Design of Secure Authentication Handover Protocol for Innovative Mobile Multimedia Services in 5G MEC Environments

Jiyoon Kim[1], Dong-Guk Han[2], Ilsun You[2*]

[1] School of Computer Science, Gyeongsang National University, Republic of Korea
[2] Department of Financial Information Security, Kookmin University, Republic of Korea
jykim92@gnu.ac.kr, christa@kookmin.ac.kr, ilsunu@gmail.com

Abstract

5G advanced technology has introduced innovative multimedia services, thus propelling the rise of the mobile multimedia market. While the ubiquity of mobile multimedia services has boosted customer convenience, it has also unlocked the way to possible security concerns that potentially damage users' privacy and service providers' properties. Several standards and researches have been proposed to counter these threats, notably those concentrating on authentication between users and service applications. Unfortunately, these studies are limited to address a new challenge, secure handover among distributed application functions in 5G Multi-access Edge Cloud (MEC) environments. Motivated by this, we present a handover authentication protocol with push-key and pull-key optimization options in 5G mobile multimedia application services. The proposed protocol provides mutual authentication, secure key exchange, confidentiality, and integrity. Furthermore, it also supports perfect forward secrecy, optimal handover, and anonymity, which were not previously considered in prior studies. While formal verification through BAN Logic and Scyther proved that the protocol is secure against attacks that violate the supported security requirements, comparative analysis of the proposed protocol against existing studies demonstrates that the protocol is simultaneously efficient.

Keywords: Authentication, 5G, Handover, AKMA, Formal verification

1 Introduction

The high portability of mobile is bringing about a massive change in the multimedia market. Currently, various users are consuming mobile content to a great extent with improved communications coupled with expanding competition in the field of new media globally. The mobile multimedia service market is expected to grow at the most significant CAGR of 14.87% from 2021 to 2027. One of the major factors in this change is the construction of 5G infrastructure due to changes in user preferences for smartphones and next-generation communication technologies [1]. The quantity and quality of the mobile multimedia application service have been rapidly expanded with the spread of 5G wireless communication technology. With the high throughput and low latency of 5G, users can use high-quality services anytime, anywhere. The service-oriented network can provide optimized services to users based on network function virtualization technology and network slicing technology [2]. Mobile multimedia application service through a service-oriented network should check the authority of the user accessing service. In other words, a terminal authentication procedure between the service user and the service provider is required to provide safe service. In general, authentication between service users and application service servers is based on credentials such as tokens, certificates, username/password, etc. [3]. In order to protect the communication between the user terminal and the application, it is reasonable to use a credential using a session key known only to the communication party. In fact, various wireless communication technologies ranging from cellular, Wi-Fi, and Bluetooth use a pre-shared key or certificate in authentication with a user terminal. However, managing a large scale of pre-shared keys and certificates can be a heavy burden on service providers.

For this reason, the Third Generation Partnership Project (3GPP) is leading the standardization of Authentication and Key Management for Application (AKMA) to support authentication for the application layer based on 3GPP credentials belonging to 5G network systems [4]. AKMA is an extension of Generic Bootstrapping Architecture (GBA) [5] and Battery Efficient Security for very low Throughput Machine Type Communication (MTC) devices (BEST) [6], which performed similar services in the previous generation of mobile communication systems. GBA and BEST require additional authentication procedures for application access in addition to authentication for network access. In contrast, AKMA is efficient by recycling the 3GPP credential created in the 5G initial authentication.

The mobile multimedia service should be able to provide a stable service to the user in any situation. Application-level handover is expected in 5G MEC networks because the forthcoming telecommunication integrated network application (NetApp) is composed of distributed application functions (AFs), each of which is deployed in an edge cloud. Moreover, users will subscribe to various application services, which will cause frequent application handover. Unfortunately, [4] did not specify an appropriate handover procedure for AKMA. Meanwhile, the mobile multimedia service frequently uses privacy information such as terminal information, application type, and user location that can specify the user. Consequently, an application-level security protocol that can support optimized handover and privacy

*Corresponding Author: Ilsun You; E-mail: ilsunu@gmail.com
DOI: 10.53106/160792642022112306008

protection while maintaining the current security level is essential considering the increasing use of mobile applications [6-8]. In this paper, we propose a secure and efficient handover authentication protocol for the 5G mobile multimedia application service that supports mutual authentication, secure key exchange, confidentiality, integrity, perfect forward secrecy, and anonymity. Furthermore, we evaluate the designed protocol through formal security analysis and comparison with existing studies. The main contributions of this paper can be summarized as follows:

