A Novel Authentication Protocol for 5G gNodeBs in Service Migration Scenarios of MEC

Pasika Ranaweera, *Member, IEEE*, Awaneesh Kumar Yadav, *Student Member, IEEE*,
Madhusanka Liyanage, *Senior Member, IEEE*, and Anca Delia Jurcut, *Member, IEEE*

Edge computing paradigms were an expedient innovation for elevating the contemporary standards of mobile and Internet networks. As specified in Multi-Access Edge Computing (MEC) standardization, edge computing serviceable infrastructures are running on virtualization technologies to provide dynamic and flexible service instances. Since the inception and operation of the services are executing at the edge level gNodeBs (gNBs), migration of services between gNBs is an imminent occurrence in edge computing that is contriving challenges to its feasible deployment. Security and service level latency requirements are vital parameters for such service migration operations conducted through gNB to gNB (g2g) connecting channels. In this paper, our focus is to ensure identity verification among the parties involved in a service migration through authentication and to secure the migrating content through a robust g2g channel establishment. Our proposed authentication protocol was designed in accordance with the MEC architectural standardization. We have verified the proposed protocol employing four different formal verification techniques: Scyther and AVISPA verification tools, GNY and ROR logical approaches. Further, we have developed the proposed protocol in a test-bed environment emulating the MEC system with an integrated 5G Core network.

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

I. Introduction

MULTI-Access Edge Computing (MEC) is a nascent edge computing paradigm proposed to overcome the limitations of the existing cloud-centric networks. The lack of locational and context awareness attributed to the cloud computing platforms and the untrusted data outsourcing feature is contriving security and privacy issues for the service subscribers. Moreover, managing the trust domain is arduous with its globally dispersed deployment. Thus, edge computing offers the unique opportunity for launching a computing-enabled serviceable platform at the edge of the mobile network within a gNodeB (gNB), where Mobile Network Operators (MNOs) can guarantee the trust-domain intrinsic with General Data Protection Regulation (GDPR) resembled standards.

The emergence of edge computing paradigms has introduced the concept of service migration to cater to the heterogeneous IoT device's ubiquitous connectivity over the mobile network. The MEC-based services are offered from the nearest MEC-enabled gNB to the subscriber. Since the service instance or the program executing at the edge platform originates there, such a service instance is not available in other MEC gNBs. In a situation where the subscriber is traversing beyond the range of the currently serving MEC gNB, the service instance should be migrated to a gNB with MEC capabilities in the proximity of the subscriber-roamed location. Once migrated and configured to the roamed MEC infrastructure, offered service to the consumer continues through the communication channels of the roamed gNB. The Quality of Service (QoS) and Quality of Experience (QoE) aspects of the offered MEC-based service depend entirely on the seamless operation of the migration process. The latency or a delay caused in the migration process will disrupt the service to the consumer device, thereby impacting both QoS and QoE factors negatively. Thus, service migration within edge computing platforms is a weaker aspect of MEC that forecasts inevitable issues.

In a typical gNB-to-gNB (g2g) communication, specialized authentication is not required as all gNBs are registered under an MNO. Though with the advent of 5G, local operators are granted the ability to launch services in the mobile network, and such operators are not quite trustable due to the scalability of 5G. There is always a possibility of a fake gNB being launched by an adversary with replicated communication protocols. Since service migrations are becoming frequent in 5G, g2g communication is becoming a regular function for emerging networks. In addition to the impact it causes on the User Equipment (UE) communication, implications to the service migration process would be severe. This severity is due to service migration conveying mostly executable content or sensitive credentials between the gNBs. Thus, validating the identity of the gNB is critical for 5G and its envisioned use cases. An authentication mechanism can validate the identity of the 5G gNB, and establish a secure migration channel afterwards. A Trusted Third Party (TTP) should be engaged as both entities' identity verification (mutual authentication) cannot be pursued in an ad-hoc or peer-to-peer manner.

One might think that a typical authentication mechanism, as presented in [1] or [2], could be sufficient for this authentication mechanism. Service migration is a unique process requiring a specialized security protocol targeting the Service Migration Channel (SMC). This intended protocol should determine the eligibility of the roaming gNB considering resource availability and SMC network capacity while validating the legitimacy of the migrating virtual instances. Further, different security profiles ought to be applied to the SMC, depending on the application specifications. Thus, the SMC security profile selection should also be communicated via this protocol establishment.

A. Related Work

Zhang et al. in [1] propose a handover authentication protocol for 5G-based Heterogeneous Networks (HetNets) called RUSH. RUSH employed chameleon hash functions considering their trapdoor collision property and blockchain for its tamper resistance, and formal logic and model-based methods were used to verify it. Though the proposed RUSH scheme is not directly related to service migrations, the context in which it forms a secure channel between 5G HetNet gNBs or access points is quite relevant to this study. Yan et al. in [3] developed

Pasika Ranaweera, Madhusanka Liyanage, and Anca Delia Jurcut are with the School of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland. Emails: pasika.ranaweera@ucd.ie, madhusanka@ucd.ie, and anca.jucut@ucd.ie.

Awaneesh Kumar Yadav is with the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India. Email: akumaryadav@cs.iitr.ac.in.

a group handover authentication mechanism for 5G vehicle-to-everything scenarios. The mutual authentication protocol proposed between the gNBs and the vehicles employs certificateless aggregated signatures free from key escrow issues and substantially reduces the signaling overhead. Although this paper accurately involves 5G core network entities in establishing the group handover scheme, edge computing or service migration aspects are not within its scope. Zhang et al. in [4] propose a blockchain-based secure edge service migration framework called Falcon, which enables Virtual Machines (VMs) or containers to be migrated as mobile agent-based carriers to make the migration process more flexible. This framework employs an immutable alliance chain decentralized to edge clouds for improving performance. The identity verification and management engaged in migration is a lacking aspect of this protocol, where a comprehensive security solution is required apart from the blockchain. Cui et al. in [5] introduce a fountain codes-based jamming strategy for service migration scenarios of edge computing environments. This solution contrives a set of Relay nodes to conduct cooperative jamming, which would eventually mislead and deteriorate the illegal eavesdropping quality of the migration channel. This strategy, however, does not provide a solution for authenticity verification among migration entities.

An authentication protocol for service migration scenarios in cloud computing was proposed by Karthick et al. in [2]. This protocol targeted vehicular applications that require migrations between two clouds, and a registration entity is performing the communication of resource allocation securely. Though this protocol has been validated with AVISPA, there are evident issues, such as needless signature exposure that opt for reuse threats, ill consideration of perfect forward secrecy, and Denial of Service (DoS) threats. In addition, the lack of a development environment or any simulated performance metrics indeterminate the feasibility of this protocol.

B. Our Contribution

To the best of our knowledge, no literature is available addressing the security issues of edge-to-edge service migration scenarios. Our work can be considered as the pioneer attempt to solve the security concerns of service migrations. This paper proposes a holistic security protocol for authenticating and establishing a secure channel for pursuing service migrations. The main contributions of this research are stated below.

- Proposing a communication protocol for service migration instigation from an MEC architectural standpoint.
- Ensuring the authenticity and integrity of parties engaged in the service migration process through a federated identity verification approach.
- Securely conveying the credentials or parameters that enable the formation of a security profile.
- Establishing a secure g2g channel for service migrations.
- Validating our claims through formal analysis employing Scyther tool, AVISPA tool, Gong, Needham, and Yahalom (GNY) logic, and Real-Or-Random (ROR) logic.
- Conducting an informal analysis to verify compliance with the proposed security goals.
- Developing a prototype MEC environment to deploy the proposed protocol.

C. Paper Organization

The rest of the paper is organized into seven sections. Section II introduces the proposed MEC service migration security framework along with the novel authentication model proposed for it, while the considered threat model and the security goals of the proposed protocol are specified afterwards. In Section III, the design aspects of the security protocol are discussed and aligned to the considered service-migrating MEC system model. The informal analysis of the proposed protocol is presented in Section IV, while the formal analysis results employing Scyther, AVISPA tools, and GNY, ROR logics are presented in Section V. Section VI presents the computational and communication cost of the proposed protocol, while the details of the developed prototype MEC environment are presented in Section VII. Finally, Section VIII concludes the paper.

II. Proposed MEC Service Migration Security Framework and the Threat Model

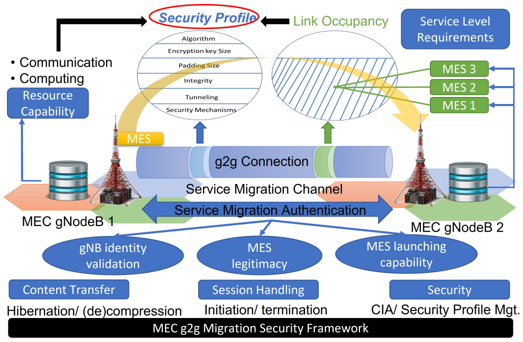


Fig. 1: Proposed MEC Service Migration Security Framework

In this paper, we assume that service migration is occurring as VMs or lightweight containers as specified in [4], [6]. In the MEC architecture specified in [7], Mobile Edge Services (MESs) are launched in Mobile Edge Hosts (MEHs). If we assume the MEHs are launched as VMs, and corresponding MESs are contained in light-weight containers as Mobile Edge Apps (ME Apps), migration of a particular MES is represented by transferring the contents of a container or a hibernated image of the said container. In such an instance, the hibernated image of the MES should be re-configured and launched at the roamed gNB MEC environment after the migration. The framework illustrated in Fig. 1 is proposed to perform the task of service migration within the g2g channel, embedded with optimal security strategies that guarantee maximal service level efficiency.

A. Proposed Service Migration Security Framework for MEC

The proposed Service Migration Security Framework (SMSF) for MEC-based gNBs is formed for two primary purposes. The first purpose is the mutual authentication of entities engaged in a service migration scenario. In the authentication phase, the main functions are 1) validation of the gNB identities, 2) determining the legitimacy/ integrity of the migrating MES, and 3) launching capability of the migrating MES at the roaming gNB MEC environment. This phase is concluded by securely transferring the credentials and parameters corresponding to the security profile.

Security Profile (SP): SP embeds the layers of key size, padding scheme, and size, integrity checking mechanism, tunneling scheme, and other additional security schemes (i.e., for DoS mitigation, jamming as in [5]). SPs are identified from their Security Profile Index (SPI), which makes the operations of selection and application more convenient. Each $SP(K_M)_i$ corresponds to a different set of layers $l_0, l_1, l_2, ...l_j$, where l_0 represents whether the SP is in the tunneling mode or not. If not, the rest of the layers are subjected to various security encryption algorithms denoted by A. The depth of layers (i.e., j) and its applied mechanism will determine the security level (L) and the incurred costs in terms of computation (C) and communication (T) perspectives. T results in the resulting BW utilization each $SP(K_M)_i$ is deploying, while C accounts for the average processor utilization for each $SP(K_M)_i$ processing.

The second purpose is to optimize the application of security level in accordance with the current service level requirements, considering the available Bandwidth (BW) of the g2g channel. In addition to the two primary purposes, the functions of content transfer, session handling, and security are handled by this framework. However, the determination of the optimized security level and modeling of the SP exceed the scope of this paper. Thus, this paper only focuses on the first purpose.

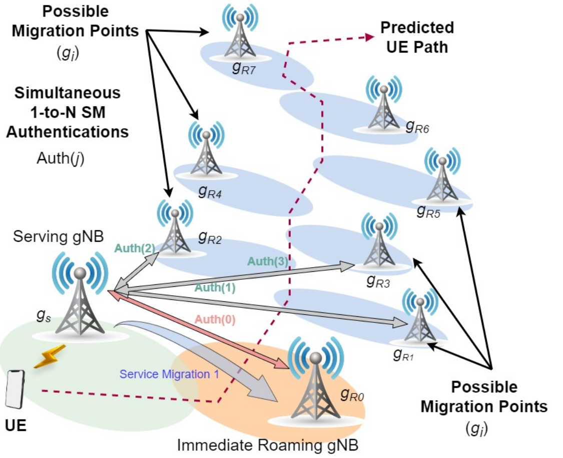


Fig. 2: Proposed MEC Secure Service Migration Model

To improve the efficiency in service migration scenarios, UE path prediction models can be considered as specified in [8], [9]. Thus, simultaneous 1-to-N authentication can be conducted among the gNBs that are positioned on the predicted path of the UE. As illustrated in Fig. 2, possible migration points are notified as g_{Ri}, where each authentication session is specified as $Auth(i)$, in which $i \in 0, 1, 2, ..., I$. Though this process is simultaneous, priority is given to the proximate g_{Ri} points, and I is determined based on the UE predicted path and the reach of gNB_S. This paper focuses on establishing a single $Auth(i)$ session to maximize security.

B. Threat Model

To examine the robustness of the proposed protocol, we employ the Delev-Yao (DY) [10] threat model. The adversary's capabilities are as follows

1) The adversary (A) has total control over a wireless channel where an attacker may remove, modify, or inject legitimate messages.
2) A can only guess one credential in polynomial time because it is impossible for the attacker to guess several values at once, such as identification or password, at the same time.
3) A can intercept messages from many sessions and launch a traceability attack.
4) A can act as a middleman and launch a man-in-the-middle attack. Adversary stealthily relays/possibly modifies communications between two parties that believe they are conversing directly with each other.
5) A may also reveal some session secrets.
6) A can also obtain the private keys of communicating parties.
7) Adversay can get the data stored on gNB_{Source} and $gNB_{Roaming}$.
8) Under this model, we also assume that TTP and MVA servers are secure and inaccessible to the adversary.
9) It is possible that any of gNB_{Source} or $gNB_{Roaming}$ might be compromised.

C. Security goals of the proposed protocol

The following are the security goals [1], [11], [12] that the designed authentication technique must meet.

- Mutual authentication: It states that before sharing any private or personal information, communication parties must check each other's legitimacy.
- Confidentiality: The identities of communication parties should not be communicated in plain text over insecure public channels.
- Perfect Forward Secrecy (PFS): This concept ensures that even if an attacker is capable of acquiring long-term credentials, revealing the prior session keys or secret content within the messages is not viable.
- Replay attack protection: It ensures that it is impossible for an attacker to replay the old message.
- Protection from DoS attack: It is difficult for an attacker to create network congestion by sending reused messages.
- Protection from Traceability attack: It is hard for an attacker to determine whether the same device is sending two distinct authentication requests.
- Protection from malicious gNB_S or gNB_R: It assures that the attacker is unable to retrieve the previous credentials even if the physical access of either gNB_S or gNB_R is seized.
- Protection from Key-Escrow attack: it state that, in any asymmetric-based encryption scheme, if the key generation authority is fully trusted and all the private keys of the users are generated by this authority, then the authority can decrypt all the ciphertexts with the help of these generated keys, but he/she can not get the previous session keys [13].

