Service Migration Authentication Protocol for MEC

Conference Paper · December 2022

DOI: 10.1109/GLOBECOM48099.2022.10001261

Pasika S Ranaweera
University College Dublin
29 PUBLICATIONS **430** CITATIONS

Awaneesh kumar Yadav
Indian Institute of Technology Roorkee
11 PUBLICATIONS **9** CITATIONS

Madhusanka Liyanage
University College Dublin
260 PUBLICATIONS **6,194** CITATIONS

Anca D Jurcut
University College Dublin
100 PUBLICATIONS **1,114** CITATIONS

Service Migration Authentication Protocol for MEC

Pasika Ranaweera*, Awaneesh Kumar Yadav†, Madhusanka Liyanage‡, Anca Delia Jurcut§
*‡§School of Computer Science, University College Dublin, Ireland
†Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India
‡Centre for Wireless Communications, University of Oulu, Finland
Email: *pasika.ranaweera@ucdconnect.ie,†akumaryadav@cs.iitr.ac.in,‡madhusanka@ucd.ie, §anca.jurcut@ucd.ie

***Abstract*—Multi-Access Edge Computing (MEC) is a novel edge computing paradigm that enhances the access level capacity of mobile networks by shifting the serviceable Data center infrastructure proximate to the end devices. With this proximate placement and service provisioning, migration of a service from one edge enabled gNodeB (gNB) to another is intrinsic to maintain the service continuity. Since such services are migrated through the channel shared between the gNBs, proper security measures should be inhibited by the communication protocol to prevent any unauthorized interception. Further, each gNB should ensure the legitimacy of the migrating gNBs to avoid any impersonation attempts. As this is an area that lacks focus in current research trends, this paper introduces MEC Service Migration Authentication Protocol (MEC-SMAP), a protocol that take place prior to the migration initiation, and specifically defined for MEC. The proposed protocol ensure the secure transfer of session key generation parameters to form a secure channel while ensuring perfect forward secrecy. It introduces an identity verification mechanism through a trusted third party service. We have validated the proposed protocol through formal analysis using GNY logic and Scyther tool. Further, a prototype virtualized MEC environment was created to evaluate its feasibility and the impact of the employed security mechanisms.**

***Index Terms*—Authentication, Edge Computing, Identity Verification, MEC, Service Migration, Security, Verification**

I. Introduction

The 5G mobile technology is the seminal advancement explored by the mobile network operators to reach beyond the constrictions of the prevailing network architecture. The existing cloud native service provisioning infrastructure however, is not suited to cater such requirements. Thus, to realize the requisites demanded by the 5G related directives, edge computing approaches are acting as the raison d'être to overcome such constrictions through a proximate storage and processing environment. Among the edge computing paradigms, Multi-Access Edge Computing (MEC) is a leading concept, that is proposed by the European Telecommunications Standards Institute (ETSI). Emergence of edge computing paradigms have introduced the concept of service migration to cater the heterogeneous IoT devices a ubiquitous connectivity over the mobile network. In a situation where the User Equipment (UE) is traversing beyond the range of the currently serving MEC gNB, the service instance should be migrated to a MEC gNB that is proximate to the UE roamed location. Once migrated and configured to the roamed MEC infrastructure, offered service will continue without disruption. The Quality of Service (QoS) and Quality of Experience (QoE) aspects of the offered service is entirely dependent on the seamless operation of the migration process. The latency caused in the migration process will result in disruption of the service to the UE; there by impacting both QoS and QoE factors negatively. Thus, service migration within edge computing platforms is a weaker aspect of MEC that forecasts inevitable issues.

The process of migration could induce unprecedented vulnerabilities and flaws in an MEC environment. As the channel involved in migration is carrying actual executable content, which are hibernated and compressed of an executing service instance at the MEC environment, perpetrators might tend to extract and replace, or inject malicious content of their fabrication through intercepting. Once migrated, there won't be any security validation conducted to verify the content, nor would the service will have time at their end to spare due to the time intensiveness of the applications of 5G. Thus, it is important to secure the communication channels that are laid across the MEC enables edge gNBs. Even if the migration channel is secure from penetration, masquerading attempts are imminent as 5G is supporting wide range of services that would eventually launch myriads of edge gNBs with limited coverage. Tracking the legitimacy of the MEC service providers is an arduous task and it is obvious that the trust domain of a MEC system level will be quite limited. Thus, a proper trust accountability function is required to further secure the migration channels.

In spite there are research conducted on service migration in edge computing paradigms, they are mostly targeted on migrating models formed to optimize the energy consumption using either Markov or Lyapunov optimization techniques [1]. Security of the service migration channel of MEC, specifically for an Edge-to-Edge (E2E) scenario is never been performed. Zhang et al. in [2] proposes a handover authentication protocol for 5G HetNets, where it is called as RUSH. Though the context is involving a gNB and its connection to UE, the requirement to secure the E2E channel was not one of their objectives. A blockchain based secure edge service migration framework called Falcon was proposed by Zhang et al. in [3]. Falcon enables containers acting as mobile agents to perform service migration securely, where the corresponding transactions are secured via an immutable alliance chain. However, it is doubtful on how the transferable content is secured when migrating, where a hardcore security solution in the protocol level is lacking. Karthick et al. proposed a