- We analyzed Authentication and Key Management for Application (AKMA) proposed in [4] and design a handover authentication protocol to support the optimized handover with two options, push key and pull key.
- We verified the proposed protocol with two well-known formal security analysis tools, BAN-Logic [9] and Scyther [10]. According to the results of formal security analysis, the proposed protocol can satisfy the security properties such as confidentiality, integrity, mutual authentication, secure key exchange, perfect forward secrecy, and anonymity.
- We compare and evaluate the proposed protocol with related studies regarding security properties, computational costs and handover latency.

The remainder of this paper is organized as follows. Section 2 describes the related works about target environments. Section 3 provides the preliminaries that require for the proposed handover authentication protocol. In Section 4, we design two kinds of authentication protocol in handover phase, push key and pull key options. Section 5 and Section 6 present formal security analysis and comparative analysis. Finally, we conclude this paper in Section 7.

2 Related Works

In this section, we discuss the target environments including Service-Based Architecture (SBA) and Authentication and Key Management for Application (AKMA).

2.1 Service-based Architecture

Service-Based Architecture (SBA) provides a modular framework for deploying common applications using components from various sources and vendors. 3GPP defined the SBA so that the control area and common data storage of the 5G network are provided through the authorized Network Function (NF) to access each other. Assuming the role of service consumer or service producer, NF is independent and reusable. Each NF service is exposed via a Service-Based Interface (SBI) using a REpresentational State Transfer (REST) interface using HyperText Transfer Protocol (HTTP)/2. 5G SBA consists of various functions as follows [11].

- Network function Repository Function (NRF): With NF built using the microservices methodology, 5G SBA is ultimately evolving into a complete service mesh that includes service discovery, load balancing, encryption, authentication, and authorization. However, the current SBA uses a centralized search framework that utilizes NRF. NRF maintains a record of available NF instances and supported services, can allow subscriptions of other NF instances, and receive registration notifications from NF instances of a specified type. The NRF supports service discovery by receiving the NF instance discovery request and detailed information supporting a specific service.
- Network Slice Selection Function (NSSF): Network slicing is a fundamentally new feature of 5G infrastructure that provides a high level of deployment flexibility and efficient resource utilization when deploying various network services and applications. Logical end-to-end network slicing has predetermined functions, traffic characteristics, and service level agreements. It includes User Plane Function (UPF), Session Management Function (SMF), Policy Control Function (PCF), Mobile Virtual Network Operator (MVNO), or subscriber Virtualized resources are included to handle the group's needs. An Access and Mobility Management Function (AMF) instance supporting a UE is common to all network slices to which the UE belongs. Identification of a network slice is made through Single Network Slice Selection Assistance Information (S-NSSAI), and selection of a network slice instance is performed by the first AMF receiving a UE registration request. The request searches for a slice allowed in Unified Data Management (UDM) and requests an appropriate network slice instance from the NSSF.
- Unified Data Management (UDM): UDM provides services to other SBA functions such as AMF, SMF, and Network Exposure Function (NEF). UDM is generally recognized as a stateful message store that holds information in local memory, but it can also be stateless and store information in the Unified Data Repository (UDR). UDM is similar to Home Subscriber Server (HSS) and is used by AMF and SMF to retrieve subscribers' data and context and prove authentication credentials.
- Policy Control Function (PCF): PCF supports an integrated policy framework within the 5G infrastructure to manage the operation of the network. PCF provides appropriate policy rules for control plane functions by accessing the subscription information needed to make policy decisions in UDM. PCF is similar to Policy and Charging Rule Function (PCRF) of the EPC structure.
- Service Communication Proxy (SCP): SCP is not required to operate 5G SBA, but is required to operate in a MEC (Multi-access Edge Computing) environment. SCP provides a single point of entry into the network function cluster upon successful discovery by the NRF. SCP forms a hierarchical 5G service mesh with NRF. This allows SCP to offload NRF from the numerous distributed services that ultimately make up the network operator's infrastructure as a delegated discovery point in the data center.

SBA is built on web technologies and protocols to enable flexible and scalable implementations using virtualization, container technologies, and cloud-based processing platforms. However, the wide range of 5G system architectures, virtualized implementations, and cloud processing make 5G SBAs with diverse high-level security requirements.