III. Security Protocol Design

Due to the distributed nature of the MEC edge computing deployments, and its system level existing at a distance (i.e., in the core network), a federated identity verification mechanism was followed in designing this protocol. In other terms, the identity of an individual entity was verified through multiple parties via multiple means. Moreover, the MEC systems' reliance on the 5G core network components (i.e., especially Access and Mobility Management Function-AMF, Session

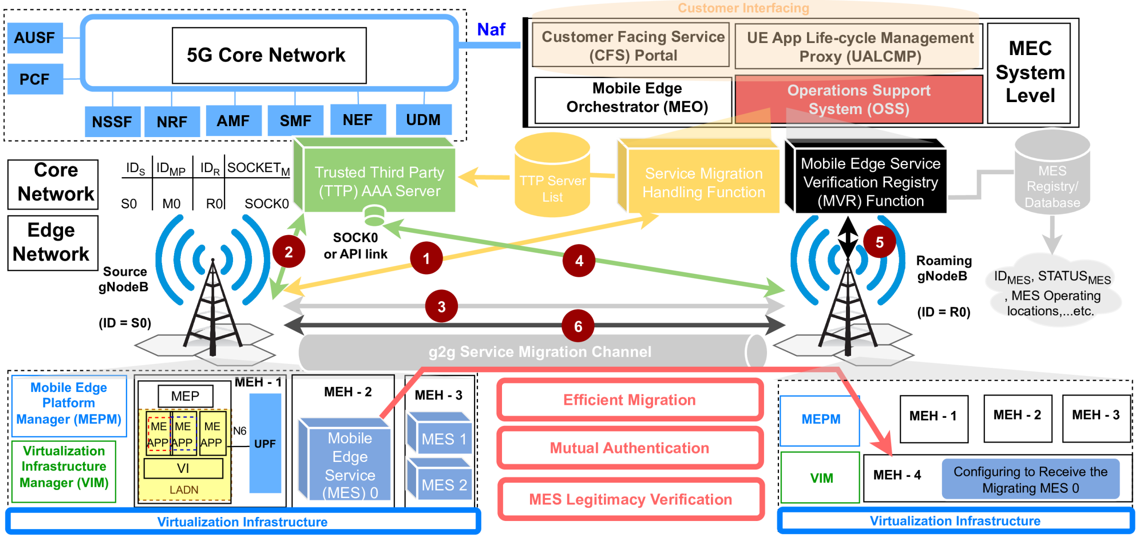

AUSF
PCF
5G Core Network
NSSF
NRF
AMF
SMF
NEF
UDM
Naf
Customer Interfacing
Customer Facing Service (CFS) Portal
UE App Life-cycle Management Proxy (UALCMP)
MEC System Level
Mobile Edge Orchestrator (MEO)
Operations Support System (OSS)
Core Network
Edge Network
ID_S
ID_{MP}
ID_R
$SOCKET_M$
S0
M0
R0
SOCK0
Trusted Third Party (TTP) AAA Server
TTP Server List
Service Migration Handling Function
Mobile Edge Service Verification Registry (MVR) Function
MES Registry/ Database
SOCK0 or API link
Source gNodeB
(ID = S0)
Roaming gNodeB
(ID = R0)
ID_{MES}, $STATUS_{MES}$, MES Operating locations,...etc.
1
2
3
4
5
6
g2g Service Migration Channel
Mobile Edge Platform Manager (MEPM)
Virtualization Infrastructure Manager (VIM)
MEP
ME APP
ME APP
ME APP
VI
LADN
N6
UPF
MEH - 1
MEH - 2
MEH - 3
Mobile Edge Service (MES) 0
MES 1
MES 2
Efficient Migration
Mutual Authentication
MES Legitimacy Verification
MEPM
VIM
MEH - 1
MEH - 2
MEH - 3
MEH - 4
Configuring to Receive the Migrating MES 0
Virtualization Infrastructure
Virtualization Infrastructure

Fig. 3: Holistic Service Migration Authentication Process from a MEC Architectural Viewpoint

Management Function-SMF, and User Plane Function-UPF) makes the designing of the protocol flow complicated [7], [14]. The 3GPP 5G architecture specified in TS 23.501 [15] published under the 3GPP Release 15 was followed in the designs of this protocol. As one of the goals of this protocol is to unburden the main MEC entities of the security and authentication concerns, a TTP can be employed as a provisioning service. The proposed protocol attribute functional and architectural goals in addition to the security goals defined under the sub-section II-C. Therefore, a complete set of goals targeted by this protocol is stated in sub-section III-B.

A holistic perspective of the proposed security protocol and its connections to the MEC architectural components, operational sequence, and functional parameters are illustrated in Fig. 3. The entity Mobile Edge Platform Manager (MEPM) is the orchestrator of the edge level, and the Virtualization Infrastructure Manager (VIM) performs the hypervisor function on virtualized resource management. MEHs are the main operational elements of the MEC system, where their ME APPs or MESs launched in the Virtualization Infrastructure (VI) are governed by the Mobile Edge Platform (MEP) entity. The Local Area Data Network (LADN) within the MEH steers the traffic with the assistance of the UPF. Further information on these entities can be found in [6]. The noteworthy entities in the same figure are described or defined below.

Operations Support System (OSS): This is one of the main entities at the MEC system level. In fact, the MEC system level is interfacing with the 5G core network through the Naf interface [16]. According to the ETSI documentation, OSS is responsible for handling the user access authorization and subscriptions with proper distinguishing of the various service types forwarded from UE App Life-cycle Management Proxy (UALCMP) and Customer Facing Service (CFS) portal [17]. As the main authority for authorization in the MEC domain, it is our main assumption that OSS is capable of handling the migration authorizations. Specifically, we are assuming that OSS is catering the functions of 1) serving as the main registry for storing the available TTP servers tasked with performing Authentication, Authorization, and Accounting (AAA) functions; 2) registry/ database for storing live MES information within the MEC system. There could be many TTP servers as MEC is a distributed architecture. But the gNodeB is considered as the edge level of the MEC. There are several $gNBs$ controlled under a system level.

Purpose of the TTP Server: Due to the allowance in 5G and B5G technologies to launch micro or macro-cell level $gNBs$ from local 5G operators, registering and monitoring all such $gNBs$ under the MEC system registries (i.e. OSS) is not viable, as certain services might be placed locally by their service providers and are launched only for application-specific instances within a limited domain. Due to this reason, there could be fake base stations or fake gNB attacks perpetrated by resourceful adversaries capable of intercepting the 5G radio bands. TTP contrives the required trust domain for the assigned geographical area, specific for migration processes. TTP offers a Migration Authentication as a Service (MAaaS) which alleviates the burden on the MEC system regarding handling secure service migrations. If a migration is required by a specific gNB, it should first register under the TTP for the migration. But this registration is only applicable to a single MES migration. However, a single TTP migration registration creates credentials required for I number of $Auth(i)$ sessions for the considered gNB_S as indicated in Fig. 2.

Mobile Edge Service Verification Registry (MVR): This is an entity we are proposing to act as the global registry for all the MESs operating under the MEC service provider. The MVR is monitoring the service instances of each registered MES across the MEC domain, and updates its registries regarding the status, launched gNB location, and migration status. In the MEC service provisioning environment, physical infrastructure is decoupled from the virtualization domain, and software operations are preferred and dominating. Such a priority given to the softwarized entities can be exploited by perpetrators to induce autonomous constructs within and obscured instilled to the code and presents the opportunity to propagate through the virtualized MEC environment with ease. Therefore, security engineers should treat physical and

TABLE I: Main Notions and Acronyms with their Definition/ Description

Acronym	Definition	Description
MEC Specific		
gNB	gNodeB	MEC enabled 5G New Radio Base Station
UE	User Equipment	The apparatus that is interfacing to the mobile network on behalf of the user
g2g	gNB-to-gNB	A connection between two gNBs
SMC	Service Migration Channel	g2g channel that is employed for service migration process
MES	Mobile Edge Service	The MEC service instance running in the MEC platform to cater services to the UE
gNB_S	Source gNB	gNB where the MES is currently running and commencing the service migration process
gNB_R	Roaming gNB	gNB that the MES is intended to migrate
TTP/AAA	Trusted Third Party/ AAA Service	This is an Authentication, Authorization, and Accountability (AAA) service formed to conduct identity verification for intended migrations
$SOCK_{TTP}$	TTP Socket	Socket or an API link for contacting the TTP server
$SOCK_{TTP_M}$	TTP Migration Socket	A unique socket or an API link generated by the TTP for a specific migration session
MVR	MES Verification Registry	MES monitoring functionality of the MEC system that tracks the accountability of MESs
MES Specific		
REQ_{MES}	MES Requirements	Minimum required storage (HDD_{min}), processor (CPU_{min}), memory (RAM_{min}), and bandwidth (BW_{min}) resources, or minimum specifications to execute an APP or an MES.
RE_R	Resource Eligibility	Eligibility of the gNB_R in terms of resources (computing/storage/networking) and SMC capacity to launch the MES. RE_R indicates whether the considered MES has satisfied the REQ_{MES}.
ID_{MP}	Migration Process ID	Represent a migration related to a single MES at a certain instance.
$DATA_{MES}$	MES Information/ Specifications	The parametric information of a certain MES such as ID_{MES}, $Name_{MES}$, $ID_{Container-MES}$, $OS_{Container-MES}$, $IP_{MES-SERVER}$, and QCI_{MES}
$STATE_{MES}$	MES Status	The running status of the MES that indicates the currently consuming processor and memory configuration to be conveyed to the gNB_R, for allocation of resources.
Security Specific		
SP	Security Profile	A template that specifies the different security features and parameters applied for a channel
ECC	Elliptic Curve Cryptography [18]	A cryptographic method modeled based on the arithmetic of elliptic curves
Enc[]	Enc[Payload, K_X]	Encryption : payload encrypted with the key K, either belonging to party X, or symmetric
Sig[]	Sig[Payload, K_Y]	Signing: Signed with the private key of party Y
PuK_X	Public Key of X	Used for encryption and unsigning or verifying
PrK_X	Private Key of X	Used for decryption and signing
SyK_{XY}	Symmetric Key between X and Y	Employed for encryption and decryption
n_X	Nonce	Nonce generated by party X
P_X	ECC Point	A point on an elliptic curve computed from the random secret $a \in Z_n$, that takes the form $P_X = a.M$, where M is a public elliptic point
$H[input]$	Hash	Hash value of the $input$. Typically SHA-256 hashing algorithm is used for this process
MIH	Message Identification Header	Message Identification Header is the application layer message naming/identifying tag, that is standardized in accordance with the protocol
TS	Timestamp	The current timestamp of the system triggered by an event

softwarized entities in this era separately. Hence, a malicious MES could penetrate its MEC environment even if a gNB is a legitimate entity. Such an MES could prompt a migration request just to propagate its malicious content to other gNBs. Therefore, it is important to verify the legitimacy of each MES that is prompting a migration. Among other tasks related to MESs, MVR is primarily tasked with handling the validation process of the MESs that are requesting a migration.

As specified earlier, MVR can be launched as a functional construct of the OSS in the MEC environment. When migration is prompted by an MES in the gNB_S towards gNB_R, an MES verification request (MES_VER_REQ) is sent from the gNB_R to the MVR that embeds the corresponding identities of MES (ID_{MES}), gNB_R (ID_R), and gNB_S (ID_S). The MVR would check its entries for the provided ID_{MES}, and whether that particular MES is registered under the ID_S. If yes, a verification code $CODE_{MES}$ is sent to the gNB_R while the same $CODE_{MES}$ is forwarded to the gNB_S. Thus, both gNB_R and gNB_S can attain the verification code and validate the MES legitimacy.

Table I specifies the Notions and acronyms used in the presented description regarding our proposed protocol.

A. Assumptions

- A1 - Assuming that authentication takes place prior to initiating migration in the pre-migration stage.
- A2 - Assuming that possible gNB_Rs are already selected and known in terms of their network addresses.
- A3 - Assuming that the most suited (e.g., through prediction based on distance) gNB is selected as the 1st gNB_R in the sequence of 1-to-N authentication sessions.
- A4 - Assuming that all the MESs should be pre-registered under the OSS, MVR, UALCMP, or CFSP.
- A5 - All the 1-to-N authentication sessions occur independently and in parallel to each other, where gNB_S is equipped with a sufficient number of 5G NR interfaces.
- A6 - Assume that all the links directed from gNB_S have sufficient and a dedicated BW to convey the messages relating to the authentication protocol so that maximum security measures can be applied.
- A7 - Assume that re-transmission protocols of the L2 and L4 of the TCP/IP stack are performing independently to detect packet losses or errors during transmission.
- A8 - Assume the entire MEC system is synchronized and Timestamp (TS) errors are negligible.
- A9 - All TTPs are trusted, and OSS contains the list of all the TTPs registered under the MNO.
- A10 - $SOCK_{TTP}$ and $SOCK_{TTP_M}$ (please refer to Table I for the definitions) have different port numbers; hence the corresponding services can operate independently and simultaneously.
- A11 - Assume that all the public certificates related to the entities gNB_S, gNB_R, OSS, MVR, and TTP are handled by the certificate authority operating within the MEC trust domain and act as the trust anchor.
- A12 - For every protocol session/ segment, all the nonces, timestamps, and HMACs are newly generated, and the

notions are only bound to the specified protocol session.

B. Goals of the Proposed Security Protocol

The main goals of the proposed protocol are mentioned below.

G1 Mutually authenticate each gNB_R selected to initiate the migration with gNB_S in pre-migration stage.

G1.1 Authenticate gNB_R to gNB_S

G1.2 - Authenticate gNB_S to gNB_R

G1.3 Form an identity verification mechanism for gNB_S

G2 Mitigate possible DoS or DDoS attempts on server interfaces of the proposed system model

G3 Validate the legitimacy of each migrating MES

G3.1 Propose a governing and monitoring entity for MESs under the MEC system

G3.2 Propose a method to authenticate and validate the MESs to the governing entity

G4 Evaluate the Eligibility of gNB_R to host the MES

G5 Propose a secure SP selection process

G5.1 Create a secure migration master key K_M, for migration session establishment

G5.2 Propose a secure method to share available SPs and to conduct the selection process

G6 Migration session establishment

G6.1 - Integrate the SP to the migration session

G6.2 - Propose a method to migrate the executing MES

C. Proposed Security Protocol

In the protocol segments described below, several methods are followed to ensure the mutual-authentication among the entities involved in communication. Such methods are:

- Public Key Encryption - The common RSA encryption based on X509 certificates.
- Elliptic Curve Cryptography (ECC) - for PFS assurance.
- Timestamp Utilization and Freshness Verification.
- Hashing Functions - SHA-512.
- Nonces and Nonce Verifying Hashes.
- Signatures - Employed to validate the authenticity of the communicating party.
- Hashed Message Authentication Code (HMAC)
- DoS Puzzle - based on [18].

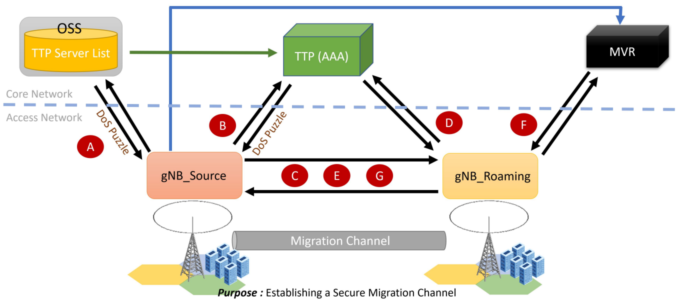


Fig. 4: High-Level Illustration of the Proposed Protocol

Fig. 4 indicates the high-level view of the g2g authentication protocol for migration. The steps of the overall process are listed below.