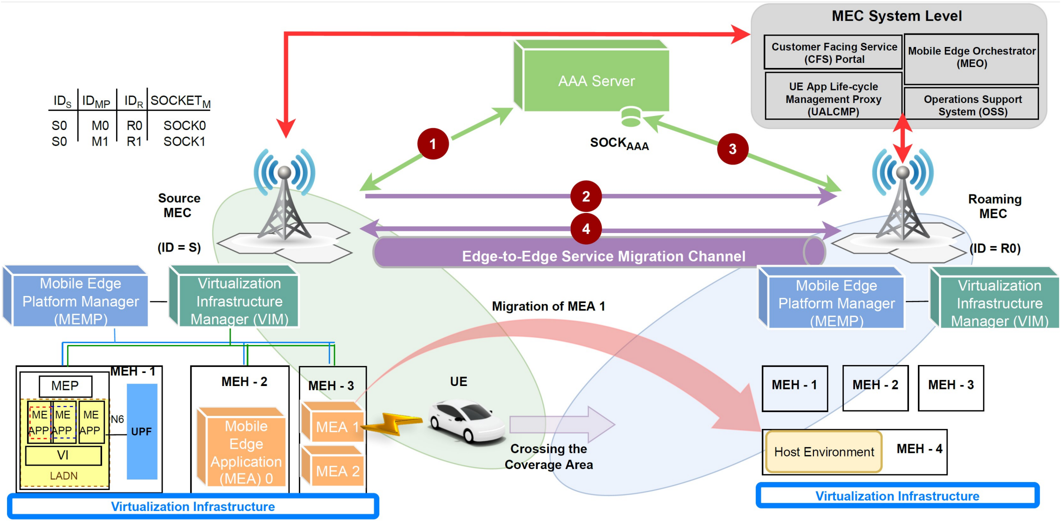

MEC System Level
Customer Facing Service (CFS) Portal
Mobile Edge Orchestrator (MEO)
UE App Life-cycle Management Proxy (UALCMP)
Operations Support System (OSS)
AAA Server
SOCK$_{AAA}$
1
2
3
4
ID$_S$ | ID$_{MP}$ | ID$_R$ | SOCKET$_M$
S0 | M0 | R0 | SOCK0
S0 | M1 | R1 | SOCK1
Source MEC
(ID = S)
Roaming MEC
(ID = R0)
Edge-to-Edge Service Migration Channel
Mobile Edge Platform Manager (MEMP)
Virtualization Infrastructure Manager (VIM)
Migration of MEA 1
MEH - 1
MEP
ME APP
ME APP
ME APP
VI
LADN
N6
UPF
MEH - 2
Mobile Edge Application (MEA) 0
MEH - 3
MEA 1
MEA 2
UE
Crossing the Coverage Area
Virtualization Infrastructure
Mobile Edge Platform Manager (MEMP)
Virtualization Infrastructure Manager (VIM)
MEH - 1
MEH - 2
MEH - 3
Host Environment
MEH - 4
Virtualization Infrastructure

Fig. 1: The Proposed MEC Service Migration Authentication Model

resource allocation security protocol for service migrations in cloud computing scenarios in [4]. Despite its validity in the cloud context and for Vehicular applications, the protocol lacks the Perfect Forward Secrecy (PFS), detection of Denial of Service (DoS) capability, and Signature reuse threat to be suited for an E2E context. A jamming strategy was introduced by Cui et al. in [5] utilizing Fountain codes. Eventhough the proposed cooperative jamming strategy proved to be valuable in deceiving the adversary as in a honeypot scenario, a resourceful perpetrator would be able reverse engineer the jamming techniques. Thus, securing the migration content at the protocol level is vital.

With the best of our knowledge, a research has not been conducted on securing the E2E migration channels, specifically for MEC. Thus, in this paper we are proposing MEC Service Migration Authentication Protocol (MEC-SMAP), which facilitate an identity verification mechanism that suites any edge computing paradigm. MEC-SMAP includes the security features to cope up with any interposing, Replaying, impersonating, and Denial of Service (DoS) threats. A formal verification by means of GNY logic and the Scyther tool was conducted for validating MEC-SMAP. Further, the proposed protocol was implemented in a prototype MEC environment to evaluate its feasibility and the effectiveness of the employed security measures. The Section II introduces the proposed migration, and threat models we consider, while Section III describes the protocol specification. The formal validation conducted is presented in Section IV, and Section V specify the contrived prototype testing environment, ultimately concluding the paper in Section VI.

II. Proposed MEC Migration and Threat Models

This section specifies the proposing authentication model in the context of the MEC architecture, and the threat models we have considered in the formulation of the protocol.

A. MEC Architectural Migration Model

The Fig. 1 depicts a typical scenario where a migration of services are required, typically when a UE is crossing into a domain of another MEC service area. In this scenario, Mobile Edge Application (MEA) currently serving the UE should be either partially or completely transferred to the allocated host environment at the Mobile Edge Host (MEH) 4 under Roaming MEC. MEHs are the primary operational elements of the MEC environment where their resource management is conducted by the Virtualization Infrastructure Manager (VIM), and Mobile Edge Platform Manager (MEPM) acts as the edge level orchestrator [6]. In an implementation perspective, these MEA can be envisaged to launched as virtualized containers while MEHs can be operated as Virtual Machines (VMs) that host such a containerized environment [7]. The standardized MEC architecture suggest that MEC edge levels are governed by the MEC system level entities, mainly via Mobile Edge Orchestrator (MEO) and Operations Support System (OSS). That makes the MEC edge environment completely autonomous, and restricts the administrative or user access to the system, which in turn would slow down the overall responsiveness.