Recognizing these changes in security requirements, security for SBA is designed to provide appropriate security for new use cases and virtualized implementations [12].

2.1.1 SBA Security for Direct Communication

Like general 5G security, SBA security is specified in 3GPP TS 33.501 [13]. An important feature of SBA is that NFs can communicate with each other. Interact with request/response or subscription/notification between NF service consumers and NF service producers. This requires careful specification of how to implement communication between NFs, secure each NF's service API, and properly authorize the use of these APIs. It also affects the choice of security protocol used to secure interactions, as the underlying protocol stack is based on web protocols such as HTTP and JavaScript Object Notation (JSON). The network operator should generally implement the following security mechanisms to secure communication between different entities.

- Authentication between communication endpoints to counter message spoofing
- Communication transport protection (confidentiality, integrity, retransmission protection) to prevent message tampering, repudiation, and information exposure
- Authorization of requests to prevent elevation of privileges

Security in direct communication between NFs can be supported through the following two components.

- Mutual authentication and transmission security between NFs based on Transport Layer Security (TLS) 1.2 [14] and TLS 1.3 [15]
- Token-based authentication for access of NF service consumers to services provided by NF service producers based on OAuth 2.0 [16]

TLS 1.2 and 1.3 are state-of-the-art protocols used to secure communications on the Internet and other networks. Previous generations of mobile networks and 5G networks outside the SBA rely on Internet Protocol Security (IPSec). Although IPSec is not without its problems from a security point of view, TLS makes it easy to terminate security directly in the NF instead of the secure gateway used to secure the entire network domain. This approach is suitable for virtualized implementations using multi-borrowing. Token-based authentication using OAuth 2.0 is a way to perform authentication in dynamic virtualization implementations. It is based on a central authentication server that issues an access token after authenticating to the client. When a client calls a service, it provides an access token to the NF service producer. The NF service producer validates the access token to grant access to the consumer. In the SBA, the NRF acts as an authorization server. Authorization rules can be provided by NF service producers themselves during registration with NRF. Token-based authorization must be combined with authentication to be valid. NRF and NF service consumers mutually authenticate before the NRF issues an access token. In addition, transmission protection is essential to prevent tokens from being intercepted or misused.

2.1.2 SBA Security for Indirect Communication

In indirect communication, instead of the NF consumer directly interacting with the NF producer, a Service Communication Proxy (SCP) is introduced into the path between the NF consumer and the producer. An important aspect from a security point of view is that consumers and producers must rely on SCPs to send service requests on behalf of NF service consumers and deliver responses from NF service producers to consumers. This affects the underlying trust model. SCPs do not simply deliver service requests from consumers. For both standardized and proprietary functions of the SCP, it must be possible to enable the SCP and modify service request messages. Therefore, mutual authentication and transmission security for each hop is based on TLS. Still, end-to-end transmission security between consumers and producers satisfied by TLS in direct communication is impossible in indirect communication. However, token-based authentication, which has already been specified for direct communication, allows producers to prove that an SCP is authorized to act on its behalf. The SCP passes the valid access token issued to the NF service consumer to the producer. SCPs in some deployment models can request access tokens on behalf of NF service consumers. This is only possible if the NRF ensures that only authorized SCPs can request access tokens on behalf of consumers. To this end, the consumer proves that the consumer has accepted it by sending a self-signed certificate used by the NRF to the SCP. This mechanism performs token-based authentication. As mentioned earlier, for communication between NF service consumers and producers, SCPs can use access tokens to prove that they have the authority to act on their behalf.

2.2 Authentication and Key Management for Application

The application service using the network improved users' quality of life and inspired the service providers to invest in infrastructure. To establish a safe application service use environment, authentication of the user accessing the application is essential. Unfortunately, GBA [5] and BEST [6], the application authentication methods of the previous network generation, are not suitable for 5G application due to resource waste and efficiency problems due to duplicate authentication. To this end, 3GPP is in the process of standardizing authentication and key management for applications based on the 3GPP credentials generated during the initial network connection.

2.2.1 Methods of the Previous Generation Network

This section analysed the authentication and key management for applications focusing on TS 33.535 [4] and TS 38.535 [17]. 3GPP proposed Authentication and Key Management for Application (AKMA) to support authentication and key management through 3GPP credentials supported by the 5G system for 3GPP services as well as current third-party applications. AKMA is a service that improves the problems of GBA and BEST used in the previous generation of mobile communication systems. It is an authentication and key management service based on the user's cellular subscription information to access the application server.