- Step A: gNB_S reaches out to OSS for requesting contact details of the assigned TTP entity.
- Step B: gNB_S is contacting the migrating AAA service at the TTP. TTP registers the respective migration request and create a unique API link/socket specific to this migration while session IDs are created.
- Step C: With the received migration credentials, gNB_S reaches out to gNB_R.
- Step D: gNB_R uses the unique socket/link to access the TTP server, and verifies the request forwarded from the gNB_S for establishing mutual authentication.
- Step E: The MES and resource requirements information is forwarded to the gNB_R by gNB_S.
- Step F: gNB_R verifies the legitimacy of the MES via MVR, and investigates the resource capability to host the MES, while a migration master key is derived from the credentials shared via the previous steps.
- Step G: Utilizing the generated master key, gNB_S is sending a set of suitable SPs to gNB_R securely. Then gNB_R informs gNB_S on the selected SP, while the authentication protocol concludes in compliance.

The following sub-sections discuss each section of the protocol extensively. The current protocol development only focuses on authentication prior to migration session creation and assumes A5 and A6.

1) Part A : gNB_S to OSS Communication for Acquiring the TTP Credentials

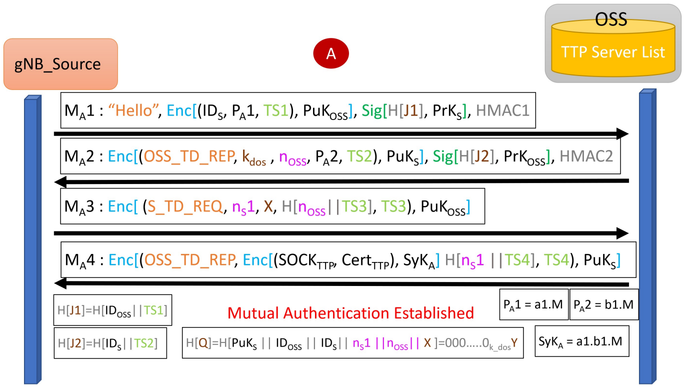

gNB_Source
A
OSS
TTP Server List
M_A1 : "Hello", Enc[(ID_S, P_A1, TS1), PuK_{OSS}], Sig[H[J1], PrK_S], HMAC1
M_A2 : Enc[(OSS_TD_REP, k_{dos} , n_{OSS}, P_A2, TS2), PuK_S], Sig[H[J2], PrK_{OSS}], HMAC2
M_A3 : Enc[(S_TD_REQ, n_S1, X, H[n_{OSS}||TS3], TS3), PuK_{OSS}]
M_A4 : Enc[(OSS_TD_REP, Enc[($SOCK_{TTP}$, $Cert_{TTP}$), SyK_A] H[n_S1 ||TS4], TS4), PuK_S]
H[J1]=H[ID_{OSS}||TS1]
Mutual Authentication Established
P_A1 = a1.M
P_A2 = b1.M
H[J2]=H[ID_S||TS2]
H[Q]=H[PuK_S || ID_{OSS} || ID_S|| n_S1 ||n_{OSS}|| X]=000.....0_{k_dos}Y
SyK_A = a1.b1.M

Fig. 5: Part A of the Proposed Security Protocol that takes place between gNB_S and the OSS

In order to communicate with the TTP, the gNB_S should first acquire the contact information of the relevant TTP assigned for the geo-domain. Assuming A11, gNB_S initiates the communication with OSS. The protocol flow of this segment A is illustrated in Fig. 5. In the first message M_A1, gNB_S chooses a random number $a1$ ($a1 \in Z_n$) to compute $P_A1 = a1.M$, where M is an ECC public point. Along with P_A1, "Hello" MIH, ID_S (i.e. Identity of the gNB_S in the MNO network), and $TS1$ are embedded and encrypted with the PuK_{OSS}. In the same message, the content $J1 = ID_{OSS}||TS1$ is hashed signed by PrK_S, while the $HMAC1 = H[ID_S||P_A1||ID_{OSS}||TS1]$ is appended. At the OSS end, ID_S is browsed in the gNB registry, while the freshness of $TS1$ and $HMAC1$ validity is inspected. The OSS then selects relevant MIH, k_{DoS}, compute $P_A2 = b1.M$ with the randomized selection of $b1$ ($b1 \in Z_n$), and generate n_{OSS} to be included inside the encrypted envelop. The $H[J2] = H[ID_S||TS2]$ is signed by the PrK_{OSS}, and $HMAC2 = H[k_{DoS}||n_{OSS}||ID_S||P_A2||TS2]$ is computed to form the M_A2 along with the encrypted content. Upon receiving M_A2, gNB_S extract k_{DoS}, P_A2 and n_{OSS} through decryption, while verifying OSS signature, $TS2$, and $HMAC2$.

After generating n_S1, gNB_S computes the puzzle using $H[PuK_S||ID_S||ID_{OSS}||n_S1||n_{OSS}||X] = 0_1 0_2 ... 0_{k_{DoS}} Y$. With X determined, the gNB_S composes M_A3 with the MIH, $n_S1, X, H[n_{OSS}||TS3]$, and $TS3$ within the encryption. Since signatures are already verified, they are no longer required. Upon receiving M_A3, OSS conducts $TS3$ and OSS nonce verification while recomputing the puzzle utilizing X to verify the compliance on the complexity parameter. After validating the identity of the gNB_S and detecting the request as a non-DoS attempt, OSS forwards the TTP SOCKET (i.e. combination of IP address and the port number of the AAA server), TTP Certificate, along with the n_S1 nonce verification to the gNB_S encrypted with PuK_S in M_A4. As $SOCK_{TTP}$ and $Cert_{TTP}$ are the selected secrets on this protocol segment, we employ light-weight AES encryption as a double-encryption ploy, where the key deriving parameters were exchanged through the ECCDH method to ensure PFS. Thus, two secrets are encrypted with AES using the key $SyK_A = a1.b1.M$. Once received at the requester end, decrypted, and validated, mutual authentication is established between gNB_S and OSS. Mutual authentication is validated by means of signatures, nonces, and timestamps. Though this is a singular connection, mutual authentication is vital to extend the trust domain to the MEC system level. Though it is not indicated in Fig. 5, the OSS function notifies the TTP server of the gNB_S request anchored through ID_S as in Fig. 4. We assume this communication is secure as this is extended within the MEC system-level trust domain.

2) Part B: TTP and the gNB_S communication for obtaining the TTP link for Migration Registration at the TTP

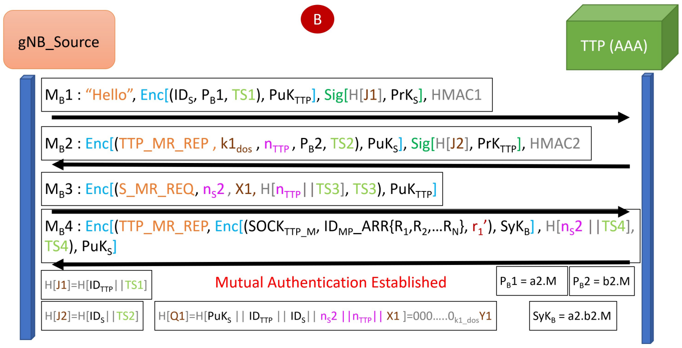

B
gNB_Source
TTP (AAA)
M_B1 : "Hello", Enc[(ID_S, P_B1, TS1), PuK_{TTP}], Sig[H[J1], PrK_S], HMAC1
M_B2 : Enc[(TTP_MR_REP , $k1_{dos}$, n_{TTP} , P_B2, TS2), PuK_S], Sig[H[J2], PrK_{TTP}], HMAC2
M_B3 : Enc[(S_MR_REQ, n_S2 , X1, H[n_{TTP}||TS3], TS3), PuK_{TTP}]
M_B4 : Enc[(TTP_MR_REP, Enc[($SOCK_{TTP\_M}$, ID_{MP}_ARR{R_1,R_2,...R_N}, r_1'), SyK_B] , H[n_S2 ||TS4], TS4), PuK_S]
H[J1]=H[ID_{TTP}||TS1]
Mutual Authentication Established
P_B1 = a2.M
P_B2 = b2.M
H[J2]=H[ID_S||TS2]
H[Q1]=H[PuK_S || ID_{TTP} || ID_S|| n_S2 ||n_{TTP}|| X1]=000.....0_{k1_dos}Y1
SyK_B = a2.b2.M

Fig. 6: Part B of the Proposed Security Protocol that takes place between gNB_S and the TTP

The gNB_S should first register under the migration TTP server so that it will issue the relevant credentials to initiate the migration authentication. This registration phase B is depicted in Fig. 6. The request M_B1 is sent by gNB_S to the TTP using the $SOCK_{TTP}$ provided by the OSS, and utilizing the TTP certificate. The initial request M_B1 includes the $ID_S, TS1$, and P_B1 (i.e. $P_B1 = a2.M$, $a2 \in Z_n$) within the encrypted envelop, while the signature formed with the hash $H[J1] = H[ID_{TTP}||TS1]$ and the $HMAC1 = H[ID_S||ID_{TTP}||P_B1||TS1]$ have been embedded to it. Upon receiving this message, TTP will ensure the freshness of the message from the decrypted $TS1$ and verifies the signature to guarantee it was sent by gNB_S. The integrity is validated from the $HMAC1$. After all the verification steps, ID_S is put under a temporary migration registration. As TTP represents a server function, for DoS mitigation, $k1_{DoS}$ is selected, P_B2 is computed from the selected $b2$ (i.e. $P_B2 = b2.M$, $b2 \in Z_n$), and n_{TTP} nonce is generated. These three parameters are sent within the encrypted reply of M_B2, along with the corresponding TTP signature that include $H[J2] = H[ID_S||TS2]$ and $HMAC2 = H[k1_{DoS}||n_{TTP}||ID_S||P_B2||TS2]$. gNB_S decrypts the message and performs the relevant checks on the MIH, TTP signature, $TS2$, and $HMAC2$. After they are verified and n_S2 is generated, it will determine $X1$ from the puzzle $H[Q1]$ utilizing $k1_{DoS}$: $H[PuK_S||ID_S||ID_{TTP}||n_S2||n_{TTP}||X1] = 0_1 0_2 0_3 ... 0_{k1_{DoS}} Y1$.

The next message M_B3 from gNB_S includes n_S2 and $X1$ parameters along with the TTP nonce verification $H[n_{TTP}||TS3]$ and timestamp within the encrypted envelop. Once the TTP receives and decrypts M_B3, nonce verification is checked, DoS Puzzle is checked using $X1$, while freshness check follows. The TTP then creates the migration socket $SOCK_{TTP_M}$ and exposes it, where it is specific and unique for the migrations initiated by gNB_S. Only gNB_S can access that socket authenticated by ID_S. Then Migration Process Identities (i.e. ID_{MP}s) specific for the gNB_S are generated. These ID_{MP} will be stored in an array: $ID_{MP}ARR(R_i) = [ID_{MP(R0)}, ID_{MP(R1)}, ID_{MP(R2)}, ...ID_{MP(RI)}]$, where each $ID_{MP(Ri)}$ represents the ID_{MP} specific for the relevant migration point g_{Ri} or gNB_R station, while the size of I is dependent on the accuracy and the range of the predicted path of the UE in reference to Fig. 2. A random value r_1 is generated and its modular value $r_1' = r_1 mod N$ is sent to the gNB_S along with the $SOCK_{TTP_M}$, and $ID_{MP}ARR(R_i)$ encrypted by AES employing SyK_B (i.e. $SyK_B = a2.b2.M$); along with the hashed S nonce $H[n_S2||TS4]$ forming M_B4. Upon receiving and decrypting M_B4, gNB_S will store the received information for further processing in the next stages. At this point, mutual authentication is established between the gNB_S and the TTP.

3) Part C and D : gNB_S to gNB_R initial communication prior to MES verification

In this stage, as illustrated in Fig. 7, both gNB_S and gNB_R entities are verified to each other leveraging the TTP connectivity shared through the $SOCK_{TTP_M}$. gNB_S is initiating the migration request towards gNB_R from M_C1, including $ID_S, ID_{TTP}, ID_{MP(R0)}, SOCK_{TTP_M}, Cert_{TTP}, n_S3$, $TS1$, and P_C1 (i.e. $P_C1 = a3.M$, $a3 \in Z_n$) within the encrypted envelop. In addition, gNB_S signature embedding $H[J1] = H[ID_R||TS1]$, and $HMAC1 = H[ID_S||ID_{TTP}||ID_{MP(R0)}||SOCK_{TTP_M}||Cert_{TTP}||n_S3 ||ID_R||P_C1||TS1]$ generated with all the stats in the encrypted and signature envelops are appended. The received M_C1 at the gNB_R is decrypted at first while signature, freshness, and $HMAC$ validations are carried out. Then gNB_R utilizes the $SOCK_{TTP_M}$ to establish the specific connection to

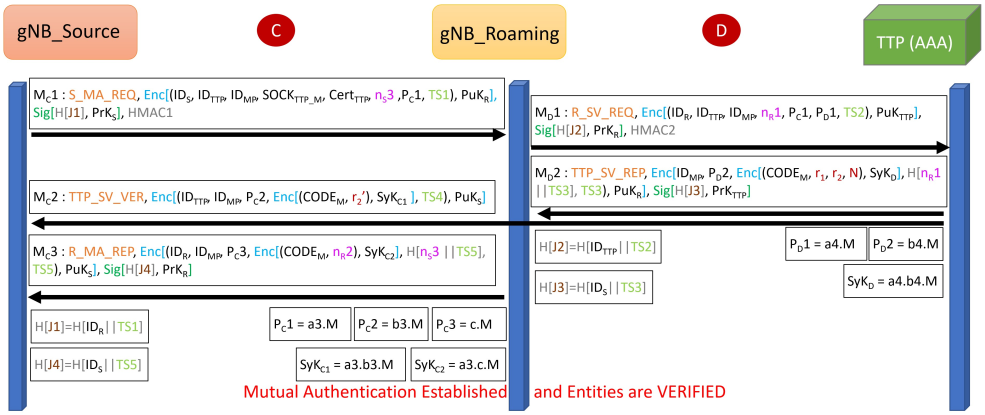

gNB_Source
C
gNB_Roaming
D
TTP (AAA)
M_C1 : S_MA_REQ, Enc[(ID_S, ID_{TTP}, ID_{MP}, $SOCK_{TTP\_M}$, $Cert_{TTP}$, n_S3 ,P_C1, TS1), PuK_R], Sig[H[J1], PrK_S], HMAC1
M_C2 : TTP_SV_VER, Enc[(ID_{TTP}, ID_{MP}, P_C2, Enc[($CODE_M$, r_2'), SyK_{C1}], TS4), PuK_S]
M_C3 : R_MA_REP, Enc[(ID_R, ID_{MP}, P_C3, Enc[($CODE_M$, n_R2), SyK_{C2}], H[n_S3 | | TS5], TS5), PuK_S], Sig[H[J4], PrK_R]
H[J1]=H[ID_R | | TS1]
H[J4]=H[ID_S | | TS5]
P_C1 = a3.M
P_C2 = b3.M
P_C3 = c.M
SyK_{C1} = a3.b3.M
SyK_{C2} = a3.c.M
M_D1 : R_SV_REQ, Enc[(ID_R, ID_{TTP}, ID_{MP}, n_R1, P_C1, P_D1, TS2), PuK_{TTP}], Sig[H[J2], PrK_R], HMAC2
M_D2 : TTP_SV_REP, Enc[ID_{MP}, P_D2, Enc[($CODE_M$, r_1, r_2, N), SyK_D], H[n_R1 | | TS3], TS3), PuK_R], Sig[H[J3], PrK_{TTP}]
H[J2]=H[ID_{TTP} | | TS2]
H[J3]=H[ID_S | | TS3]
P_D1 = a4.M
P_D2 = b4.M
SyK_D = a4.b4.M
Mutual Authentication Established
and Entities are VERIFIED

Fig. 7: Part C and D of the Proposed Security Protocol that takes place between gNB_S, gNB_R and the TTP

TTP, and formulates the encrypted envelop using $Cert_{TTP}$ including $ID_R, ID_{TTP}, ID_{MP(R0)}, n_R1, P_C1, TS2$, and P_D1 (i.e. $P_D1 = a4.M$, $a4 \in Z_n$). Since the connection information was shared by an outside party, verifying the identity of the TTP is vital for gNB_R. Thus, both a nonce and the gNB_R signature $H[J2] = H[ID_{TTP}||TS2]$ are embedded in M_D1, in addition to $HMAC2 = H[ID_R||ID_{TTP}||ID_{MP(R0)}||n_R1||P_C1||P_D1||TS2]$.