In the proposed authentication model, the source MEC (i.e. MEC_{Source}) is seeking out the possible Roaming MECs (i.e. $MEC_{Roaming}$s) suitable and secure to host the migrating MEA. As the futuristic networks support many macro/ micro-cell based edge base stations, it is important to validate the legitimacy of the $MEC_{Roaming}$. Thus, we consider the mobile network as untrusted during our design. Due to this untrust, a trusted third party service can be employed to perform the trust verification task. Therefore, we are proposing an Authentication, Authorization, and Accountability (AAA) server (see in Fig. 1) to outsource the identity verification tasks, as it would unburden the MEC system level from such abundant migration attempts. In fact, this AAA server is performing Migration Authentication as a Service (MAaaS). Since AAA service represents an uninterruptible service, we have embedded a Denial of Service (DoS) detection puzzle [8] in this phase, where AAA server is providing a challenge of d_{dos} to the MEC_{Source}, while the puzzle of $H[ID_S||ID_AAA||n_1||n_2||X] = 0_1 0_2 0_3 ... 0_{d_{DoS}} Y$ should be solved to determine X. Hence, X will be sent to AAA to ensure of the attempt is not a DoS threat. According to the proposed model, prior to contacting the respective $MEC_{Roaming}$ station, MEC_{Source} will reach out to the AAA server in order to register its migration. We assume that all the legitimate MEC service hosts are registered under the AAA service. This contact is concluded with MEC_{Source} attaining the respective $SOCK_{AAA}$, a unique socket or an API link created for this migration; and ID_M migration IDs that are valid only for this migration session. In the next authentication phase MEC_{Source} contacts $MEC_{Roaming}$ and conveys the gathered credentials, while $MEC_{Roaming}$ will employ the credentials to contact the AAA service. AAA server will generate a unique value $CODE_M$ along with session key parameters and send them to both MEC_{Source} and $MEC_{Roaming}$. Once the conveyed codes are verified and identity verification/ mutual authentication is established, MEC_{Source} transfers the $SPEC_{MEA}$ (i.e. specifications of the MEA), and REQ_{MEA} (i.e. resource requirements to host the MEA) to $MEC_{Roaming}$. If the $MEC_{Roaming}$ possess the specified requirements, the Launching Feasibility (LF) will be conveyed, and the respective Security Profile (SP) will be selected via the communication secured with the migration key K_{S-R}. The SP stands for the overall security credential template that is in agreement with the two MEC environments for migration. As migrations can be cumbersome content, SP allows the MEC environments to decide which suite of credentials to be utilized for the secure transfer. Such signalling message transfers are secured with the K_{S-R}.

B. Threat Model

To verify the resilience, we adopt the Delev-Yao (DY) [9] and the Canetti and Krawczyk's adversary model (CK-adversary model) [10] threat model. In this threat model, the adversary can alter, capture, and insert into the public channel communication. Consequently, we are targeting Man-in-the-Middle (MitM), Replay, Relay, and Malicious injection threats. Aside from that, an adversary could obtain the communicating entities' private keys or secrets generated during the session, which compromise the PFS.

III. Proposed MEC-SMAP Protocol

This section presents the registration phase and MEC service migration authentication phase of MEC-SMAP.

A. Registration Phase

In this phase, MEC_{Source} requests the necessary unique/secret credentials from the AAA server.

Step-1 ($MEC_{Source} \rightarrow AAA$): At the starting, MEC_{Source} selects the timestamp T_1 and random number m_1 (i.e., $m_1 \in Z_n$) in order to send the $E_{PuK_{AAA}}(ID_S, P_1, T_1)$ and $HMAC_1 = H(ID_S, ID_{AAA}, T_1)$ to the AAA server.

Step-2 ($AAA \rightarrow MEC_{Source}$): When AAA server receives the message then it decrypts the message to obtain the secrets (ID_S, T_1, P_1) and computes $HMAC_1^* = H(ID_S, ID_{AAA}, T_1)$. Afterwards, it verifies the freshness condition $(T_r - T_s \leq \triangle T)$ (i.e., $(T_2 - T_1 \leq \triangle T)$), if it matches then it searches the ID_S in the database, if it matches then it aborts otherwise it compares the $\{HMAC_1 == HMAC_1^*\}$ in order to verify that message is send from the legitimate MEC_{Source}. AAA server selects the random number m_2 (i.e., $m_2 \in Z_n$) and time stamp in order to compute the response of the received message $E_{PuK_S}(d_{dos}, n_1, P_2, T_2)$, $HMAC_2 = H(d_{dos}, n_1, ID_S, T_2)$ and forwards this to the MEC_{Source}.

Step-3 ($MEC_{Source} \rightarrow AAA$): After receiving the message from AAA server, MEC_{Source} decrypts the message to obtain the secrets (d_{dos}, n_1, P_2, T_2) in order to compute $\{HMAC_2^* = H(d_{dos}, n_1, ID_S, T_2)\}$. Afterwords, MEC_{Source} verifies the freshness condition, if it matches then it compares $\{HMAC_2 == HMAC_2^*\}$, if it matches then it believes that it is the authentic response to the previously send message. Now MEC_{Source} computes the $E_{PuK_{AAA}}(n_2, X, H[n_1, T_3], T_3)$ and sends this to the AAA server.

Step-4 ($AAA \rightarrow MEC_{Source}$): AAA server decrypts the message in order to compute $H^* = H(n_1, T_3)$. Afterwords it verifies the freshness condition along with the received secrets of the message, if it matches then believes the MEC_{Source} is legitimate then compute $K_1 = m_1.m_2.M$, $E_{PuK_S}((Enc_{K_1}(SOCK_{AAA_M}, ID_M - ARR(R_1, R_2 ... R_N)$ $, r_1')), H(n_2 \parallel T_4), T_4)$ and forwards this to the MEC_{Source}.

Step-5: When MEC_{Source} receives the message then it decrypts the message to obtain $(SOCK_{AAA_M}, ID_M - ARR(R_1, R_2 ... R_N), r_1')$. After getting the credentials, it verifies the freshness condition and the received credentials, if they matches then it saves the credentials for further communication and believes that AAA server is authentic.

