrow{g^{\alpha}(\text{mod } p)}$			
		$u = 1, g^{1}(\text{mod } p)$		
			$\xrightarrow{g^{\alpha}(\text{mod } p),\ g^{1}(\text{mod } p)}$	
			$\xleftarrow{\gamma}$	
	$\xleftarrow{\gamma}$			
Adv' wins if $g^{\gamma} \equiv g^{\alpha}(\text{mod } p)$				

TABLE VI
CONSTRUCTION: **ADV'** WINS IF $g^{\gamma} \equiv g^{\alpha} mod p$.

Assume that Probability:

$$Pr\left[\textbf{Adv} \text{ wins}\right] \geq \mu(n) \tag{1}$$

where $\mu(n)$ is non-negligible. Note that:

$$g^{\gamma} \equiv g^{\alpha u}(\text{mod } p), \tag{2}$$

but $u = 1$, so:

$$g^{\gamma} \equiv g^{\alpha}(\text{mod } p), \tag{3}$$

Then notice that:

$$\begin{aligned} Pr\left[\textbf{Adv}' \text{ wins}\right] &= Pr\left[g^{\gamma} \equiv g^{\alpha}(\text{mod } p)\right] \\ &= Pr\left[g^{\gamma} \equiv g^{\alpha u}(\text{mod } p)\right] \\ &= Pr\left[\textbf{Adv} \text{ wins}\right], \end{aligned} \tag{4}$$

and by using Equation (1) in (4), we get:

$$Pr\left[\textbf{Adv}' \text{ wins}\right] \geq \mu(n). \tag{5}$$

Therefore, **Adv**' wins with non-negligible probability, i.e., **Adv**' can break the DLP ($\rightarrow\leftarrow$). □

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

3) Integrity: We divide the messages flow into four possible stages; communication between the UE and the source MEC platform (C1), handover of UE from source eNB to target eNB (C2), communication between UE and target MEC (C3), and communication between source MEC and target MEC (C4). TC3A and TS3A must support *integrity* across these four stages.

C1: As we tested our proposed TC3A and TS3A for the MECs deployed in 4G-LTE cellular networks, the security features provided by the 3GPP apply to our solutions as well. There are multiple security levels provided by the 3GPP such as network access security, network domain security, user domain security, and application domain security [29]. The network access security protects against various attacks on the radio link and hence, makes the communication between UE and the EPC safe. This makes our radio link secure for communication with MEC as well because the traffic for MEC goes through the same radio link. The application domain security provides the security features that allow the UE applications to securely exchange messages with the application servers deployed by the providers.

Apart from that, to ensure the integrity of C1, when the MEC A generates a token, it creates a signature $vrfy$ for the target MEC, includes the MEC identity information g^{α}, identity information g^{u} for the intended UE, timestamp and cookie and encrypts them with the common secret key *K*. The common secret key *K* is not shared with the UE and hence, it cannot see the contents of $vrfy$. If the UE modifies the contents of the token, it can be detected later at the C3 stage when the token is presented to the target MEC. This ensures the *integrity* over C1.

C2: Stage C2 involves the handover of the UE from source eNb to the target eNB. The security of this stage is ensured by the 3GPP handover security. When the UE moves between two eNBs that belong to the same MME, both the MME and the UE derive the eNB key (K_{eNB}) along with the next hop (NH) parameter and during such a handover, a new session eNB key (K^{*}_{eNB}) is derived, to be used between the UE and the target eNB, from the already active (K_{eNB}) and the NH parameter [29]. There are no tokens exchanged by the TS3A and TC3A during handover.

C3: Stage C3 involves the communication between the UE and the target MEC. After C2, handover is completed and the UE gets connected to the target eNB and consequently, the target MEC which we refer as MEC B in section IV.C. Here the UE needs to communicate with the target MEC over the radio link via eNB. Again, network access security provided by 3GPP comes into play and secures this radio link [29]. Also, the token generated by the MEC A contains a field $vrfy$, to ensure the *integrity* guarantee.

C4: Stage C4 involves the communication between source MEC and target MEC. As these MECs are deployed in the 4G LTE cellular networks, we have assumed them to be non-malicious as defined in our threat model. It is the responsibility of the cellular network to keep the MEC platforms secure. Additionally, we use the common secret key for communication between MECs.