Upon receiving the message M_D1 from gNB_R through the exposed socket $SOCK_{TTP_M}$, the message will be decrypted and the usual violation detection/ verification schemes are carried out. With the received information, $ID_{MP(R0)}$ is anchored into the TTP migration registry for locating ID_S and cross-checking with ID_{TTP}. Then the ID_R is temporarily registered for a possible migration. The migration code is generated $CODE_M = H[ID_S||ID_R||ID_{MP(R0)}||n_{RAND}]$ to verify the migration registration with all the parties. n_{RAND} is a random nonce known only to TTP. This $CODE_M$ is disseminated to both gNB_R and gNB_S with the messages M_D2 and M_C2 respectively. The M_D2 include the $ID_{MP(R0)}$ to improve the convenience over the browsing through the migration registry, and the generated $CODE_M$. TTP performs several computations to select the values r_2 and r_2', where r_1 and r_1' are already available, where $r_2' = r_2 mod N$. The r values are selected as $[r_1, r_1', r_2, r_2'] : (r_1, r_2 > N) \wedge (r_1' \times r_2' < N)$.

The M_D2 contain r_1, r_2, N, and $CODE_M$ within the AES encryption generated from SyK_D (i.e. $SyK_D = a4.b4.M$); in addition to the $ID_{MP(R0)}$, P_D2 (i.e. $P_D2 = b4.M$, $b4 \in Z_n$), hashed nonce $H[n_R1||TS3]$, and the TTP signature. With this reply successfully received and verified, gNB_R is ensured of the legitimacy and trustworthiness of TTP. In M_C2, $CODE_M$, and r_2' are AES encrypted with SyK_{C1} (i.e. $SyK_{C1} = a3.b3.M$), while $ID_{TTP}, ID_{MP(R0)}, P_C2$ (i.e. $P_C2 = b3.M$, $b3 \in Z_n$), and $TS4$ are conveyed additionally. After M_D2, gNB_R compiles a reply to the gNB_S including $CODE_M, n_R2$ within the AES encrypted envelop created by SyK_{C2} (i.e. $SyK_{C2} = a3.c.M$); and $ID_R, ID_{MP(R0)}$, P_C3 (i.e. $P_C3 = c.M$, $c \in Z_n$), $H[n_S3||TS5]$, and $TS5$ contained in the RSA encrypted envelop, while gNB_R signature is also appended in M_C3. The received two $CODE_M$s will be cross-checked at the gNB_S end with the received information. This protocol phase concludes with registering the migration under the TTP entity and establishing mutual authentication among gNB_R and gNB_S. Further, AES-based double-encryptions conducted from SyK_D, SyK_{C1}, and SyK_{C2}, which were computed from the factors disseminated by ECCDH means, ensure the PFS for sensitive credentials of the protocol.

4) Part E and F: MES Verification by MVR to improve the Trust domain

One of the main requirements of this authentication protocol is the validation of the MESs that are intended to be migrated, as specified in G3. This portion of the authentication protocol depicted in Fig. 8 deals with that requirement, where the MVR entity provides the intrinsic validation logic towards the MEC system. In addition, the gNB_R is investigating whether the intended MES has sufficient resources to launch the service in its virtualization environment. Upon receiving the migration verification code $CODE_M$ from the gNB_R, gNB_S forms a message including $ID_{MP(R0)}, ID_{MES}, STATE_{MES}, DATA_{MES}, REQ_{MES}$, ID_{MVR}, P_E1 (i.e. $P_E1 = a5.M$, $a5 \in Z_n$), and $TS1$ within its encrypted envelop and appends the integrity measure $HMAC1 = H[ID_{MP(R0)}||ID_{MES}||STATE_{MES}||DATA_{MES}|| REQ_{MES}||ID_{MVR}||P_E1||TS1]$. Further details on these indexes are specified in Table I. The gNB_R is contacting the relevant MVR server function (i.e. operating under the OSS entity) leveraging the ID_{MVR}. In the first contact message M_F1, gNB_R include $ID_{MES}, STATE_{MES}, ID_S, ID_R, n_R3, TS2$, P_E1, and P_{F1} (i.e. $P_F1 = a6.M$, $a6 \in Z_n$) along with the $gNBR$ signature and $HMAC2$. The MVR after decrypting the received message and validating/ verifying, the ID_{MES} is anchored to seek the ID_S in its database. If the respective ID_{MES} is bound to ID_S, ID_R is temporarily registered in the MVR registry as a MES user. Since the MES claim is legitimate, a code is generated indicating the validation denoted by $CODE_{MES} = H[ID_{MES}||STATE_{MES}||ID_{MVR}||ID_S||ID_R||n1_{RAND}]$. $n1_{RAND}$ is a random nonce similar to the generation of $CODE_M$. The verification status of the MES is indicated by VER_{MES} (i.e., YES or NO). The MVR then composes the reply M_F2 with VER_{MES} and $CODE_{MES}$ within the AES encryption envelop created from SyK_F (i.e. $SyK_F = a6.a7.M$); and include ID_{MVR}, P_F2 (i.e. $P_F2 = a7.M$, $a7 \in Z_n$), $H[n_R3||TS3]$, and $TS3$ within the RSA encrypted envelop along with the MVR signature.

Upon reception, decryption, and validation of M_F2, gNB_R

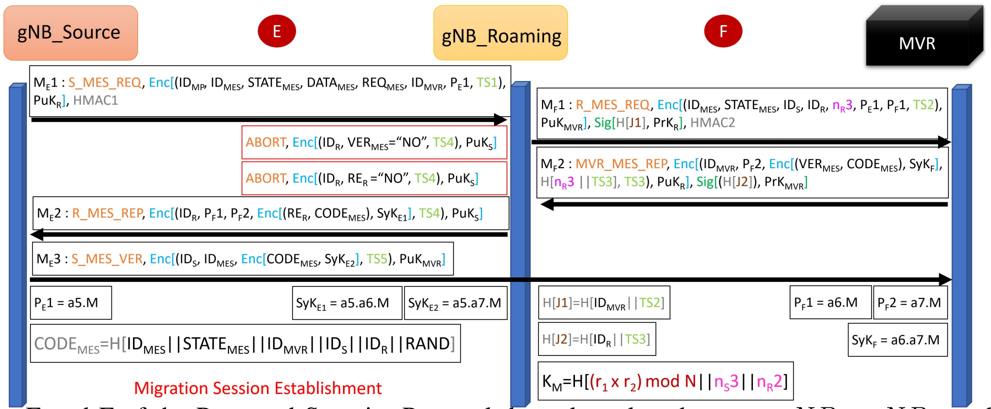

gNB_Source
E
gNB_Roaming
F
MVR
M_E1 : S_MES_REQ, Enc[(ID_{MP}, ID_{MES}, $STATE_{MES}$, $DATA_{MES}$, REQ_{MES}, ID_{MVR}, P_E1, TS1), PuK_R], HMAC1
ABORT, Enc[(ID_R, VER_{MES}="NO", TS4), PuK_S]
ABORT, Enc[(ID_R, RE_R ="NO", TS4), PuK_S]
M_E2 : R_MES_REP, Enc[(ID_R, P_F1, P_F2, Enc[(RE_R, $CODE_{MES}$), SyK_{E1}], TS4), PuK_S]
M_E3 : S_MES_VER, Enc[(ID_S, ID_{MES}, Enc[$CODE_{MES}$, SyK_{E2}], TS5), PuK_{MVR}]
P_E1 = a5.M
SyK_{E1} = a5.a6.M
SyK_{E2} = a5.a7.M
$CODE_{MES}$=H[ID_{MES}||$STATE_{MES}$||ID_{MVR}||ID_S||ID_R||RAND]
Migration Session Establishment
M_F1 : R_MES_REQ, Enc[(ID_{MES}, $STATE_{MES}$, ID_S, ID_R, n_R3, P_E1, P_F1, TS2), PuK_{MVR}], Sig[H[J1], PrK_R], HMAC2
M_F2 : MVR_MES_REP, Enc[(ID_{MVR}, P_F2, Enc[(VER_{MES}, $CODE_{MES}$), SyK_F], H[n_R3 ||TS3], TS3), PuK_R], Sig[(H[J2]), PrK_{MVR}]
H[J1]=H[ID_{MVR}||TS2]
P_F1 = a6.M
P_F2 = a7.M
H[J2]=H[ID_R||TS3]
SyK_F = a6.a7.M
K_M=H[(r_1 x r_2) mod N||n_S3||n_R2]

Fig. 8: Part E and F of the Proposed Security Protocol that takes place between gNB_S, gNB_R and the MVR

stores the $CODE_{MES}$. If the verification value is VER_{MES} = NO, the process will be aborted, and the TTP will be notified. If VER_{MES} = YES, the resource availability check is conducted in accordance with the procedures specified in [19]. If the available resources in the gNB_R are sufficient to host the MES (i.e. REQ_{MES}), then the entity proceeds to the next step and the indicator RE_R = YES. gNB_R then convey the message M_E2 embedding $RE_R, CODE_{MES}$ within a AES encrypted envelop created from SyK_{E1} (i.e. $SyK_{E1} = a5.a6.M$); and appending ID_R, P_F1, P_F2, and $TS4$ inside the RSA encryption. If RE_R = NO, the process is aborted and notified to both gNB_S and TTP. After sending the final message, gNB_R computes the shared migration master key, $K_M = H[(r_1 \times r_2 modN)||n_S3||n_R2]$.

After receiving the RE_R and $CODE_{MES}$, gNB_S computes the same migration master key, $K'_M = H[(r'_1 \times r'_2)||n_S3||n_R2]$. Due to the properties of modular arithmetic, $K_M = K'_M$ and can be used as the master configuration key for the migration session. It can be noted that N is never sent to gNB_S, and TTP is unaware of the values n_S3 and n_R2. Hence, both gNB_S and gNB_R are computing the K_M in different means, while TTP or any other entity is unaware of all the required values. After initiating the migration process, gNB_S notifies the MVR composing M_E3, with $CODE_{MES}$ encrypted with AES key SyK_{E2} (i.e. $SyK_{E2} = a5.a7.M$), and including ID_S, ID_{MES}, and $TS5$ to notify and make the gNB_R registration of ID_{MES} permanent.

5) Part G: Migration Session Establishment

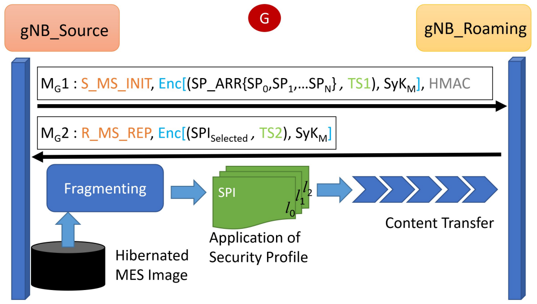

gNB_Source
G
gNB_Roaming
M_G1 : S_MS_INIT, Enc[(SP_ARR{SP_0,SP_1,...SP_N} , TS1), SyK_M], HMAC
M_G2 : R_MS_REP, Enc[($SPI_{Selected}$, TS2), SyK_M]
Fragmenting
SPI
l_0
l_1
l_2
Content Transfer
Application of Security Profile
Hibernated MES Image

Fig. 9: Migration Session Establishment Phase of the Proposed Protocol

As specified in Subsection II-A, and the security goals G5 and G6, the proposed authentication protocol concludes in a situation where the most suited SP can be applicable to the transferring of the migrating content. Thus, this stage can be considered as the migration session establishment phase of the protocol as illustrated in Fig. 9. Since K_M is already determined, an AES-512-based symmetric key is generated utilizing K_M (i.e. K_M is used as the secret key spec), which forms the SyK_M. This SyK_M is intended to be employed in signaling message transfers during the migration session. At the initiation, M_G1 is sent from gNB_S to gNB_R composing all the available SPs that can be bared by gNB_S. The M_G1 is encrypted by SyK_M while a $HMAC$ is appended, as the integrity of this message is vital for the migration session establishment. Upon receiving, gNB_R will select the most suited SP to initiate the migration considering its computational and bandwidth capability. Hence, the SPI of the selected SP is conveyed to the gNB_S encrypted with SyK_M in M_G2. Since the migration g2g tunnel between gNB_S and gNB_R is established, the containerized MES is hibernated into an image that includes its running configuration. Then the fragmented image content is subjected to the relevant cryptographic operations specified under the selected SP. The encrypted content is then migrated to the gNB_R MEC environment to be decrypted, assembled, and configured to launch the MES in the new environment.

IV. Informal Analysis

This section provides the informal or descriptive analysis of the proposed protocol that establishes proofs for the 7 specified propositions.

Proposition 1. The proposed protocols provide Mutual authentication.

Proof. This preposition explains how the proposed protocols (parts A and part C& D) deliver mutual authentication.

- **Proof for Part A:** When gNB_S receives message $((OSS - TD - RP, (SOCK_{TTP}, Cert_{TTP})_{SyK_A}, H[n_s1, TS4], TS4)_{PuK_s})$ from the OSS. gNB_S decrypts this message and compute the $H[n_s1, TS4]^*$ in order to compare $H[n_s1, TS4]^* == H[n_s1, TS4]$ with the received. If it matches then believe that OSS is authentic because n_s1 was sent using the public key of OSS and OSS only knows. On the other hand, when OSS receives $((S_{TD-REQ}, n_s1, X, H[n_{OSS}, TS3])_{PuK_{OSS}})$ from the gNB_S then it decrypts the message to obtain the credentials. After decrypting the message, OSS computes $H[n_{OSS}, TS3]^*$ and compares with the received $H[n_{OSS}, TS3]$ (i.e.,

$H(n_{OSS}, TS3) == H(n^*_{OSS}, TS3))$. If it matches then OSS believes that gNB_S is authentic because gNB_S only knows the n_{OSS}.