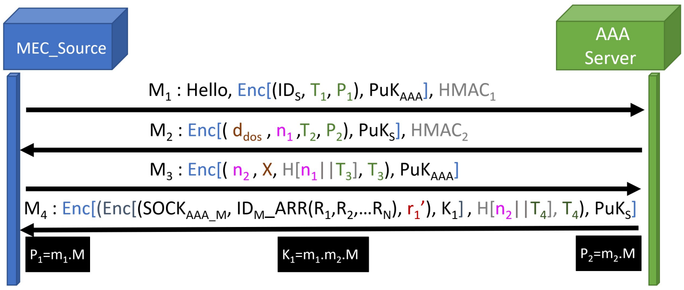

MEC_Source
AAA Server
M_1 : Hello, Enc[(ID_S, T_1, P_1), PuK_{AAA}], $HMAC_1$
M_2 : Enc[(d_{dos} , n_1 ,T_2, P_2), PuK_S], $HMAC_2$
M_3 : Enc[(n_2 , X, H[n_1||T_3], T_3), PuK_{AAA}]
M_4 : Enc[(Enc[($SOCK_{AAA\_M}$, ID_M_ARR(R_1,R_2,...R_N), r_1'), K_1] , H[n_2||T_4], T_4), PuK_S]
P_1=m_1.M
K_1=m_1.m_2.M
P_2=m_2.M

Fig. 2: Registration Phase of the MEC-SMAP that takes place between MEC_{Source} and the AAA server

B. Authentication Phase

In this phase, MEC_{Source} and $MEC_{Roaming}$ verifies their legitimacy with the help of AAA through the $SOCK_{AAA-M}$. The proposed protocol combines RSA and ECC, where RSA facilitating secrecy over the public channel and ECC being used to provide PFS.

Step-1 ($MEC_{Source} \rightarrow MEC_{Roaming}$): MEC_{Source} selects the timestamp T_1 and random number m_3 (i.e.,

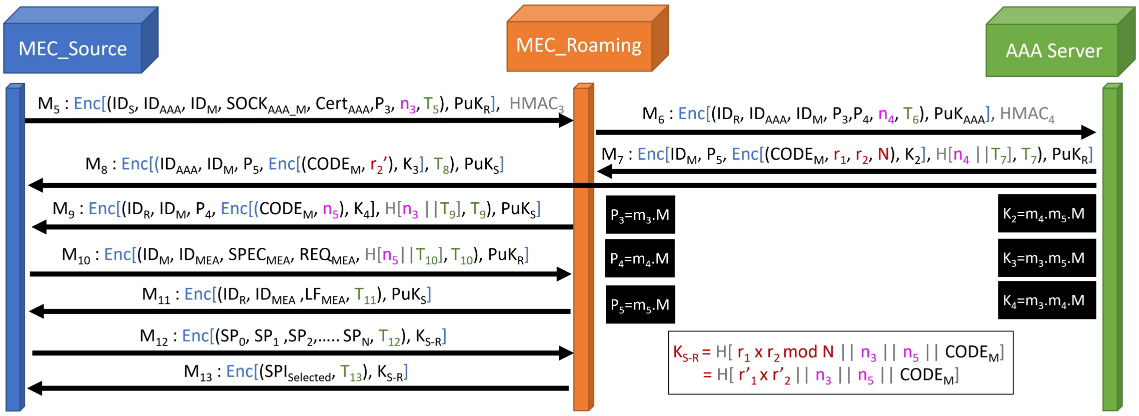

MEC_Source
MEC_Roaming
AAA Server
M_5 : Enc[(ID_S, ID_{AAA}, ID_M, $SOCK_{AAA\_M}$, $Cert_{AAA}$,P_3, n_3, T_5), PuK_R], $HMAC_3$
M_6 : Enc[(ID_R, ID_{AAA}, ID_M, P_3,P_4, n_4, T_6), PuK_{AAA}], $HMAC_4$
M_7 : Enc[ID_M, P_5, Enc[($CODE_M$, r_1, r_2, N), K_2], H[n_4 | |T_7], T_7), PuK_R]
M_8 : Enc[(ID_{AAA}, ID_M, P_5, Enc[($CODE_M$, r_2'), K_3], T_8), PuK_S]
M_9 : Enc[(ID_R, ID_M, P_4, Enc[($CODE_M$, n_5), K_4], H[n_3 | |T_9], T_9), PuK_S]
M_{10} : Enc[(ID_M, ID_{MEA}, $SPEC_{MEA}$, REQ_{MEA}, H[n_5| |T_{10}], T_{10}), PuK_R]
M_{11} : Enc[(ID_R, ID_{MEA} ,LF_{MEA}, T_{11}), PuK_S]
M_{12} : Enc[(SP_0, SP_1 ,SP_2,..... SP_N, T_{12}), K_{S-R}]
M_{13} : Enc[($SPI_{Selected}$, T_{13}), K_{S-R}]
P_3=m_3.M
P_4=m_4.M
P_5=m_5.M
K_2=m_4.m_5.M
K_3=m_3.m_5.M
K_4=m_3.m_4.M
K_{S-R} = H[r_1 x r_2 mod N | | n_3 | | n_5 | | $CODE_M$]
= H[r'_1 x r'_2 | | n_3 | | n_5 | | $CODE_M$]

Fig. 3: Authentication Phase of the MEC-SMAP that takes place between MEC_{Source}, $MEC_{Roaming}$, and the AAA

$m_3 \in Z_n$) and sends the request by forwarding the $E_{PuK_R}(ID_S, ID_{AAA}, ID_M, SOCK_{AAA_M}, Cert_{AAA}, P_3$ $n_3, T_5)$, $HMAC_3 = H(ID_S \parallel ID_{AAA} \parallel ID_{M_{RO}} \parallel SOCK_{AAA_M} \parallel Cert_{AAA} \parallel n_3 \parallel ID_R \parallel T_5)$ to the $MEC_{Roaming}$.