4) Forward Secrecy: We refresh the private keys (α and u) and public intermediate keys (g^{α} and g^{u}) keys every 24 hours to ensure the key freshness and forward secrecy. Also, the public intermediate keys work together to generate a specific common key for each pair of MEC and UE. Therefore, if a malicious UE gets hold of a specific common key, it will not be able to guess other common keys or decrypt all the communication that is happening between different MECs and UEs.

5) Replay Resistance: In replay attack, the UE can present an already used token. In order to maintain token freshness, we use time stamp with the token. When the source MEC A generates a token for the UE, it appends the time stamp with that token and the $vrfy$ works as a random salt. The timestamp is seen by the target MEC B when it extracts the token information from the encrypted message C_{UB}. If the UE replays a token, the MEC B easily detects it with the help of timestamp and $vrfy$. The UE is also not able to modify the values of $vrfy$ and as it can only be decrypted by another MEC. Hence, the replayed tokens are discarded at the target MEC.

6) MITM Resistance: In a MITM attack, a malicious UE can intercept the communications between the UE and MEC or between source and target MEC. However, it will not be able to access the information inside the captured packets as they are protected via encryption as explained in section IV.C. The attacker can get hold of the public intermediate keys of the UE and the MEC (g^{α} and g^{u}), but it will not be able to calculate the common key $g^{\alpha u}$ that is used to encrypt the communication between the MEC A and the UE. The attacker can insert malicious data into the intercepted packets too, but we provide signatures in the encrypted messages to check the integrity of the message and identity of the sender. The modification is easily detected at the MECs through $vrfy$ and such modified packets are discarded by the target MECs. The malicious UE can also sit between the two parties and pretend to be a legitimate party to both sides. This can easily be detected as we provide mutual authentication among the involved parties and the man-in-the-middle cannot appear to be legitimate because it does not possess the private keys (α and u) of legitimate users and without these keys, it cannot pose to be someone else during mutual authentication.

Our proposed TS3A and TC3A satisfy the security objectives as identified by Jing Zhang et al. [16]. We did not incorporate the identity privacy preserving and traceability objectives in this work as they are related to privacy which is out of scope of this work.

B. Possible Attacks

1) DoS Attacks: Malicious UEs can also cause denial of service (DoS) by sending multiple attach requests to the EPC or creating multiple sessions with the application servers deployed in the MEC. This can make service unavailable and it is a threat for the cellular networks. The DoS attack at the application layer level can be detected and mitigated in several ways such as providing the UEs with JavaScript computational challenge to calculate, deploying the web application firewall,

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

installing the intrusion detection and management system to detect abnormal behavior. Restricting the number of UEs connecting with an eNodeB can also help in avoiding this attack.

2) Malicious/compromised UE Attacks: In our threat model, we have assumed that a malicious UE may obtain the token from the genuine UE and present it to the target MEC. In such a case, malicious UE will be able to access a secured network. In order to avoid this, we use the field $vrfy$. When the MEC A generates a token, it appends the token with $vrfy$, which shows the issuer of the token and the intended audience, and encrypts them with the common secret key *K* so that it could only be decrypted by the target MEC. The MEC A then further encrypts the token with the $g^{\alpha u}$ which can only be decrypted by the legitimate UE and hence the malicious UE cannot extract the contents of this encrypted message. If we assume that a legitimate UE is compromised, and decrypts the message C_{AU}, it will still not be able to further extract the contents of $vrfy$ as it does not have the common secret key *K*. If the compromised UE modifies the contents of $vrfy$ without extracting it, the target MEC detects it via common secret key *K*. Hence, our solution provides security against malicious and compromised UE attacks.

VIII. Conclusion and Future Work

MEC technology is one of the key technologies in 5G networks as it supports applications with low latency requirements. In the near future, there will be many users accessing application servers in an MEC instead of routing through core cellular networks to the internet. Mobile users will face the issue of authentication and application mobility as they would not want to lose a session with the application server in the MEC, and would also not want to repeatedly provide authentication information. In order to address these issues, we proposed two approaches, TC3A and TS3A, which provide the seamless transfer of authentication information and application session of user from one MEC to another MEC while achieving low latency.

The results showed that TS3A and TC3A successfully re-authenticated the UEs with the target MEC at 49.76–59.72% reduced latency compared to simple login method. The TC3A and TS3A also eliminated the need of keeping multiple accounts for applications at different MECs and most importantly provided application service continuity, through state transfer during cross-system handover. Both of these proposed approaches proved to be useful for different types of applications. TS3A proved to be suitable for applications with strict latency requirements as it provided the application service continuity at 47.05–51.25% reduced latency compared to TC3A. TC3A, on the other hand, provided application service continuity while retaining the session state in its entirety and proved to be useful for applications that cannot afford even a slight state loss. The security analysis showed that TC3A and TS3A also fulfill the necessary security requirements.