- **Proof for Part C&D:** When gNB_S receives the message $((ID_{TTP}, ID_{MP}, P_C2, (CODE_M, r_2^{'})_{SyK_{C1}}, TS4\ )_{PuK_S})$ from the TTP and $((ID_R, ID_{MP}, P_C3, (CODE_M, n_s2)SyK_{C2}, H[n_s2, TS5], TS5)_{PuK_S})$ from the gNB_R. gNB_S decrypts this message and compute the $H[n_s3, TS5]^*$ in order to compare $(H[n_s3, TS5]^* == H[n_s2, TS4], CODE_{M_{gNB_R}} == CODE^*_{M_{TTP}})$. If it matches then believe that gNB_R is authentic because n_s3 was sent using the public key of gNB_R and gNB_R can only know. On the other hand, when gNB_R receives the message $((ID_{MP}, P_{D2}, (CODE_M, r_1, r_2, N_1)_{SyK_D},\ H[n_R1, TS3], TS3)_{PuK_R})$ from the TTP. After decrypting the message, gNB_R computes $H[n_R1, TS3]^*$ with the stored n_R1 in order to compares $H(n_R1, TS3) == H(n_R1^*, TS3), ID_{MP} == ID_{MP})$ than gNB_R believes that gNB_S is authentic because gNB_S only knows the ID_{MP}.

Thus this shows that all three proposed protocols provide Mutual Authentication.

Proposition 2. The proposed protocols provide Confidentiality.

Proof. In the proposed protocols (i.e., part A and part C& D), the identities of the gNB_S, gNB_R, OSS, and TTP are exchanged in the encrypted form instead of plaintext over the insecure channel. For the part A, identities of entities involve in communication gNB_S and OSS are transmitted in encrypted and hashed form $(ID_S, P_A1, TS1)_{PuK_{OSS}}$ and $HMAC1 = H(ID_S, ID_{OSS}, TS1)$. For the part C&D, identities of entities involve in communication gNB_S, gNB_R and TTP are transmitted in encrypted and hashed form $(ID_S, ID_{TTP}, ID_{MP}, P_{C1}, SOCK_{TTP}, n_s3, TS1)_{PuK_R}$. Therefore, we can clearly see that even if an attacker eavesdrops or captures the exchanged messages, he will be unable to get the identity of gNB_S, gNB_R, OSS, TTP because they are exchanged in encrypted from using the public key encryption instead of plaintext. Thus, our proposed protocols provide Confidentiality.

Proposition 3. The proposed protocol provides Perfect Forwards Secrecy.

Proof. The proof of this proposition elaborates that the attacker can not acquire the session key even though he has the private key of communicating entities.

- Part A: If an attacker obtains the private key of gNB_S and OSS (i.e., PrK_S, PrK_{OSS}) then he can not determine the $SOCK_{TTP}$ and $Cert_{TTP}$ because they are encrypted with SyK_A. It is impossible for the attacker to compute the $SyK_A = a1.b1.M$ due to the intractability of ECDL and ECCDH problem [20].
- Part C & D: If an attacker obtain the private key of gNB_S, gNB_R and TTP (i.e., PrK_S, PrK_R, PrK_{TTP}) then he can can not determine the $CODE_M, r_1, r_2, N$, $CODE_M, r_2'$ and $CODE_M, n_R2$ because they are encrypted with $SyK_{C1}, SyK_{C2}, SyK_D$. It is impossible for the attacker to compute the $SyK_{C1} = (a3.b3.M), SyK_{C2} = (a3.C), SyK_D = (a4.d4)$ due to intractability of ECDL and ECCDH problem [20].

Proposition 4. The proposed protocol is resilient against the Replay attack.

Proof. To assure the replay attack protection, we employ the timestamp and nonce in each message exchange through which communicating parties gNB_S, gNB_R, OSS, and TTP could verify the freshness of the exchanged message. For, e.g., in Part A, when OSS receives the $((ID_S, P_A1, TS1)_PuKOSS, Sig[H(J1)]_PrK_s, HMAC1)$ then first it verifies the freshness of the exchanged message by checking the freshness condition $(TS_c - TS_r < \Delta T)$ (i.e., $TS2 - TS1 \leq \Delta$). If it holds, then it accepts the message and decrypts the message. Otherwise, abort the process. The same approach is applied for the rest of the message exchanges for part A, and part C& D to verify their freshness by the receiving end. Hence, the proposed protocols are resilient against Replay attacks.

Proposition 5. The proposed protocols are resilient against the Denial-of-service (DoS) attack.

Proof. In the proposed protocols, we use the timestamp and nonce to examine the freshness of the message (i.e., the message was not sent previously). All the communication entities such as gNB_S, gNB_R, OSS, and TTP of Part A, B, C& D verify the freshness of the message by checking the freshness conditions. If the freshness condition meets, then they again verify the timestamps after verifying the signature of the message. If the timestamp is found correct in both checks, then only the message is accepted. Otherwise, they reject the message and abort. The analysis shows that the attacker can not replay the captured message due to the proper use of timestamps and nonce. Therefore, it is hard for an attacker to launch a denial of service attack. Hence, the proposed protocol is resilient against the DoS attack.

Proposition 6. The proposed protocols are resilient against traceability attacks.

Proof. This attack is impossible due to the usage of random numbers, timestamps, and nonces which are changed after each successful authentication request. The random numbers, timestamps, and nonce utilized in two separate sessions are completely unrelated to one another. Assume the attacker obtains a copy of the messages exchanged between the several sessions. In such a situation, he will not be able to connect communications from one session to messages from another since each session's signature and authentication replies are generated using fresh random numbers, nonce, and timestamps. Consequently, an attacker is unable to link the messages of one session to those of another. As a result, the proposed protocols are resistant to traceability attacks.

Proposition 7. The proposed protocols provide protection from malicious gNB_S and gNB_R.

Proof. Malicious gNB_S or gNB_R is meant by the physical compromise of the entity during its operation. The nature of service delivery in the MEC system, as explicated in [21], is dependent on the virtualization platform that extends from the edge to the core of the network towards its system level. In fact, operations of the MEC system decouple the physical and virtual domains. The governing entity of the MEC edge level, Mobile Edge Platform Manager is constantly communicating with the Mobile Edge Orchestrator and the rest of the system-level entities to decide on the inception, operation, and termination of services; while Virtualization Infrastructure Manager controls the resource allocations and manipulations depending on the service requests. Thus, this autonomous environment is operating without any network administrator at the edge level, while all the governing decisions are conveyed from the system level to the edge through the virtual channels. Hence, access to the MEC edge-level servers or entities cannot be gained by an intruder who has physical access to the system. In fact, an interface is not required for administrative access at the edge, other than a monitoring terminal. Thus, MEC-enabled gNBs are secured against physical threats to the system by design. The reliance on the system level for maintaining the operations, however, induces a possibility for attackers to block the communication channels between the system and edge level, which would isolate the gNB. Therefore, the security of this edge-core channel is a prime requirement for MEC deployments.

Proposition 8. The proposed protocols are resilient against key escrow attacks.

Proof. If the attacker got the public keys then he/she can not determine the session keys due to the use of ECC. For example in Part A, if the attacker has private keys of gNB_S and OSS, then he can only get the P_A1, and P_A2. From P_A1, and P_A2, he can not get the SyK_A due to the hardness of logarithm discrete problem [20]

We do the informal analysis of part A and part C& D since part A is identical to part B and part C&D is identical to part E&F. Therefore, for B and parts C & D, we can follow the same informal analysis approach as part A and part C& D respectively.

V. Formal Analysis

This section presents the formal analysis of the proposed protocol using the GNY logic, ROR logic, Scyther tool [22], and the AVISPA tool.

A. Model Based Verification with Scyther

The Scyther tool was employed to verify the proposed protocol for its resiliency, as it is a well-known automated tool for validating security protocols following a model-based approach. The Security Protocol Description Language (SPDL) is used to specify the protocols in Scyther, where testing protocol segments are specified under spdl files. Since the proposed protocol is described under the segments A, B, C, D, E, and F, the specifications were conducted under 4 SPDL files. A and B segments were specified under two different files, while C & D, and E & F segments were specified with 3 roles.

In the tool, verification, advanced, and graph output parameters are controlling how the validation output is conveyed. Under verification, the maximum number of runs was set to 100 for every parameter sequence in the protocol that was tested. In the first sequence, the matching type was set to *find basic type flaws* while search pruning in advanced parameters was set to *find all attacks*. In the second sequence, *find all type flaws*, and *find all attacks* as the search pruning parameter. For both sequences, the specified claims were verified, and no possible attacks were detected by the tool. The corresponding partial SPDL specification scripts and the verification results of parts A, B, C & D, and E & are depicted in Fig. 10. This overall verification guarantees that the proposed protocol satisfies all timing requirements and nonce's being Alive/ Fresh, weak-agree, Nisynch, and sensitive parameters being secret.

B. Formal verification using the AVISPA

In order to confirm that proposed protocols (i.e., Part A, Part B, and Part C&D) are resilient against the attack, the AVISPA tool [23] is used. AVISPA stands for Automated Validation of Internet Security Protocols and Applications. This tool offers modular and expressive formal language to specify the security features of the authentication protocols. Apart from that, it uses different types of backend server that helps carry out different types of implementation using various automatic analysis techniques ranging from protocol falsification to abstraction-based verification methods for both finite and infinite numbers of sessions. This tool uses the role-based language HLPSL (High-Level Protocols Specification Language) to model the authentication protocol in order to examine its security properties. There are four types of backend servers specified by the tool:1) On- the-fly Model-Checker (OFMC), 2) Constraint Logic-based Attack Searcher (CL-AtSe), 3) SAT-based Model-Checker (SATMC), 4) Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

We use the OFMC and CL-Atse backend servers to verify the proposed protocols similar to [1], [24].

1) Simulation of Part A& Part B using AVISPA tool

We do the simulation of Part A and Part B by modeling the protocol into HLPSL language. Part A protocol is modeled into the two roles $gNBSource$ and OSS, sessions, and environment in order to carry out the simulation. The same was used for Part B, two roles $gNBSource$ and TTP, session, and environment() to carry out the simulation. Fig 11a, Fig 11b, Fig 12a and Fig 12b show that the protocol is safe and secure.

2) Simulation of Part C&D using AVISPA tool

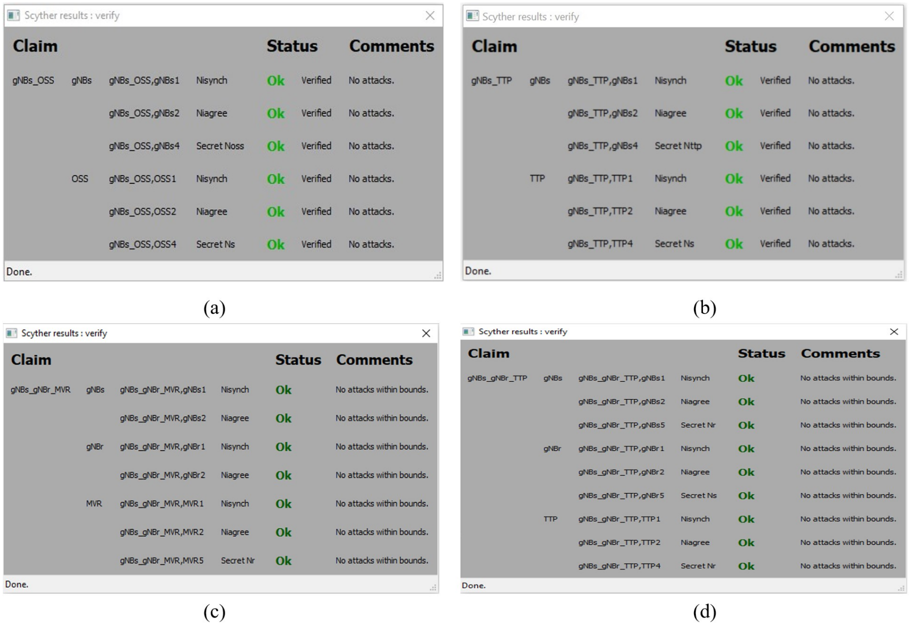
We do the simulation of Part C&D by modeling the protocol into HLPSL language. In Part, C&D, the protocol is modeled into the three roles $gNBSource$, $gNBroamig$, and OSS, sessions, and environment in order to carry out the simulation. Fig 13a and Fig 13b show that the protocol is safe and secure.

C. Formal security analysis using GNY logic

This section discusses the formal verification of the proposed protocol using the GNY logic [25] (i.e., extended version of BAN logic) to prove the robustness of the proposed protocol and securely mutually authenticate each other.

1) GNY Notations

Let A and B be the two entities communicating, and m be the message. We have used the general symbolism described in [25] to specify the GNY logic, while the logical postulates of Being told rules, Possession rules, and Freshness rules were utilized in the proving process.



Scyther results : verify

Claim				Status		Comments
gNBs_OSS	gNBs	gNBs_OSS,gNBs1	Nisynch	Ok	Verified	No attacks.
		gNBs_OSS,gNBs2	Niagree	Ok	Verified	No attacks.
		gNBs_OSS,gNBs4	Secret Noss	Ok	Verified	No attacks.
	OSS	gNBs_OSS,OSS1	Nisynch	Ok	Verified	No attacks.
		gNBs_OSS,OSS2	Niagree	Ok	Verified	No attacks.
		gNBs_OSS,OSS4	Secret Ns	Ok	Verified	No attacks.

Done.

(a)

Scyther results : verify

Claim				Status		Comments
gNBs_TTP	gNBs	gNBs_TTP,gNBs1	Nisynch	Ok	Verified	No attacks.
		gNBs_TTP,gNBs2	Niagree	Ok	Verified	No attacks.
		gNBs_TTP,gNBs4	Secret Nttp	Ok	Verified	No attacks.
	TTP	gNBs_TTP,TTP1	Nisynch	Ok	Verified	No attacks.
		gNBs_TTP,TTP2	Niagree	Ok	Verified	No attacks.
		gNBs_TTP,TTP4	Secret Ns	Ok	Verified	No attacks.

Done.

(b)

Scyther results : verify

Claim				Status	Comments
gNBs_gNBr_MVR	gNBs	gNBs_gNBr_MVR,gNBs1	Nisynch	Ok	No attacks within bounds.
		gNBs_gNBr_MVR,gNBs2	Niagree	Ok	No attacks within bounds.
	gNBr	gNBs_gNBr_MVR,gNBr1	Nisynch	Ok	No attacks within bounds.
		gNBs_gNBr_MVR,gNBr2	Niagree	Ok	No attacks within bounds.
	MVR	gNBs_gNBr_MVR,MVR1	Nisynch	Ok	No attacks within bounds.
		gNBs_gNBr_MVR,MVR2	Niagree	Ok	No attacks within bounds.
		gNBs_gNBr_MVR,MVR5	Secret Nr	Ok	No attacks within bounds.

Done.