GBA consists of GBA Bootstrapping and GBA Bootstrapping Usage. In GBA Bootstrapping, the authentication protocol is executed through the Bootstrapping Server Function (BSF), which acts as an intermediary for the authentication of the UE and the home server. GBA Bootstrapping procedure allows the UE and the BSF to share the bootstrap key. When the bootstrap key is shared through GBA Bootstrapping, GBA Bootstrapping Usage is executed. In this procedure, the UE may use a bootstrap key to secure communication with a Network Application Function (NAF). The UE provides the temporary identifier to the NAF, and the NAF delegates the session key calculation by sending the temporary identifier of the UE to the BSF. Through this, the UE and the NAF can share the session key to protect the communication channel. GBA did not support SBI, but with the advent of 5G, it is improving in the direction that can help SBI. SBI refers to API-based communication between virtualized network functions, and API calls through SBI can be used to call services. However, it is inefficient in that the application key is directly derived from CK and IK in the key-derivation procedure, and an additional Authentication and Key Agreement (AKA) protocol is executed after initial authentication.

BEST is similar to GBA but is optimized for low-computational devices with limited battery life and high latency. BEST provides the key exchange service and the user plane protection service. BEST routes all messages through a Home Security Endpoint (HSE). User plane protection service is divided into UE-to-HSE mode and UE-EAS mode. The UE-to-HSE mode protects the user plane traffic sent from the UE to the HSE, and the UE-EAS mode protects the traffic through a key established between the UE and the Enterprise Application Server (EAS). BEST does not consider SBI and cannot operate in a 5G network at the same time. Also, similar to GBA, the application key is derived from CK and IK, and additional authentication procedures are required after initial authentication.

2.2.2 AKMA Architecture

In a 5G network, only authorized users should be able to access approved applications. It requires authentication of the user terminal at the application layer. The authentication can be based on credentials such as a token, certificate, or user ID/password between the user terminal and the application [3]. 3GPP proposed the authentication and key management for application (AKMA) scheme through TS 33.535 [4] and TS 38.535 [17]. AKMA supports authentication and key exchange at the application layer regardless of the access medium, cellular, Wi-Fi, or Bluetooth. Also, [4] and [17] presented new functions and security requirements for AKMA, as shown in Table 1 and Table 2.

Table 1. Network function for AKMA

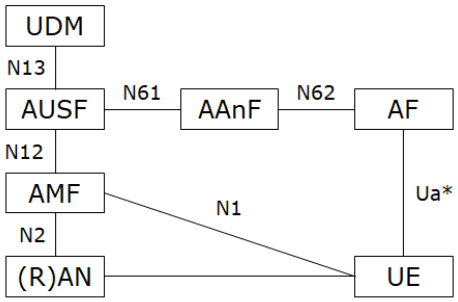
Network Function	Role
AKMA Anchor Function (AAnF)	• Anchor function of Home Public Land Mobile Network (HPLMN) • Storing the K_{AKMA} received from AUSF after the initial authentication procedure • Create key material to be used between UE and AF and maintain UE AKMA Context
Application Function (AF)	• Request K_{AF} to AAnF with AKMA Key Identifier (A-KID) • AF should be authenticated and authorized to the network before receiving K_{AF} • AF located in the internal network performs AAnF Selection
Network Exposure Function (NEF)	• Activation and approval for the external AF • Forward the request of the external AF to AAnF • Perform AAnF Selection
Authentication Server Function (AUSF)	• Provide AAnF with SUPI and AKMA key material • Perform AAnF Selection
Unified Data Management (UDM)	• Storing the AKMA subscription data

Table 2. Security requirements for AKMA

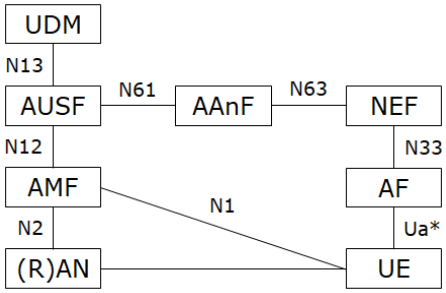
Security requirement	Rule
General	• Reuse the same UE subscriptions and credentials used for 5G access • Reuse the 5G initial authentication procedure and method specified in 3GPP TS 33.501 • The SBA interface between AAnF and AUSF should support confidentiality, integrity, and replay protection • The SBA interface between AAnF and AF/NEF should support confidentiality, integrity, and replay protection • K_{AF} should be provided with a maximum lifetime
Ua* reference point	• The Ua* reference point depends on the application • Ua* protocol should transfer A-KID • Ua* reference point should be protected with K_{AKMA}
A-KID	• A-KID is globally unique • A-KID should be usable as a key identifier in Ua* protocol • AF should be able to identify serving AAnF from A-KID