Step-2 ($MEC_{Roaming} \rightarrow AAA$): When $MEC_{Roaming}$ receives the message from MEC_{Source}, first it decrypts the message to obtain the secrets $\{ID_S, ID_{AAA}, ID_M, SOCK_{AAA_M}, Cert_{AAA}, P_3, n_3, T_5\}$ in order to verify the freshness condition, if it matches then it computes $HMAC_3^*$ and compares $\{HMAC_3 == HMAC_3^*\}$. If it matches then selects the random number m_4 (i.e., $m_4 \in Z_n$) to compute the $E_{PuK_{AAA}}(ID_R, ID_{AAA}, ID_M, P_4, P_3, n_4, T_6)$ and forwards this to the AAA server for verification.

Step-3 ($AAA \rightarrow MEC_{Roaming}$): After receiving the message from the $MEC_{Roaming}$, AAA server decrypts this message to obtain the credentials $\{ID_R, ID_{AAA}, ID_M, n_4, T_6, P_3, P_4\}$. After that it compute the two messages, first for the $MEC_{Roaming}$ (i.e., $E_{PuK_R}(ID_M, P_5, (Enc_{K_2}(CODE_M, r_1, r_2, N)), H(n_4 \parallel T_7), T_7))$ and later for MEC_{Source} (i.e., $E_{PuK_S}(ID_{AAA}, ID_M, P_5, (Enc_{K_3}(CODE_M, r'_2)), T_8))$. After computing these messages, it forwards them to the MEC_{Source} and $MEC_{Roaming}$.

Step-4 ($MEC_{Roaming} \rightarrow MEC_{Source}$): $MEC_{Roaming}$ receives the message and decrypts it to obtain the credentials $\{ID_M, CODE_M, r_1, r_2, N, P_5, H(n_4 \parallel T_7), T_7\}$, it then compute $H(n_4 \parallel T_7)$ and compare it with the received if matches then compute the $E_{PuK_S}(ID_R, ID_M, P_4, Enc_{K_4}(CODE_M, n_5), H[n_3$ $, \parallel T_9], T_9)$ and forwards to the MEC_{Source}.

Step-5 ($MEC_{Source} \rightarrow MEC_{Roaming}$): MEC_{Source} decrypts both the message to obtain the credentials $\{ID_{AAA}, ID_M, P_5, CODE_M, r'_2, T_8\}$, $\{ID_R, ID_M, P_4, CODE_M, n_5, H(n_3 \parallel T_9), T_9\}$. Afterwords, it compares the $CODE_{M_{Roaming}} == CODE_{M_{AAA}}$, if matches then it believes the $MEC_{Roaming}$ and AAA are authentic then computes a message $E_{PuK_R}(ID_M, ID_{MEA}, SPEC_{MEA}, REQ_{MEA}, H[n_5$ $, \parallel T_{10}], T_{10})$ and forwards it to the $MEC_{Roaming}$.

Step-6 ($MEC_{Roaming} \rightarrow MEC_{Source}$): When $MEC_{Roaming}$ receives the message from the MEC_{Source}, it then decrypts the message and verifies the freshness of the message. If freshness condition matches then it compares the credentials based on that it computes $E_{PuK_S}(ID_R, ID_{MEA}, LF_{MEA}, T_{11})$ and forwards it to the MEC_{Source}.

Step-7 ($MEC_{Source} \rightarrow MEC_{Roaming}$): When MEC_{Source} receives the message then decrypts this and verifies the freshness. Afterwords, it compute the $E_{K_{S-R}}(SP_0, SP_1, SP_2.....SP_N, T_{12})$ and forwards this to the $MEC_{Roaming}$.

Step-8 ($MEC_{Roaming} \rightarrow MEC_{Source}$): $MEC_{Roaming}$ dercypts the received message and verifies the freshness then based on the received SP_i, it selects $SPI_{Selected}$ in order to compute $E_{K_{S-R}}(SP_{Selected}, T_{13})$, forwards to the MEC_{Source}.

IV. Formal Verification of MEC-SMAP and Efficacy

This section shows how the proposed protocol was formalized using the GNY logic [11] and Scyther tool [12]. The proposed protocol is formal validated to ensure that it fits all security requirements and generates the secrets.

A. Security Validation of MEC-SMAP using GNY Logic

The proposed protocol's security is verified utilizing the GNY logic between the $MEC_{Source}(S)$, $MEC_{Roaming}(R)$, and the $AAA(A)$. We used the notations given in [11] in conducting this validation, while following the logical postulates of Being told rules (1-3), Possession rules (1-3), and the Freshness rule.

1) Initial assumptions for the protocol

$H_1 : S \ni (PrK_S, K_3, K_4), H_2 : S \ni (ID_{AAA}, ID_M)$
$H_3 : S\#(T_8, T_9, T_{11}, T_{13}), H_4 : R\#(T_5, T_7, T_{10}, T_{12})$
$H_5 : R \ni PrK_R, H_6 : R \ni (n_4, K_2)$
$H_7 : R \ni (ID_R, ID_M, ID_{AAA}), H_9 : A \ni PrK_{AAA}$
$H_{10} : A\#(T_6), H_{11} : A \ni (ID_{AAA}, ID_M, ID_R)$

2) *Security goals of the proposed protocol:*

$R \mid\equiv \#(ID_S, ID_{AAA}, ID_M, SOCK_{AAA}, Cert_{AAA}, P_3, n_3)$
$R \ni (SP_0, SP_1, SP_2....SP_N, T_{12})$
$S \ni (SPI_{Selected}, T_{13})$
$R \ni (SP_0, SP_1, SP_2....SP_N, T_{12})$