(c)

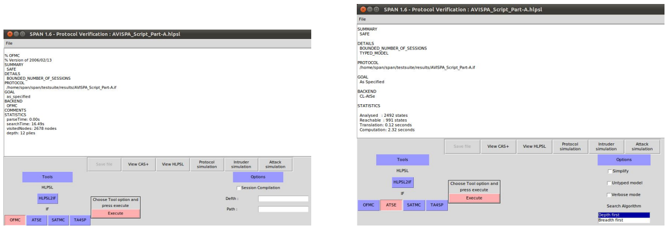
Scyther results : verify

Claim				Status	Comments
gNBs_gNBr_TTP	gNBs	gNBs_gNBr_TTP,gNBs1	Nisynch	Ok	No attacks within bounds.
		gNBs_gNBr_TTP,gNBs2	Niagree	Ok	No attacks within bounds.
		gNBs_gNBr_TTP,gNBs5	Secret Nr	Ok	No attacks within bounds.
	gNBr	gNBs_gNBr_TTP,gNBr1	Nisynch	Ok	No attacks within bounds.
		gNBs_gNBr_TTP,gNBr2	Niagree	Ok	No attacks within bounds.
		gNBs_gNBr_TTP,gNBr5	Secret Ns	Ok	No attacks within bounds.
	TTP	gNBs_gNBr_TTP,TTP1	Nisynch	Ok	No attacks within bounds.
		gNBs_gNBr_TTP,TTP2	Niagree	Ok	No attacks within bounds.
		gNBs_gNBr_TTP,TTP4	Secret Nr	Ok	No attacks within bounds.

Done.

(d)

Fig. 10: Scyther Verification Results of the Protocol: (a) Part A; (b) Part B; (c) Part C & D; (d) Part E & F



SPAN 1.6 - Protocol Verification : AVISPA_Script_Part-A.hlpsl

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/AVISPA_Script_Part-A.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 16.49s
  visitedNodes: 2678 nodes
  depth: 12 plies
```

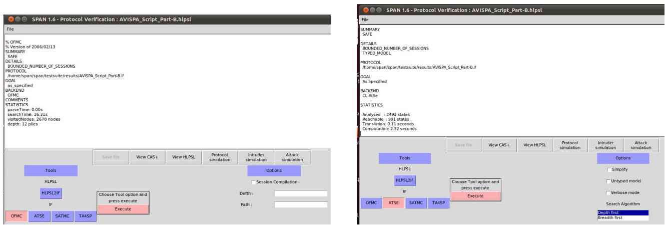
(a)

SPAN 1.6 - Protocol Verification : AVISPA_Script_Part-A.hlpsl

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/AVISPA_Script_Part-A.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2492 states
  Reachable  : 991 states
  Translation: 0.12 seconds
  Computation: 2.32 seconds
```

(b)

Fig. 11: AVISPA outcome for Part A using (a) OFMC backend server (b) CL-Atse backend server.



SPAN 1.6 - Protocol Verification : AVISPA_Script_Part-B.hlpsl

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/AVISPA_Script_Part-B.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 16.31s
  visitedNodes: 2678 nodes
  depth: 12 plies
```

(a)

SPAN 1.6 - Protocol Verification : AVISPA_Script_Part-B.hlpsl

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/AVISPA_Script_Part-B.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2492 states
  Reachable  : 991 states
  Translation: 0.11 seconds
  Computation: 2.32 seconds
```

(b)

Fig. 12: AVISPA outcome for Part B using (a) OFMC backend server (b) CL-Atse backend server

2) *Security verification of Part A of the proposed protocol that takes place between the gNB_S and the OSS using the GNY logic*

Initial assumptions for the protocol

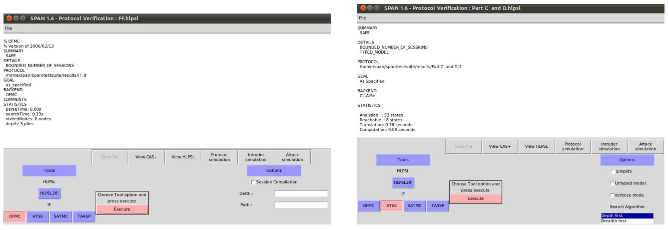
$H_1 : gNB_S \ni PrK_s, H_2 : gNB_S \ni n_s1, H_3 : gNB_S \ni SyK_A$
$H_4 : gNB_S\#(TS2, TS4), H_5 : gNB_S \ni (ID_S, ID_{OSS})$
$H_6 : OSS \ni PrK_{OSS}, H_7 : OSS \ni ID_{OSS}, ID_S$
$H_8 : OSS\#(TS1, TS3), H_9 : OSS \ni n_{OSS}$

The protocol's security goals are as follows:

$OSS \ni H(ID_S, ID_{OSS}, TS_1)$
$gNB_S \ni H(K_{dos}, n_{OSS}, ID_S, TS2, ID_{OSS})$
$gNB_S \ni (SOCK_{TTP}, Cert_{TTP})$

The idealized form of the proposed protocol:

M_{10}: $OSS\triangleleft : *(*ID_S, *TS_1, P_A1)_{PuK_{OSS}}$,
M_{11}: $OSS\triangleleft$: $*H(*ID_S, *ID_{oSS}, *TS_1)$,



SPAN 1.6 - Protocol Verification : FF.hlpsl

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/FF.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.13s
  visitedNodes: 6 nodes
  depth: 3 plies
```

(a)

SPAN 1.6 - Protocol Verification : Part C and D.hlpsl

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/Part C  and D.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 53 states
  Reachable  : 8 states
  Translation: 0.18 seconds
  Computation: 0.00 seconds
```

(b)

Fig. 13: AVISPA outcome for Part C & D using (a) OFMC backend server (b) CL-Atse backend server

M_{20}:$gNB_S\triangleleft$:$*(*k_{dos}, *n_{OSS}, P_A2, *TS_2, R2)_{PuK_S}$
M_{21}:$gNB_S\triangleleft$:$*H(*k_{dos}, *n_{OSS}, *ID_S, *TS2*)$
M_{30}:$OSS\triangleleft$:$*(*n_s1, *X, H[n_{OSS}, TS_3], *TS_3])_{PuK_{OSS}}$,
M_{31}:$OSS\triangleleft$: $H(*(*n_{OSS}, *TS_3)$,
M_{40}:$gNB_S\triangleleft$:$*(*(SOCK_{TTP}, *Cert_{TTP})_{SyK_A}, *H[n_s1, TS_4], *TS_4)_{PuK_s}$
M_{41}:$gNB_S\triangleleft$: $*H(*n_s1, *n_{OSS}, *X)$.

Proof and derivation of security goals:

By applying the BTR_1, BTR_3 and Pr_1 rule on M_{10} based on H_7, we get

$S_1 : S_1 : OSS \ni (ID_S, TS_1, P_A1)$

We apply the BTR_2 and PR_2 rule based on S_1 and H_7, we get

$S_2 : OSS \ni H(ID_S, ID_{OSS}, TS_1)$

Applying the FR rule on S_1 and S_2 based on S_1 and H_8 we get

$S_3 : OSS \mid\equiv \#(ID_S, ID_{OSS}, P_A1)$

By applying the BTR_1, BTR_3 and Pr_1 rule based on H_1, we get

$S_4 : gNB_S \ni (K_{dos}, n_{OSS}, TS2, P_A2)$

We apply the BTR_2 and PR_2 rule based on H_5, S_4

$S_5 : gNB_S \ni H(K_{dos}, n_{OSS}, ID_S, TS2, ID_{OSS})$

Applying the FR rule based on S_4 and H_4, we get

$S_6 : OSS \mid\equiv \#(K_{dos}, n_{OSS}, ID_S, P_A2, ID_{OSS})$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{30} based on H_7, we get

$S_7 : OSS \ni (n_s1, X, H[n_{OSS}, TS3], TS3])$

We apply the BTR_2 and PR_2 rule on M_{31} based on, H_9 and S_7

$S_8 : OSS \ni H(n_{OSS}, TS3)$

Applying the FR rule based on S_8 and H_8, we get

$S_9 : OSS \mid\equiv \#(n_s1, X, H[n_{OSS}])$

By applying the BTR_1, BTR_3 and Pr_1 rule on M_{40} based on H_1, we get

$S_{10} : gNB_S \ni ((SOCK_{TTP}, Cert_{TTP})_{SyK_A}, H[n_s1, TS_4], TS_4)$

We apply the PR_3 rule on S_{10}

$S_{11} : gNB_S \ni (SOCK_{TTP}, Cert_{TTP})_{SyK_A}$

We apply the BTR_4 rule on S_{11} based on $H3$

$S_{12} : gNB_S \ni (SOCK_{TTP}, Cert_{TTP})$

We apply the BTR_2, PR_3 rule on M_{41} based on S_{10}, H_{12}

$S_{13} : gNB_S \ni H(n_s1, n_{OSS}, X)$

Applying the FR rule based on on S_{10} and H_3, we get

$S_{14} : gNB_S \mid\equiv \#(SOCK_{TTP}, Cert_{TTP}, H[n_s1, TS_4]))$

Since Part B of the protocol is identical to Part A, we can prove that Part B follows the same approach.

3) Security verification of Part C and D of the proposed protocol that takes place between the gNB_S, gNB_R and the TTP using the GNY logic

Initial assumptions for the protocol

$H_1 : gNB_S \ni PrK_s, H_2 : gNB_S \ni ID_S, ID_{TTP}, ID_{MP}$

$H_3 : gNB_S\#(TS4, TS4), H_4 : gNB_S \ni (SyK_{C1}, SyK_{C2})$

$H_5 : gNB_R\#(TS1, TS3), H_6 : gNB_R \ni PrK_R$

$H_7 : gNB_R \ni n_R1, H_8 : gNB_R \ni (ID_R, ID_{TTP}, ID_{MP})$

$H_9 : TTP \ni (ID_R, ID_{TTP}, ID_{MP}), H_{10} : TTP\#(TS2)$

$H_{11} : TTP \ni PrK_{TTP}, H_{12} : gNB_S \ni (SyK_D)$

The protocol's security goals are as follows:

$gNB_R \ni TTP_{ID}, MP_{ID(R0)}, SOCK_{TTP_M}, Cert_{TTP}, n_s3, TS1, gNB_R$

$gNB_R \ni (H((n_R1, TS_3))$

$gNB_S \mid\equiv OSS \ni (SOCK_{TTP}, Cert_{TTP})$

$gNB_S \mid\equiv OSS \mid\equiv U_E \overset{SK}{\longleftrightarrow} H_N$

The following steps demonstrate the idealized form of the proposed protocol:

$M_{10} : gNB_R\lhd : *(*ID_S, *ID_{TTP}, *ID_{MP}, *SOCK_{TTP}, *Cert_{TTP}, *n_s3, P_C1, *TS1)_{PuK_R},$

M_{11}: $gNB_R\lhd$: $H(*ID_S, *ID_{TTP}, ID_{MP}, *SOCK_{TTP}, Cert_{TTP} * n_s3, ID_R, *TS_1),$

$M_{20} : TTP\lhd$: $*(*ID_R, *ID_{TTP}, ID_{MP}, n_R1, P_C1, P_C2, TS2)_{PuK_{TTP}}$

M_{21}:$TTP\lhd$: $*H(*ID_R, *ID_{TTP}, *ID_{MP}, n_R1, TS2))$

$M_{30} : gNB_R\lhd : *(ID_{MP}, P_D2, (CODE_M, r_1, r_2, N)_{SyK_D}, H[n_R1, TS3], TS3)_{PuK_R},$

$M_{31} : gNB_R\lhd : H(*(*n_R1, *TS_3))),$

$M_{40} : gNB_S\lhd$:$*(*ID_{TTP}, *ID_{MP}, P_C2, (CODE_M, r'_2)_{SyK_{C1}}$,$*TS4)_{PuK_s},$

$M_{50} : gNB_S\lhd$: $*(*ID_R, *P_C3, *(CODE_M, n_R2)_{SyK_{C2}}, *n_R2$,$H[n_R3, TS5], TS5)_{PuK_S},$

$M_{51} : gNB_S\lhd$: $*H(*n_S3, TS5),$

Proof and derivation of security goals:

By applying the BTR_1, BTR_3 and PR_1 rule on M_{10} based on H_6, we get

$S_1 : gNB_R \ni (ID_S, ID_{TTP}, ID_{MP}, SOCK_{TTP}, Cert_{TTP}, n_s3, P_C1, TS1)$

BTR_2 and PR_2 rules on M_{10} based on S_1 and H_8 was applied.

$S_2 : gNB_R \ni H(ID_S, ID_{TTP}, ID_{MP}, SOCK_{TTP}, Cert_{TTP}, n_s3, ID_R, TS_1)$

Applying the FR rule on S_1, S_2 based on H_5, we get

$S_3 : gNB_R \mid\equiv \#(ID_S, ID_{TTP}, ID_{MP}, SOCK_{TTP},) Cert_{TTP}, n_s3, P_C1$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{20} based on H_{11} , we get

$S_4 : TTP \ni (ID_R, ID_{TTP}, ID_{MP}, n_R1, P_C1, P_D1, TS2)$

We apply the BTR_2 and PR_2 rule on M_{21} based on S_4 and H_9

$S_5 : TTP \ni H(ID_R, ID_{TTP}, ID_{MP}, n_R1, TS2)$

Applying the FR rule on S_4 based on H_{10}, we get

$S_6 : TTP \mid\equiv \#(H(ID_R, ID_{TTP}, ID_{MP}, n_R1, P_C1, P_D1,))$

By applying the BTR_1 and BTR_3 and PR_1 rule on M_{30} based on H_6, we get

$S_7 : gNB_R \ni (ID_{MP}, P_D2, (CODE_M, r_1, r_2, N)_{SyK_D}, H[n_R1, TS3], TS3)$

We apply the BTR_2 and PR_2 rule on M_{31} based on, H_8 and S_6

$S_8 : gNB_R \ni (H(n_R1, TS_3))$

We apply the PR_3 rule on S_7

$S_9 : gNB_R \ni (CODE_M, r_1, r_2, N)_{SyK_D}$

We apply the BTR_4 rule on S_9 based on H_{12}

$S_{10} : gNB_R \ni (CODE_M, r_1, r_2, N)$

Applying the FR rule based on S_5 and H_5 we get

$S_{11} : gNB_R \mid\equiv \#(ID_{MP}, P_D2, (CODE_M, r_1, r_2, N),) H[n_R1, TS3], TS3$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{40} based on H_1, we get

$S_{12} : gNB_S \ni ID_{TTP}, ID_{MP}, P_c2, (CODE_M, r'_2)_{SyK_{C1}}, TS_4$

We apply the PR_3 rule on S_{12}

$S_{13} : gNB_S \ni (CODE_M, r'_2)_{SyK_{C1}}$

We apply the BTR_4 rule on S_{13} based on H_4

$S_{14} : gNB_S \ni (CODE_M, r'_2)$

Applying the FR rule based on S_{10} and H_3, we get

$S_{15} : OSS \mid\equiv \#(ID_{TTP}, *ID_{MP}, P_C2, (CODE_M, r'_2)), *TS_4$

By applying the BTR_1, BTR_3 and PR_1 rule on M_{50} based on H_1, we get

$S_{16} : gNB_S \ni ID_R, P_C3, (CODE_M, n_R2)_{SyK_{C2}}, n_R2, H[n_R3, TS5], TS5$

We apply the BTR_2 and PR_3 rule on M_{51} based on S_{16}

$S_{17} : gNB_S \ni H(n_S3, TS5)$

We apply the PR_3 rule on S_{15}

$S_{18} : gNB_S \ni (CODE_M, n_R2)_{SyK_{C2}}$

We apply the BTR_4 rule on S_{13} based on H_4

$S_{19} : gNB_S \ni (CODE_M, n_R2)$

Applying the FR rule based on S_{16} and H_3, we get

$S_{20} : OSS \mid\equiv \#(ID_R, M2, (CODE_M, n_R2), n_R2,) H[n_R3, TS5]$

Since Parts E & F of the protocol is identical to Parts C & D, we can prove Parts E & F in the same manner.