[17] defines the application provider as AF, and AF delegates the user authentication to HPLMN. As a result, AF maintains less sensitive user subscription data, and users manage less password information. In addition to the existing 5G network functions, the AKMA structure includes two new network functions: AAnF and AF. AAnF performs an anchor function for AKMA in HPLMN. AAnF receives and stores K_{AKMA} from AUSF after the initial authentication procedure. The K_{AKMA} granted by AUSF is used to derive the key later. AF is a function that provides a service to a user. K_{AF} is given

to AAnF to configure a session with the user terminal when an access request is received from a user terminal. AKMA is divided into two architectures in reference point according to the location of AF, as shown in Figure 1(a) shows an architecture for accessing the internal application, and Figure 1 (b) shows for accessing the external application.



(a) Internal application



(b) External application

Figure 1. Architecture of AKMA

2.2.3 AKMA Key Hierarchy

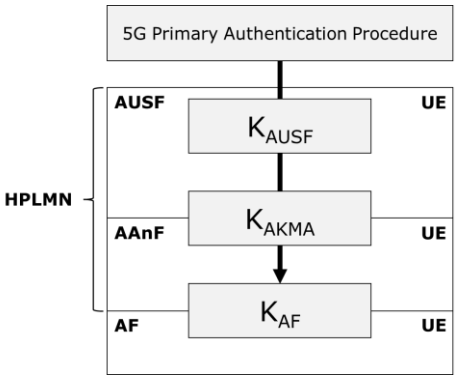


Figure 2. Key hierarchy of AKMA

Each key used in AKMA is derived based on the K_{AUSF} exchanged in 5G initial authentication. K_{AKMA}, the key used in AAnF, is derived from K_{AUSF} by ME and AUSF and is delivered by AUSF to AAnF. In addition, K_{AF}, the key used in AF, ME, and AAnF, derives from K_{AKMA}, and AAnF provides K_{AF} to the authorized AF. The key hierarchy of AKMA is proposed in TS 33.535 [4] as shown in Figure 2. K_{AKMA} and A-KID used at the top of the AKMA key hierarchy are implicitly valid until the next initial authentication succeeds. Whenever the initial authentication, communication participants renew K_{AKMA} and A-KID. According to the network operation policy, K_{AF} uses an explicit lifetime, and AAnF provides the K_{AF} to AF.

2.2.4 AKMA Procedure

The above section analysed GBA and BEST used in the previous generation networks and AKMA, currently standardized in 3GPP. Establishing a secure channel between users and applications using GBA and BEST requires additional authentication procedures and server functions. 3GPP supports authentication and key management by using network initial authentication information by improving these shortcomings of GBA and BEST through AKMA. Figure 3 shows the AKMA procedure defined in TS 33.535.

Step 1: AUSF requests the UE's subscriber credential and authentication information from UDM/ARPF through Nudm_UEAuthentication_Get Request message during the 5G initial authentication procedure.
Step 2: If UE is a legitimate application user, UDM/ARPF determines whether to generate KAKMA, and transmits Nudm_UEAuthentication_Get Response message including necessary information to AUSF. Once confirming that UE is authorized for AKMA, AUSF derives K_{AKMA} and A-KID from K_{AUSF} through the routing identifier received from UDM/ARPF. Similarly, UE also computes K_{AKMA} and A-KID from K_{AUSF} prior to communicating with AF.
Step 3: AUSF selects AAnF to provide the service to UE through the AAnF Selection procedure and transmits the SUPI and K_{AKMA} of UE in Naanf_AKMA_Key Registration Request message.
Step 4: AAnF stores the SUPI and K_{AKMA} of UE and transmits Naanf_AKMA_KeyRegistration Response message to AUSF. AUSF does not need to store AKMA key material after communication with AAnF. When re-authentication is required, AUSF generates new AKMA key material and transmits it to AAnF.
Step 5: UE derives K_{AKMA} and A-KID from K_{AUSF} before initiating communication with AF(1) and transmits A-KID to AF(1) through Application Session Establishment Request message.
Step 6: AF(1) searches for a context matching with A-KID, and if it does not exist, selects AAnF through AAnF Selection. Then, it requests the selected AAnF to provide KAF(1) by sending Naanf_AKMA_Application Key_Get_Request message which includes A-KID received from UE and its identifier AF_ID.
Step 7: AAnF checks if AF(1) can provide service, and identifies K_{AKMA} through the received A-KID. Then, $K_{AF(1)}$ is derived from K_{AKMA} and its lifetime is decided, both of which are in turn sent back to AF(1) through Naanf_AKMA_ApplicationKey_Get Response message.
Step 8: AF(1) stores the received $K_{AF(1)}$ and its lifetime, followed by informing UE that session establishment is complete through Application Session Establishment Response message. Upon receipt of the response message, UE computes $K_{AF(1)}$ while being ready for the next step.