This work addressed the authentication problem and application handover problem between the MECs that belong to one cellular service provider. TS3A and TC3A provide authentication token and state token in order to authenticate the users with the target MEC and resume the service for the users in target MEC from where they left the service in the source MEC while considering that both the MECs are deployed in the same cellular network. In future, we will work on the authentication and state transfer among MECs that belong to two different cellular networks and can either be in one or two different countries. We will consider a federated proxy solution for the MECs that belong to different cellular networks in order to enhance the coverage, capacity, and capability. Furthermore, we will also study the Quality of Experience (QoE) and Quality of Service (QoS) issues by considering the metrices like Mean Opinion Score, Peak-Signal-to Noise-Ratio (PSNR) and Mean Square Error (MSE), join time, buffering ratio, rate of buffering events, average bitrate and rendering quality.

References

[1] Asad Ali, Ying-Dar Lin, Chi-Yu Li, and Yuan-Cheng Lai, "Transparent 3rd-party Authentication with Application Mobility for 5G Mobile Edge Computing," In *European Conference on Networks and Communications (EuCNC)*, pages 219–224. IEEE, 2020.

[2] Mobile Edge Computing, "A Key Technology towards 5G," *ETSI White Paper* No, 11, 2015.

[3] 3GPP Technical Specification Group Services and System Aspects, "System Architecture for the 5G System: Stage 2 (release 15)," *3GPP Standard*, TS23.501 V0.4.0, 2017.

[4] Fabio Giust et al., "MEC Deployments in 4G and Evolution towards 5G," *ETSI White Paper*, 24:1–24, 2018.

[5] Tarik Taleb et al., "On Multi-access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys & Tutorials*, 19(3):1657–1681, 2017.

[6] Michael Jones, John Bradley, and Nat Sakimura, "JSON Web Token (JWT)," *Technical Report*, 2015.

[7] Dick Hardt and Yaron Goland, "Simple Web Token (SWT)," Version 0.9, 5:1427, 2009.

[8] Ronald Monzillo et al., "Web Services Security: SAML Token Profile," 2006.

[9] Sami Kekki et al., "MEC in 5G Networks," *ETSI white paper*, 28:1–28, 2018.

[10] GSMEC ETSI. 003,"Multi-access Edge Computing (MEC); Framework and Reference Architecture," *Technical report, ETSI*.[Online]. Available: https://www. etsi. org/deliver . . . , 2019.

[11] Yuting Li, Qingfeng Cheng, Ximeng Liu, and Xinghua Li, "A secure Anonymous Identity-based Scheme in New Authentication Architecture for Mobile Edge Computing," *IEEE Systems Journal*, 2020.

[12] Xiaoying Jia, Debiao He, Neeraj Kumar, and Kim-Kwang Raymond Choo, "A Provably Secure and Efficient Identity-based Anonymous Authentication Scheme for Mobile Edge Computing," *IEEE Systems Journal*, 14(1):560–571, 2019.

[13] Junho Lee, Dongwook Kim, Jinhyun Park, and Hyungweon Park, "A Multi-server Authentication Protocol Achieving Privacy Protection and Traceability for 5G Mobile Edge Computing," In *IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4. IEEE, 2021.

[14] Kaihong Han, Maode Ma, Xiaohong Li, Zhiyong Feng, and Jianye Hao, "An Efficient Handover Authentication Mechanism for 5G Wireless Network," In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–8. IEEE, 2019.

[15] Cong Wang, Yiying Zhang, Xi Chen, Kun Liang, and Zhiwei Wang, "SDN-based Handover Authentication Scheme for Mobile Edge Computing in Cyber-physical Systems," *IEEE Internet of Things Journal*, 6(5):8692–8701, 2019.

[16] Jing Zhang, Hong Zhong, Jie Cui, Miaomiao Tian, Yan Xu, and Lu Liu, "Edge Computing-based Privacy-preserving Authentication Framework and Protocol for 5G-enabled Vehicular Networks," *IEEE Transactions on Vehicular Technology*, 69(7):7940–7954, 2020.