3) *Idealized form of the proposed protocol:*

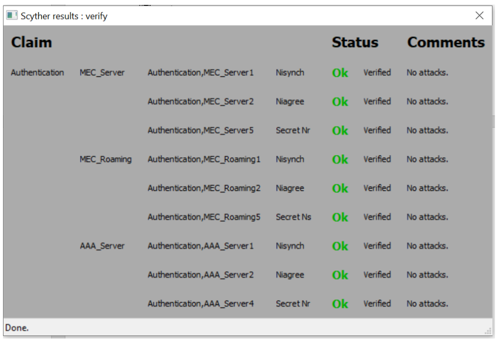
M_1:$R\lhd : *(*ID_S, *ID_{AAA}, *ID_M, *SOCK_{AAA}, *Cert_{AAA}, *P_3, *n_3, *T_5)_{PuK_R}$
M_2: $A\lhd$: $*(*ID_R, *ID_{AAA}, *ID_M, *n_4, *P_3, *P_4 * T_6)_{PuK_{AAA}}$
M_3:$R\lhd$:$*(ID_M, *P_5, *(CODE_M, *r_1, *r_2, N)_{K_2}, *H(n_4, T_7), *T_7)_{PuK_R}$,
M_4:$S\lhd$:$*(*ID_{AAA}, *ID_M, *P_5, *(CODE_M, *r_2')_{K_3}, *T_8)_{PuK_S}$,
M_5: $S\lhd$: $*(*ID_R, *ID_M, *P_4, (CODE_M, n_5)_{K_4}, *H[n_3, T_9], T_9)_PuK_S$
M_6:$R\lhd$: $(ID_M, ID_{MEA}, SPEC_{MEA}, REQ_{MEA}, *H[*n_5, T_{10}], T_{10})_{PuK_R}$
M_7:$S\lhd$: $(ID_R, ID_{MEA}, LF_{MEA}, T_{11})_{PuK_S}$
M_8:$R\lhd$: $(SP_0, SP_1, SP_2....SP_N, T_{12})_{K_{S-R}}$
M_9:$S\lhd$: $(SPI_{Selected}, T_{13})_{K_{S-R}}$

4) *Proof and derivation of security goals:*

1: By applying the BTR_1, BTR_3 and PR_1 rule on M_1 based on H_5, $S_1 : R \ni (ID_S, ID_{AAA}, ID_M, SOCK_{AAA}, Cert_{AAA}, P_3, n_3, T_5)$
2: We apply the BTR_2, PR_3 rule based on S_1 and H_4, $S_2 : R \ni H(ID_S, ID_{AAA}, ID_M, SOCK_{AAA}, Cert_{AAA}, P_3, n_3, T_5)$
3: Applying the FR rule on S_1, S_2 based on H_4, $S_3 : R \mid\equiv \#(ID_S, ID_{AAA}, ID_M, SOCK_{AAA}, Cert_{AAA}, P_3, n_3)$
4: By applying the BTR_1, BTR_3 and PR_1 rule on M_2 based on H_9, $S_4 : A \ni (ID_R, ID_{AAA}, ID_M, n_4, P_3, P_4, T_6)$
5: We apply the BTR_2, PR_3 rule based on S_4 and H_{11}, $S_5 : A \ni H(ID_R, ID_{AAA}, ID_M, n_4, P_3, P_4, T_6)$
6: Applying the FR rule on S_{10}, S_6 : $A \mid\equiv \#(ID_R, ID_{AAA}, ID_M, n_4, P_3, P_4)$
7: By applying the BTR_1 and BTR_3 and PR_1 rule on M_3 based on H_5, S_7 : $R \ni (ID_M, P_5, (CODE_M, r_1, r_2, N)_{K_{R-AAA}}, H(n_4, T_7), T_7))$
8: We apply the PR_3 and BTR_4 rule on S_7 based on H_6, $S_8 : R \ni (CODE_M, r_1, r_2, N)$
9: Applying the FR rule based on S_6 and H_4, S_9 : $R \mid\equiv \#((ID_M, P_5, (CODE_M, r_1, r_2, N)_{K_{R-AAA}}, H(n_4, T_7))$
10: By applying the BTR_1, BTR_3 and PR_1 rule on M_4 based on H_1, $S_{10} : S \ni (ID_{AAA}, ID_M, P_5, (CODE_M, r_2')_{K_3}, T_8)$
11: We apply the PR_3 and BTR_4 rule on S_{10} based on H_1, $S_{11} : S \ni (CODE_M, r_2')$
12: Applying the FR rule based on S_{10} and H_3, $S_{12} : S \mid\equiv \#((ID_{AAA}, ID_M, P_5, (CODE_M, r_2')_{K_3})$
13: By applying the BTR_1, BTR_3 and PR_1 rule on M_5 based on H_1, $S_{13} : S \ni (ID_R, ID_M, P_4, (CODE_M, n_5)_{K_4}, H[n_3, T_9], T_9)$
14: We apply the PR_3 and BTR_4 rule on S_{13} based on H_1, $S_{14} : S \ni (CODE_M, n_5)$
15: Applying the FR rule based on S_{13} and H_3, $S_{15} : S \mid\equiv \#(ID_R, ID_M, P_4, (CODE_M, n_5)_{K_4}, H[n_3, T_9])$
16: By applying the BTR_1, BTR_3 and PR_1 rule on M_6 based on H_5, S_{16} : $R \ni (ID_M, ID_{MEA}, SPEC_{MEA}, REQ_{MEA}, H[n_5, T_{10}], T_{10})$
17: Applying the FR rule based on S_{16} and H_4, S_{17} : $R \mid\equiv \#(ID_M, ID_{MEA}, SPEC_{MEA}, REQ_{MEA}, H[n_5, T_{10}])$
18: By applying the BTR_1, BTR_3 and PR_1 rule on M_7 based on H_1, $S_{18} : S \ni (ID_R, ID_{MEA}, LF_{MEA}, T_{11})$
19: Applying the FR rule based on S_{18} and H_3, $S_{19} : S \mid\equiv \#(ID_R, ID_{MEA}, LF_{MEA})$
20: By applying the BTR_1, BTR_3 and PR_1 rule on M_8 based on H_5, $S_{20} : R \ni (SP_0, SP_1, SP_2....SP_N, T_{12})$
21: Applying the FR rule based on S_{20} and H_4, $S_{21} : R \mid\equiv \#((SP_0, SP_1, SP_2....SP_N))$
22: By applying the BTR_1, BTR_3 and PR_1 rule on M_9 based on H_1, $S_{22} : S \ni (SPI_{Selected}, T_{13})$
23: Applying the FR rule based on S_{16} and H_4, $S_{23} : S \mid\equiv \#(SPI_{Selected})$