In 5G MEC environments, users can access various subscribing application services, sequentially or in parallel. In this time, the connection between the user equipment and the AF should also be handover. However, the current AKMA does not specify a proper handover procedure except for the AF access procedure in the external network. A secure and

efficient handover procedure is essential to provide a seamless application service to users.

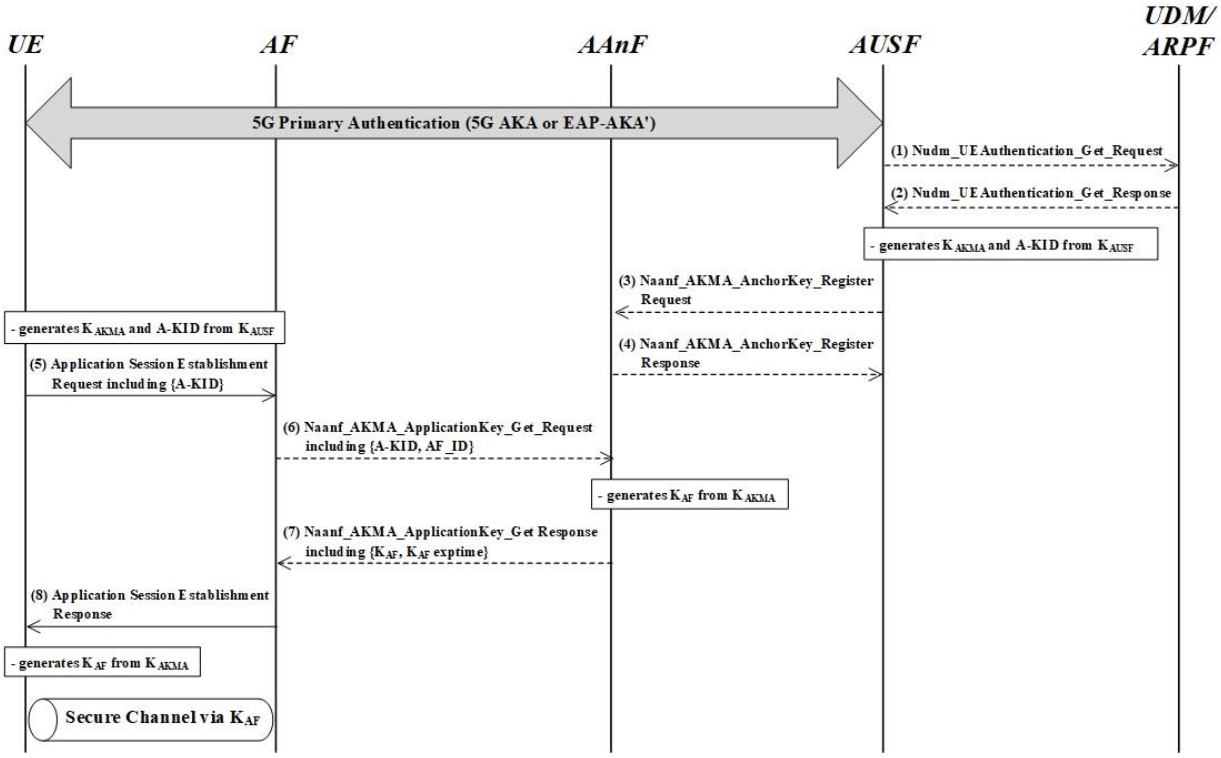


Figure 3. Procedure of AKMA

3 Preliminaries

In this section, we introduce the system model of the proposed secure handover authentication protocol and describe the adversary model for 5G application handover scenario and the basic concept of the Elliptic-Curve Diffie-Hellman (ECDH).