D. Formal security analysis using ROR Logic

We use Real-or-Random (ROR) logic proposed by Abdalla et al. [26] in order to verify the session key security. ROR is considered a more suitable case than the original model proposed by Bellare et al. [27] to formally simulate real attacks on the authentication scheme for 5G gNodeBs in Service Migration Scenarios of MEC. Some of the basic concepts following their work are omitted in this paper, such as participants, long-term keys, freshness, etc. In the ROR model, the security model is defined by a game between two probabilistic polynomial-time Turing machines, namely a challenger (CH) and an adversary (A_d). It is assumed that CH owes a real system that has already applied our proposed scheme P_{MEC} to it. In order to evaluate $P^{'}_{MEC}$ security, CH intended to invite A_d to launch a real attack on P_{MEC}, but CH worried about A_d would learn enormous useful information about the real system. So CH developed an oracle system and designed a game to play with A_d. After initialization, the oracle flips an unbiased coin c (c=0, 1), and the goal of A_d is to guess the value of CH. To increase the chance of winning this game, A_d is provided a series of queries to ask the oracle. There are three communicating parties, such as (gNB_S, gNB_R, TTP) involved in the authentication of 5G gNodeBs in Service Migration Scenarios of MEC. Let instances of f, g and h of gNB_S , gNB_R and TTP are denoted by gNB_S^f and gNB_R^g and TTP^h respectively. In the ROR model, it is assumed that A_d can perform several activities such as deleting, inserting, and editing the captured exchanged message. A_d can perform these activities by executing the queries defined in the ROR model. The description of these queries is as follows.

- *Execute (gNB_S^f , gNB_R^g, TTP^h): A_d* executes this query to intercept the exchanged message between the gNB_S^f , gNB_R^g, TTP^h.
- *Reveal* $((\Pi^l)$*: A_d* executes this query to obtain the current session key between the gNB_S^f, gNB_R^g, TTP^h.
- *Send* $((\Pi^l, mess)$*: A_d* executes this query to forge the captured message (i.e., it modified the captured message and then replay this message to the gNB_S^f, gNB_R^g, TTP^h.) so that he receives the response of the forged message.
- *Test* (E^h)*:* This query is used to examine the session key security of the communicating entities D^f, and AS^g). To examine the session key security, a coin is tossed before starting the game. Based on the tossed outcome, A takes a decision (i.e., c=0, the communicating party returns the random number or c=1, then communicating party returns the session key. Otherwise, a null value is returned.)

Theorem 1: If A_d tries to crack the session key (SK) in polynomial time. Then $Adv_{A_d} \leq \frac{H_Q^2}{2^U} + 2A_{A_d}^{ECDDH}$

Where H_Q, A^{ECDDH}, U stands for a number of $Hash$ queries, hardness of the discrete logarithm problem, and hash function output value, respectively.

Proof: Since Part A, Part B, Part C&D, and Part E&F use the combination of ECC and RSA to protect the exchange message confidentiality and integrity. Here, we do the proof for Part C&D since all the parts employ the same mechanism. The proof of the protocol is shown using the three games known as G_1, G_2, G_3. An event $S_{A_dG_1}$ is defined as the success probability of the A_d to guess the session key or to win the game.

Game (G_1)**:** By executing this game, A_d tries to get the actual value of c at the start of the game before Oracle initializes the procedure of the game. So, we get

$$Adv_{A_d} = |2P[S_{A_dG_0}] - 1] \tag{1}$$

Game (G_2)**:** A_d executes the *Execute* query to win the game by intercepting the exchange message $\{M_C1, M_C2, M_C3, M_D1, M_D2\}$ between the $gNB_{SOURCE}, gNB_{Roaming}$, and TTP. When A_d obtains the exchange message, then it tries to get the correct secret value by guessing the value of c based on the execution of *Test* query. Since we use the random numbers $\{a_3, b_3, c, a_4, a_5\}$ derived from the elliptic curve. They are random (i.e., used only once in the protocol) and exchanged in encrypted form between the communicating entities. So, finding any clue for the random numbers is tough so that A_d will get the session key. Therefore, A_d will lose the game, and the winning possibility of G_1 will be similar to the G_2. Hence we can get

$$P[S_{A_dG_2}] = P[S_{A_dG_1}] \tag{2}$$

Game(G_3)**:** Since during the G_2 execution attacker was unable to get the right session key, but he has the intercepted message. In this game, A_d tries different ways to get the session key by modeling this game as an active attack by executing the *Send* query. We use the ECC and RSA that protects the exchange message and will not let the A_d derive any secrets of the protocol, especially to determine the random number from these computed parameters $\{P_c1, P_C2, P_C3P_D1, P_D2\}$ due to the hardness of the discrete logarithm problem [20]. This shows that A_d will not be able to determine any insight to derive the session key and will not be any collision while the *Hash* will run. So, the winning probability of G_3 will be similar to the previous game. Hence, This can be obtained by adopting the birthday paradox

$$P[S_{A_dG_2}] - P[S_{A_dG_3}] \leq \frac{H_Q^2}{2^{U+1}} + +2A_{A_d}^{ECDDH} \tag{3}$$

Now, all the game has been executed by the A_d in order to predict the correct value of C, So we can

$$P[S_{A_dG_3}] = \frac{1}{2} \tag{4}$$

from Eq(1) (2), and (4), we can obtain

$$\begin{aligned} Adv_{A_d} &= |2P[S_{A_dG_1}] - 1| \\ \frac{1}{2}Adv_{A_d} &= |P[S_{A_dGame_1}] - \frac{1}{2}| \\ &= P[S_{A_dG_2}] - P[S_{A_dG_3}] \end{aligned} \tag{5}$$

We obtain the following outcome from Eq (3) and (5).

$$\begin{aligned} \frac{1}{2}Adv_{A_d} &\leq \frac{H_Q^2}{2^{U+1}} + 2A_{A_d}^{ECDDH} \\ Adv_{A_d} &\leq \frac{H_Q^2}{2^U} + 2A_{A_d}^{ECDDH} \end{aligned} \tag{6}$$

The outcome after executing the game indicates that A_d can not obtain the session key in a polynomial amount of time.

VI. Validation and Performance Comparison

This section addresses the proposed protocols' performance measurements in terms of security verification, computational, communication, and energy consumption and compares them to their equivalent counterparts in the literature.

A. Security features verification

This section presents the security features verification as mentioned in II-C. We conduct the informal as well as the formal (Syther and GNY logic) security verification to show the proposed protocol's robustness against the identified attacks. The research carried out in [28], [29] shows that [1] is vulnerable to various attacks such as confidentiality and MiTM. [1], [30] shows that [31] does not offer the perfect forward secrecy, no formal verification, and confidentiality. The security analysis depicted in [1], [32], [33] confirms that [24] does not offer confidentiality and is also vulnerable to traceability attacks. Further, [34] does not offer confidentiality, and is vulnerable to DoS threats according to [1], [35], [36].

The comparison outcome shown in Table II demonstrates that the proposed protocol has the capability to offer all the identified security features while [1], [24], [31], [34] fails in some sort of security features verification. The protocol in [37] however, meets all the security features covered in our protocol. The fact that we employ a random number, timestamp, and nonce that changes after each successful authentication is the major rationale for providing all of the security features.

TABLE II: Comparing security features of existing protocols/L_1-Mutual Authentication; L_2-confidentiality; L_3-PFS; L_4-Replay protection; L_5-DoS protection; L_6-Traceability protection; L_7-Protection from malicious gNBs; L_8-Formal analysis

Protocols	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
[1]	✓	×	✓	✓	✓	✓	✓	✓
[31]	✓	×	×	✓	✓	✓	✓	×
[24]	✓	×	✓	✓	✓	×	✓	✓
[37]	✓	✓	✓	✓	✓	✓	✓	✓
[34]	✓	×	✓	✓	×	×	×	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓

B. Computational Cost

This section evaluates the number of cryptographic operations used in the proposed protocol and its counterparts. We consider the cost of cryptographic operation as mentioned in [1], using two cores on Intel i7-6600U CPU @ 2.60 GHz as gNBs and the OpenSSL with two cores on Intel i4-2500 @3.30 GHz as OSS shown in Table III. The notation T_{RSA}, T_H, T_P, T_E, T_{SM} and T_{MSM} stands for RSA signature, Secure Hash Algorithm ($SHA2$) function, pairing operation, modular exponential, elliptic curve scalar multiplication and multi elliptic curve scalar multiplication, respectively. Table IV contains the computational cost for all the proposed protocols. We also compare the proposed protocol (Part A) with [1] [31] [24] [37] [34] in terms of computational cost, which is displayed in Table VI, demonstrating that the proposed protocol is the least expensive. The proposed protocol is the least costly since it is based on the combination of T_{RSA} and T_{SM} which is less costly compared to [1] [31] [24] [37] [34] that uses the T_E, T_{SM} and T_{MSM} as shown in Table III obtained through experimental analysis as [1]. As stated earlier, current literature lacks any authentication mechanisms to contrast against phase C& D and phase E& F.

TABLE III: Computation cost for cryptographic operations

Protocols	T_{SM}	T_H	T_{MSM}	T_{RSA}	T_P	T_E
gNB	0.2025	0.0032	0.2532	0.127	2.87	0.225
OSS	0.03	0.00029	0.0375	0.019	0.7616	0.0337

C. Communication Cost

This section evaluates the number of bits transmitted in the channel for the proposed protocol and its counterparts. To compute the number of bits transmitted, we use the same bit size as used in [1]. The notation M_{RSA}, M_{ID}, M_{TS}, M_{SM} and M_H stands for the packet is encrypted with RSA size of 2048 bit, the identity of 32 bits, timestamp of 32 bit, elliptic curve multiplication of 224 bits and hashed by 256 bit, respectively. Table V shows the communication cost for the proposed protocols. We also compare the proposed protocol (Part A) with [1] [31] [24] [37] [34] in terms of communication cost, which is illustrated in Table VI, indicating that the proposed protocol takes the high-cost [1] [31] [24] [37] and is the less compared to [34]. Although our proposed protocol takes communication cost high cost compared to [1] [31] [24] [37], but offers the better security, less computational cost, and additional parts such as part C&D and part E& F.

D. Storage Cost

This section determines the memory required for the gNB_S and gNB_R. We take the size of cryptographic operations as stated in [1], with RSA being 2048 bits, identity being 32 bits, and hashed output being 256 bits. Table V shows the storage cost required for the proposed protocols.

E. Energy Consumption

This section determines the energy consumption for the cryptographic operations utilized in the various part of the proposed protocol. We have used the same energy consumption as [38] to measure energy consumption. To compute the energy consumption, experiments are performed using a "Strong ARM" CPU running at 133 MHz doing various tasks is described as the energy required for transmitting a bit, AES symmetric enc/dec, Hashed output, enc/dec RSA are 0.00066 mj, 0.00217 mj, 0.000108 mj, 15.3 mj, respectively. Table IV shows the energy consumption required for the proposed protocols.

VII. Implementation

This section explicates the pragmatic feasibility of the protocol while justifying the necessity for the embedded security measures. The specifications of the implemented prototype MEC environment are further elucidated while the conducted experiments are described within the valuation context.

A. Developed Experimental MEC Environment

This research attempt was initiated having a large focus on the SMSF specified in section II-A. Thus, a prototype testbed of the MEC service migration environment was developed and emulated for evaluating the feasibility of the proposed protocol in a practical deployment scenario. Fig. 14 represents the formation of the entities, gNB_S, gNB_R, TTP, and the OSS. A high-performance server (i.e. Processor: Intel Xeon 2.2 GHz 24 CPU, RAM: 98 GB, OS: Ubuntu 16.04 LTS) was launched as the MEC system level, that embeds both the OSS and 5G Core entities operating as VMs. The 5G Core was launched

TABLE IV: Computational cost and energy consumption of the proposed protocols

Protocol Segment	Computational cost	Total time (ms)	Energy required	Energy consumption (mj)
Part A	$6T_{RSA}+6T_H+4T_{SM}+1T_{AES}$	0.91	(12800×0.00066+6×0.000108+6×15.1+4×8.8+1×.00208)	134.24
Part B	$6T_{RSA}+6T_H+4T_{SM}+1T_{AES}$	0.91	(12800×0.00066+6×0.000108+6×15.1+4×8.8+1×.00208)	134.24
Part C& D	$10T_{RSA}+10T_H+9T_{SM}+3T_{AES}$	2.14	(18994×0.00066+10×0.000108+10×15.1+9×8.8+3×.00208)	242.74
Part E& F	$7T_{RSA}+8T_H+6T_{SM}+3T_{AES}$	1.97	(14848×0.00066+8×0.000108+7×15.1+6×8.8+3×.00208)	168.31

TABLE V: Communication and storage costs of the proposed protocols

Protocol Segment	Message exchanges	Communication cost (bits)	Stored credentials	Storage cost (bits)
Part A	$((2M_{RSA}+M_H),(2M_{RSA}+M_H),(M_{RSA}),(M_{RSA}))$	12800	$(ID_{OSS},ID_S,PrK_{OSS},PrK_S)$	4160
Part B	$((2M_{RSA}+M_H),(2M_{RSA}+M_H),(M_{RSA}),(M_{RSA}))$	12800	$(ID_{TTP},ID_S,PrK_{TTP},PrK_S)$	4160
Part C& D	$((2M_{RSA}+M_H),(2M_{RSA}+M_H),(2M_{RSA}),(M_{RSA}),(2M_{RSA}))$	18994	$(2ID_{TTP},2ID_{MP},2ID_s,SOCK_{TTP},Cert_{TTP},,2ID_R,PrK_R,PrK_S,PrK_{TTP},r_1,r_2)$	10560
Part E& F	$(M_{RSA}+M_H),(2M_{RSA}+M_H),(2M_{RSA}),(M_{RSA}),(M_{RSA}))$	14848	$(2ID_{TTP},2ID_{MP},2ID_S,SOCK_{TTP},Cert_{TTP},2ID_R,PrK_R,PrK_S,r_2',PrK_{TTP},r_1,r_2)$	10592

TABLE VI: Comparison of computational and communication costs of Part A segment with its counterparts