[17] Qimei Cui, Zengbao Zhu, Wei Ni, Xiaofeng Tao, and Ping Zhang, "Edge-Intelligence-Empowered, Unified Authentication and Trust Evaluation for Heterogeneous Beyond 5G Systems," *IEEE Wireless Communications*, 28(2):78–85, 2021.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3296085

[18] Kuljeet Kaur, Sahil Garg, Georges Kaddoum, Mohsen Guizani, and Dushantha Nalin K Jayakody, "A Lightweight and Privacy-preserving Authentication Protocol for Mobile Edge Computing," In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[19] Shijie Zhang and Jong-Hyouk Lee, "A Group Signature and Authentication Scheme for Blockchain-based Mobile-edge Computing," *IEEE Internet of Things Journal*, 7(5):4557–4565, 2019.

[20] Eduard Grasa, Miguel Ponce de Leon, Sven van der Meer, Diego Lopez, and Miguel Tarzan, "Open Multi-access Edge Computing and Distributed Mobility Management with RINA," In *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–2. IEEE, 2017.

[21] Ping Zhang, Mimoza Durresi, and Arjan Durresi, "Multi-access Edge Computing Aided Mobility for Privacy Protection in Internet of Things," *Computing*, 101(7):729–742, 2019.

[22] Tejas Subramanya, Giovanni Baggio, and Roberto Riggio, "lightMEC: A Vendor-Agnostic Platform for Multi-access Edge Computing," In *14th International Conference on Network and Service Management (CNSM)*, pages 198–204. IEEE, 2018.

[23] Erik Richard Stauffer and Jibing Wang, "Fifth Generation New Radio Edge Computing Mobility Management," *US Patent*, 10,499,304, December 3, 2019.

[24] Osamah Ibrahiem, Li-Chun Wang, Shahzoob Bilal Chundrigar, and Kuei-Li Huang, "Enabling Mobile Service Continuity across Orchestrated Edge Networks," *IEEE Transactions on Network Science and Engineering*, 2019.

[25] Chi-Yu Li, Ying-Dar Lin, Yuan-Cheng Lai, Hsu-Tung Chien, Yu-Sheng Huang, Po-Hao Huang, and Hsueh-Yang Liu, "Transparent AAA Security Design for Low-latency MEC-integrated Cellular Networks," *IEEE Transactions on Vehicular Technology*, 69(3):3231–3243, 2020.

[26] Vipindev Adat Vasudevan, Muhammad Tayyab, George P Koudouridis, Xavier Gelabert, and Ilias Politis, "An Integrated Approach for Energy Efficient Handover and Key Distribution Protocol for Secure NC-enabled Small Cells," *Computer Networks*, 108806, 2022.

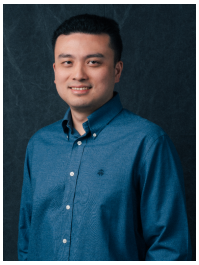
[27] Ziyu Xiao and Harry Perros, "Response Time of the S1 and X2 Handover Procedures Between (H)eNBs in a Virtualized Environment."

[28] Xinyu Lei, Guan-Hua Tu, Chi-Yu Li, and Tian Xie, "SecWIR: Securing Smart Home IoT Communications via Wi-Fi

[29] Maode Ma, "Security Investigation in 4G LTE Wireless Networks," *School of Elecrical and Electronic Engineering, Nanyang Technological University, Singapore*, 2012.

Asad Ali received his PhD degree from National Yang Ming Chiao Tung University, Taiwan in Electrical Engineering and Computer Sciences. He did his Master degree in Electrical Engineering from National University of Science & Technology, Pakistan. His research interests are cyber seciroty, cyber threat intelligence, network security, 5G, multi access edge computing, wireless communications, network design and optimization.

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 and standardized into IEEE 802.11s, IEEE 802.15.5, IEEE 802.16j, and 3GPP LTE-Advanced. 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.



Chi-Yu Li is an associate professor in the Department of Computer Science at NYCU and leads the NEtworking and Mobile Systems (NEMS) laboratory. He received his PhD degree in Computer Science from UCLA. He received his MS degree in Computer Science from NYCU in 2006, and two Bachelor's degrees in Computer Science and Management Science from NYCU in 2004. His research interests include wireless networking, mobile networks and systems, and network security.

Yuan-Cheng Lai received his PhD degree from the Department of Computer and Information Science from National Yang Ming Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at National Taiwan University of Science and Technology in August 2001 and has been a professor since February 2008. His research interests include performance analysis, network security, 5G mobile networks and Internet of Things.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on September 04,2023 at 15:11:40 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.