B. *Formal Analysis using Scyther Tool*

We use the Scyther tool [12] to examine the correctness of the proposed protocol. Scyther tool is used to examine the security properties of the proposed protocol using the specified claims. It uses the Security Protocol Description Language (.spdl) to model the security protocols. There are four types of security claims such as Alive (i.e., assures that the communicating parties carry out all events), Weakagree (i.e., guarantees that the protocol is not vulnerable to impersonation attacks), Nisynch (i.e., guarantees that the sender sends all messages and that the recipient receives them), and Secret (i.e., secrets are unknown to the attacker) defined in the scyther tool. The outcome of the scyther tool as indicated in Fig. 4 clearly shows that proposed protocol passes all the security claims which means there is no attack. Therefore, we can infer from the outcome that proposed protocol is secure against all the identified attacks.



Scyther results : verify

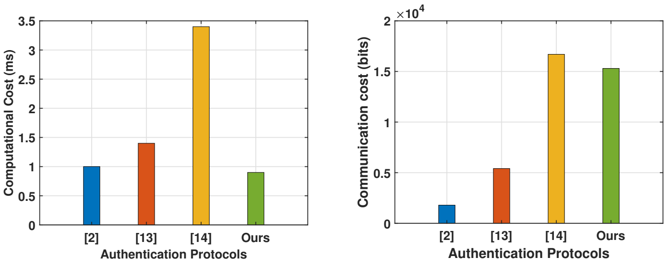
Claim				Status		Comments
Authentication	MEC_Server	Authentication,MEC_Server1	Nisynch	Ok	Verified	No attacks.
		Authentication,MEC_Server2	Niagree	Ok	Verified	No attacks.
		Authentication,MEC_Server5	Secret Nr	Ok	Verified	No attacks.
	MEC_Roaming	Authentication,MEC_Roaming1	Nisynch	Ok	Verified	No attacks.
		Authentication,MEC_Roaming2	Niagree	Ok	Verified	No attacks.
		Authentication,MEC_Roaming5	Secret Ns	Ok	Verified	No attacks.
	AAA_Server	Authentication,AAA_Server1	Nisynch	Ok	Verified	No attacks.
		Authentication,AAA_Server2	Niagree	Ok	Verified	No attacks.
		Authentication,AAA_Server4	Secret Nr	Ok	Verified	No attacks.

Done.

Fig. 4: Scyther Tool Result for Mutual authentication

C. *Efficacy Measurements*

To evaluate the cost and size of the cryptographic operations, the parameters specified in [2] were benchmarked. The comparison of computational and communication cost as shown in Fig. 5 indicates that MEC-SMAP takes less computational cost compared to [2], [13], [14], and less communication cost compared to [14]. Though, MEC-SMAP attribute higher communication cost compared to [2] and [13]. Since MEC servers are launched in a considerably resourceful environment, and the authentication is taking place prior to the service migration, higher communication cost can be manageable considering the gained security impact.



(a)

(b)

Fig. 5: Comparison of (a) computational and (b) communication cost of authentication protocols.

V. Prototype Testing Environment

A prototype emulation environment depicted in Fig. 6 was contrived for testing the feasibility of the proposed security protocol. The two MEC environments were emulated using laptops with considerable resource specifications to improve their dynamic nature. The AAA server was launched at a Windows Server 2016 environment bearing Processor: Intel Xeon 2.4 GHz 4 CPU, and RAM: 8 GB. The protocol flow was implemented using Java socket based platform, where interfaces of each entity was programmed under different programs. The two MEC emulators were configured with a bare-metal hypervisor VMWare ESXi to host the dockerized service provisioning environment (i.e. as specified in [15]). RSA-4096, AES-256, SHA-512, $d_{dos} = 4$, and clock skew as 50 ms was used in specifying the protocol. Further, P-256 EDCH described under RFC-5903 was followed in deploying the ECC based PFS mechanism.

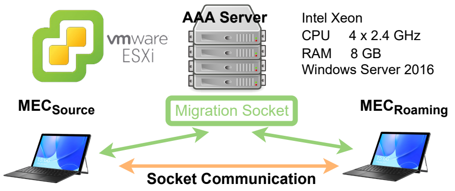
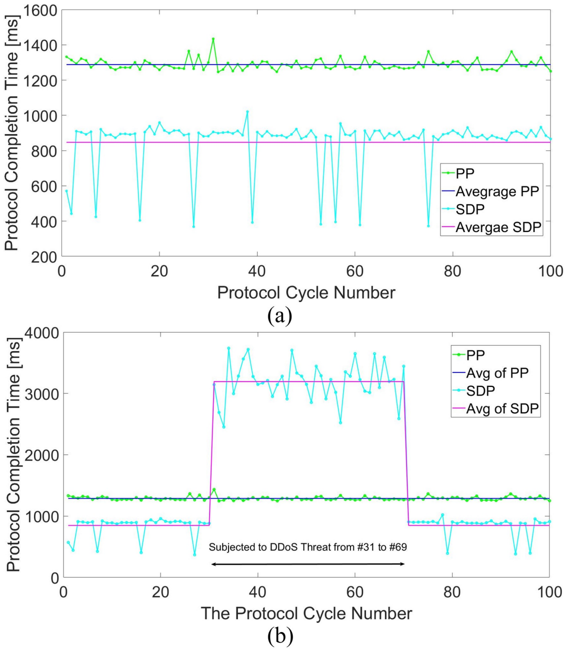