Protocols	UE side	Total time (ms)	Message exchange	Total cost (bits)
[1]	$6T_{SM}+2T_{MSM}$	0.9982	$((3M_{SM}+M_{ID},M_{TS}),(3M_{SM}+M_{ID}+M_{TS}+M_H),(M_H))$	1792
[31]	$8T_E+2T_{RSA}$	1.18	$((2M_{RSA}+M_{ID}+M_{SM}+M_{TS}),(2M_{RSA}+M_{ID}+M_{SM}+M_{TS}+M_H),(M_{TS}))$	9088
[24]	$6T_{SM}+4T_{MSM}$	1.3		
[37]	$3T_E+5T_{SM}+3T_P+T_{MSM}$	3.46	$((M_{RSA}+M_{ID}+3M_{SM}+M_{TS}),(M_{RSA}+M_{ID}+M_{SM}+M_{TS}+M_H),(M_{TS}))$	5408
[34]	$3T_P+6T_{SM}+7T_{MSM}$	10.2	$((M_{ID}+M_{SM}+M_{TS}),(M_{ID}+3M_{SM}+M_{TS},(7M_{RSA}+M_{TS}))$	15692
Ours	$6T_{RSA}+6T_H+4T_{SM}+1T_{AES}$	0.91	$((2M_{RSA}+M_H),(2M_{RSA}+M_H),(M_{RSA}),(M_{RSA}))$	12800

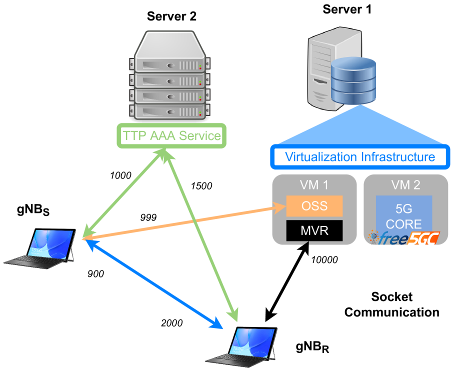
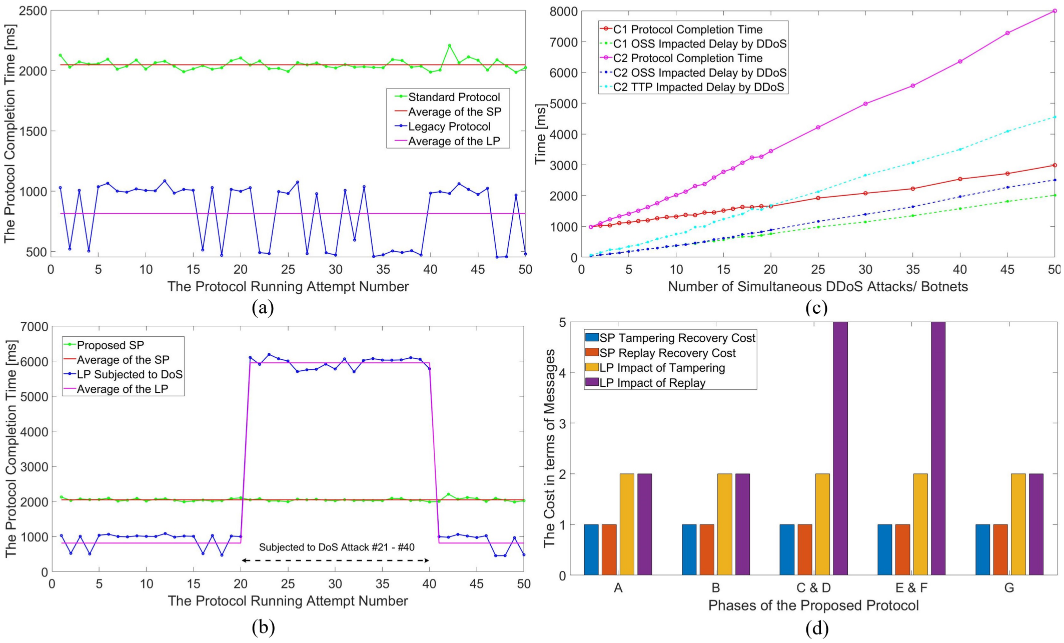


Fig. 14: Prototype Implementation of the Proposed Protocol

leveraging the free5GC (i.e. https://www.free5gc.org/) tool. In addition, the TTP or AAA server function was launched at a separate server bearing moderate specifications of Processor: Intel Xeon 2.4 GHz 4 CPU, RAM: 8 GB, OS: Windows Server 2016 64bit. The MEC virtualization platforms of the two emulating $gNBs$ were maintained in two laptops, due to the requirement for them to be mobile and dynamic to conduct the current and future emulations. The connections between the entities or interfacing were established via the socket-based Inter-Process Communication (IPC) approach. The protocol steps were specified using a Java base. For the cryptographic operations, RSA-4096 bit, AES-256, SHA-512, clock-skew 50 ms, and K_{DoS} as 4 parameters were used. The complexity K_{DoS} exceeding 4 would consume more than 500 ms for solving, which is not ideal for the context of the protocol. The P-256 ECDH construct described in RFC5903 was deployed for relevant ECC-based PFS mechanisms. The developed prototype MEC setup converged the protocol to an average completion time of 2047 ms, which covered the phases from A to G. A comparatively higher value is exhibited due to the involvement of many identity and legitimacy verification entities, in addition to the adoption of DoS mitigation and PFS ensuring methods, that incur formidable delays to the protocol.

B. Conducted Emulation-based Experiments

In order to implicate a scenario where the proposed security measures were not applicable, we have removed the main security measures from the developed protocol and denoted it as the Legacy Protocol (LP). The LP, therefore, withdrew 2 messages from each A and B phases, lingering only the request and reply messages with their inherent Asymmetric security measures. All the timing, hashing, nonce, and DoS measures were detached from the message flows. Since we have already conducted a cost-wise comparison with existing protocols in Table VI, the impact of integrating the proposed security measures into the protocol was evaluated to justify their security-heavy nature. Fig. 15-(a) presents a comparison between the protocol Completion Times (CTs) of the Standard Protocol (SP) and the LP. With the reduced security overhead, LP converges to an average CT of 814 ms. On the contrary, Fig. 15-(b) divulge the impact of a DoS attack on the LP from attempt 21 to 40, where the CTs are clearly accumulated beyond 6000 ms. Fig. 15-(c) further elaborates on the impact of a DDoS threat, where DDoS bots were emulated towards the server interfaces of the protocol and assumed that each request was handled sequentially while running the setup. Since there are two server interfaces, C1 represents the case where only the OSS is subjected to the DDoS threat, while C2 represents the scenario where both OSS and TTP entities are under the influence of the stated threat. The emulation deduced that DoS measures are essential to mitigate the wasted timing on the system in addition to the exposed idling server interfaces. Fig. 15-(d) depicts a cost-wise perspective of the proposed SP and the LP in case of either tampering or Replay attempts were directed toward the different phases of the protocols. The lack



(a)

(c)

(b)

(d)

Fig. 15: The Results of the Emulations Conducted in the Developed Testbed Environment

of integrity or freshness measures of LP proves costly.

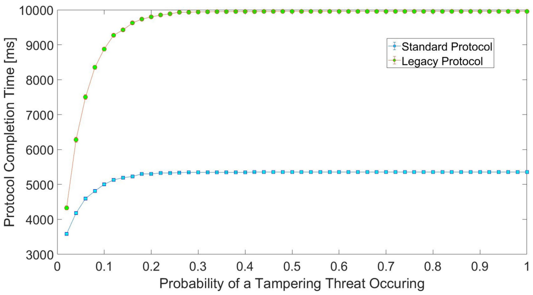


Fig. 16: The Impact of Tampering to the Protocol Completion Time, based on a Probabilistic Approach

C. Simulation to Evaluate the Impact of Tampering

Further, we have simulated the behavior of the protocol in case of a tampering threat, and that is depicted in Fig. 16. In this simulation, the protocol completion time was computed considering the delays that ensued for re-transmissions with the perpetrated tampering attempts. This simulation compares the behavior of the two protocols SP and LP, in the context of the probability that a tampering threat is occurring. It is clearly observable that both SP and LP are converging to their maximum re-transmission delays approximately at 0.2 and 0.3 probabilistic instances. LP protocol is undoubtedly exceeding the delays as it lacks the means to detect the ensued tampering attempts. Thus, security mechanisms integrated into the proposed protocol are vital for safeguarding the entities and content involved with service migration processes of MEC environments.

VIII. Conclusion

In this work, our focus was to address the security issues associated with the edge-to-edge service migration processes of MEC deployments, that take place between two gNBs. As the MEC is deployed in a dynamic environment that feeds the 5G-related services, managing the security in long-lasting migration processes is an apparent conundrum. Our primary focus was to maximize the security measures along with mitigating DoS attempts and address the legitimacy issues inherent in novel virtual MEC-based deployments. The proposed protocol ensures mutual authentication among engaged entities through signature and nonce-based verification methods. In addition, a method for verifying the legitimacy of the operating MESs was proposed. The proposed protocol ensures the formation of the migration master key at the conclusion, while it can be utilized to safely configure the respective security profiles. The conducted formal analysis with Scyther and AVISPA tools along with GNY and ROR logics and the informal analysis specifications proved the correctness of our protocol. The comparable values of the efficiency measurements and the timing measurements from the testbed implementation prove the feasibility of the protocol in a server environment suited for a MEC deployment. The future focus of this research tends to the development of the service migration security framework introduced in Subsection II-A.

Acknowledgement

This work is partly supported by the Science Foundation Ireland under CONNECT Research Center (13 RC/2077_P2).

References

[1] 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.

[2] 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.

[3] X. Yan, M. Ma, and R. Su, "Efficient group handover authentication for secure 5g-based communications in platoons," IEEE Transactions on Intelligent Transportation Systems, 2022.

[4] 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.

[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, 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.

[7] ETSI, "Mobile Edge Computing (MEC) Framework and Reference Architecture," ETSI White Paper #3, 2016, last accessed 02 June 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf

[8] A. Nadembega, A. S. Hafid, and R. Brisebois, "Mobility prediction model-based service migration procedure for follow me cloud to support qos and qoe," in 2016 IEEE International Conference on Communications (ICC). IEEE, 2016, pp. 1–6.

[9] A. Nadembega, A. Hafid, and T. Taleb, "A destination and mobility path prediction scheme for mobile networks," IEEE transactions on vehicular technology, vol. 64, no. 6, pp. 2577–2590, 2014.

[10] 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.

[11] A. K. Yadav, M. Misra, P. K. Pandey, A. Braeken, and M. Liyange, "An improved and provably secure symmetric-key based 5g-aka protocol," Computer Networks, vol. 218, p. 109400, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622004340

[12] A. K. Yadav, M. Misra, P. K. Pandey, and M. Liyanage, "An eap-based mutual authentication protocol for wlan connected iot devices," IEEE Transactions on Industrial Informatics, pp. 1–12, 2022.

[13] K. Sowjanya, M. Dasgupta, and S. Ray, "A lightweight key management scheme for key-escrow-free ecc-based cp-abe for iot healthcare systems," Journal of Systems Architecture, vol. 117, p. 102108, 2021.

[14] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on multi-access edge computing security and privacy," IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1078–1124, 2021.

[15] 3GPP, "System architecture for the 5g system (5gs): Ts 23.501," 2021.

[16] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin et al., "Mec in 5g networks," ETSI white paper, vol. 28, no. 2018, pp. 1–28, 2018.

[17] ETSI, "Mobile-Edge Computing–Introductory Technical White Paper," ETSI White Paper #1, 2014, last accessed 16 May 2019. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf

[18] A. Braeken, P. Porambage, A. Puvaneswaran, and M. Liyanage, "Essmar: Edge supportive secure mobile augmented reality architecture for healthcare," in 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech). IEEE, 2020, pp. 1–7.

[19] 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.

[20] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A new mutual authentication and key agreement protocol for mobile clientserver environment," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1275–1286, 2021.

[21] 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.

[22] C. J. F. Cremers, Scyther: Semantics and verification of security protocols. Eindhoven university of Technology Eindhoven, Netherlands, 2006.

[23] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani et al., "The avispa tool for the automated validation of internet security protocols and applications," in International conference on computer aided verification. Springer, 2005, pp. 281–285.

[24] 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.

[25] 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.

[26] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based Authenticated Key Exchange in the Three-party Setting," in International Workshop on Public Key Cryptography. Springer, 2005, pp. 65–84.

[27] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in International conference on the theory and applications of cryptographic techniques. Springer, 2000, pp. 139–155.

[28] P. K. Roy, P. Sahu, and A. Bhattacharya, "Fasthand: A fast handover authentication protocol for densely deployed small-cell networks," Journal of Network and Computer Applications, p. 103435, 2022.

[29] S. Gupta, B. L. Parne, N. S. Chaudhari, and S. Saxena, "Seai: Secrecy and efficiency aware inter-gnb handover authentication and key agreement protocol in 5g communication network," Wireless Personal Communications, vol. 122, no. 4, pp. 2925–2962, 2022.

[30] M. Ramadan, F. Li, C. Xu, A. Mohamed, H. Abdalla, and A. A. Ali, "User-to-user mutual authentication and key agreement scheme for lte cellular system." Int. J. Netw. Secur., vol. 18, no. 4, pp. 769–781, 2016.

[31] J. Choi and S. Jung, "A handover authentication using credentials based on chameleon hashing," IEEE communications letters, vol. 14, no. 1, pp. 54–56, 2009.

[32] Z. Zhou, H. Zhang, and Z. Sun, "An improved privacy-aware handoff authentication protocol for vanets," Wireless personal communications, vol. 97, no. 3, pp. 3601–3618, 2017.

[33] S. Gupta, B. L. Parne, and N. S. Chaudhari, "A proxy signature based efficient and robust handover aka protocol for lte/lte-a networks," Wireless Personal Communications, vol. 103, no. 3, pp. 2317–2352, 2018.

[34] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," IEEE Transactions on Wireless Communications, vol. 9, no. 1, pp. 168–174, 2010.

[35] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," IEEE Transactions on Wireless Communications, vol. 11, no. 1, pp. 48–53, 2011.

[36] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," IEEE transactions on wireless communications, vol. 10, no. 2, pp. 431–436, 2010.

[37] 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.

[38] Z. Xu, X. Li, J. Xu, W. Liang, and K.-K. R. Choo, "A secure and computationally efficient authentication and key agreement scheme for internet of vehicles," Computers & Electrical Engineering, vol. 95, p. 107409, 2021.

Pasika Ranaweera (Member, IEEE) is currently Post Doctoral Researcher at the School of Computer Science, University College Dublin, Ireland. More Info: https://people.ucd.ie/pasika.ranaweera

Awaneesh Kumar Yadav (Student Member, IEEE) is currently working toward a PhD degree at the Indian Institute of Technology, Roorkee, Uttarakhand, India (akumaryadav@cs.iitr.ac.in).

Madhusanka Liyanage (Seiner Member, IEEE) is an Assistant Professor/Ad Astra Fellow at the School of Computer Science, University College Dublin, Ireland. More Info: :http://madhusanka.com.

Anca Delia Jurcut is an Assistant Professor at the UCD School of Computer Science, Ireland. More Info:https://people.ucd.ie/anca.jurcut