Fig. 6: Prototype MEC Migration Model

In order to evaluate the impact of the embedded security mechanisms in the proposed protocol, several experiments were conducted under two scenarios. We have considered the Proposed Protocol (PP) as the Scenario 1, and we have detached all the instilled security features from the PP, and referred it as the Security Detached Protocol (SDP); considered SDP as scenario 2. In scenario 2 the number of messages has been reduced to 11, where M_2 and M_3 were dropped. Fig. 7-(a) demonstrate the Completion Time (CT) variation of the PP and SDP for consecutive 100 cycles/ runs. The PP converged into an average CT of 1288 ms while SDP convergence time is 847 ms. In Fig. 7-(b), a DDoS attempt was emulated into the protocol between 31st to 69th cycles, and the CT behaviour was plotted. The emulated DDoS attempt was generated assuming 35 DoS agents attacking sequentially. The average of the SDP CT was elevated to 3193 ms during the threat. This increment in the CT justify the use of DoS security measure in MEC-SMAP.



(a)

(b)

Fig. 7: The Impact of the Embedded Security Features into the MEC-SMAP

VI. Conclusion

It is evident that service migration channel of the MEC deployments can be subjected to security threats, and even an intended delay might reflect with negative QoS and QoE. Thus, our proposed MEC-SMAP was designed to cater such requirements of the future networks. We have validated the proposed MEC-SMAP protocol formally. This verification yielded that the PP is robust against known types of attacks, while DoS, tamper, and Replay detection security features are embedded into the protocol in addition. Further, our implementation results suggest that the proposed protocol can be launched feasibly as a pre-migration strategy. The performance in terms of computational cost proves to be quite acceptable considering the asymmetric crypto usage. The novel concept of the security profile, introduced in this research would allow the MEC stations to manage the security even in scarce situations. The MEC-SMAP will be beneficial to stakeholders in improving the feasibility of MEC pragmatic deployment. The future work is directed towards assessing the impact of latency for level of security in this E2E channel.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A Survey on Mobile Edge Computing: The Communication Pserspective,” IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] Y. Zhang, R. H. Deng, E. Bertino, and D. Zheng, “Robust and Universal Seamless Handover Authentication in 5G HetNets,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 2, pp. 858–874, 2019.

[3] X. Zhang, W. Wu, S. Yang, and X. Wang, “Falcon: a Blockchain-based Edge Service Migration Framework in MEC,” Mobile Information Systems, vol. 2020, 2020.

[4] G. Karthick, G. Mapp, F. Kammueller, and M. Aiash, “Formalization and Analysis of a Resource Allocation Security Protocol for Secure Service Migration,” in 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). IEEE, 2018, pp. 207–212.

[5] M. Cui, H. Zhang, Y. Huang, Z. Xu, and Q. Zhao, “A Fountain-Coding Based Cooperative Jamming Strategy for Secure Service Migration in Edge Computing,” Wireless Networks, pp. 1–14, 2021.

[6] P. Ranaweera, A. D. Jurcut, and M. Liyanage, “Realizing Multi-Access Edge Computing Feasibility: Security Perspective,” in 2019 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, 2019, pp. 1–7.

[7] P. Ranaweera, V. N. Imrith, M. Liyanag, and A. D. Jurcut, “Security as a Service Platform Leveraging Multi-access Edge Computing Infrastructure Provisions,” in ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020, pp. 1–6.

[8] A. Braeken, M. Liyanage, P. Kumar, and J. Murphy, “Novel 5G Authentication Protocol to Improve the Resistance Against Active Attacks and Malicious Serving Networks,” IEEE Access, vol. 7, pp. 64 040–64 052, 2019.

[9] D. Dolev and A. Yao, “On the Security of Public Key Protocols,” IEEE Transactions on information theory, vol. 29, no. 2, pp. 198–208, 1983.

[10] R. Canetti and H. Krawczyk, “Universally Composable Notions of Key Exchange and Secure Channels,” in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2002, pp. 337–351.

[11] L. Gong, R. M. Needham, and R. Yahalom, “Reasoning about Belief in Cryptographic Protocols,” in IEEE Symposium on Security and Privacy. Citeseer, 1990, pp. 234–248.

[12] C. J. F. Cremers, Scyther:Semantics & Verification of Security Protocols. Eindhoven university of Technology Eindhoven, Netherlands, 2006.

[13] Y. Zhang, X. Chen, J. Li, and H. Li, “Generic Construction for Secure and Efficient Handoff Authentication Schemes in EAP-based Wireless Networks,” Computer Networks, vol. 75, pp. 192–211, 2014.

[14] D. He, D. Wang, Q. Xie, and K. Chen, “Anonymous Handover Authentication Protocol for Mobile Wireless Networks with Conditional Privacy Preservation,” Science China Information Sciences, vol. 60, no. 5, pp. 1–17, 2017.

[15] G. Dilanka, L. Viranga, R. Pamudith, T. D. Gamage, P. Ranaweera, I. A. Balapuwaduge, and M. Liyanage, “A Novel Request Handler Algorithm for Multi-access Edge Computing Platforms in 5G,” in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2022, pp. 126–131.