3.1 System Model

As shown in Figure 4, the system model of the proposed secure handover authentication protocol consists of 5 network entities; User Equipment (UE), Application Function (AF), AKMA Anchor Function (AAnF), Authentication Server Function (AUSF), and Unified Data Management (UDM).

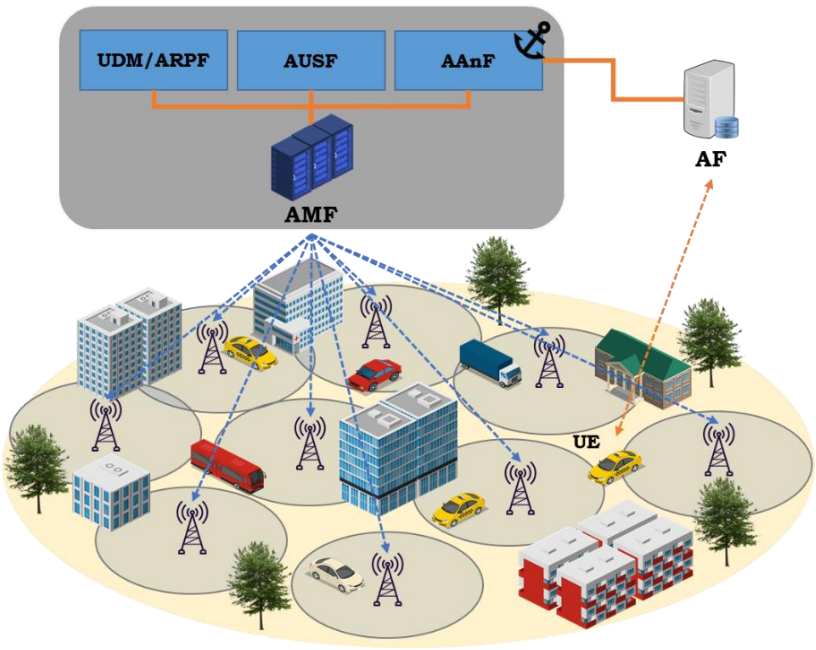


Figure 4. System model of the proposed handover authentication protocol

3.2 Adversary Model

Users in the network can be lost or harmed by various threats. These threats represent violations of the confidentiality, integrity, and availability of properties. They can be caused by vulnerabilities in a system, such as design flaws, configuration mistakes, or inaccuracies in security policies.

The handover authentication protocol we propose needs to work in an open channel where an attacker can access encrypted information. In other words, an attacker can get a message (either encrypted or not) traversing the network, initiate a new communication, or impersonate another participant. We model these attackers with the well-known Dolev-Yao threat model [18]. In the Dolev-Yao threat model, an adversary can try any attacks against all messages exchanged via an open channel, including intercepting, eavesdropping, and modifying messages. Thus, the proposed handover authentication protocol should support following security requirements.

- Mutual Authentication: Communication participants should authenticate each other.
- Secure Key Exchange: The authentication key and cipher key should be securely negotiated between communication participants.
- Perfect Forward Secrecy: The secret key used in the current session should not be derived in any way from the past key.
- Integrity and Confidentiality: Communication messages exchanged between authorized participants should be protected from access and modification by illegal participants.
- Anonymity: The identifier of the service user should not be disclosed to others.

3.3 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is an efficient method for asymmetric key exchange and encryption with low memory and key size [19]. The proposed handover authentication protocol supports a relatively secure and efficient public key exchange between UE and AF through Elliptic Curve Diffie-Hellman key exchange (ECDH) [20]. As shown in Table 3, we choose ECDH Ephemeral (ECDHE) option, which uses the temporary key and supports the perfect forward secrecy.

Table 3. Elliptic Curve Cryptography

The domain parameters ($p, a, b, g, n, and\ h$) are set between participants Alice and Bob.

Step 1: Generating the key pairs
1-1. Alice chooses an ephemeral private key d_A from {1, …, *n*-1}.
1-2. Alice derives a public key Q_A from d_A via the following operation.

$$Q_A = d_A \,.\, g$$

1-3. Bob also chooses an ephemeral private key d_B from {1, …, *n*-1}.
1-4. Bob derives a public key Q_B from d_B via the following operation.

$$Q_B = d_B \,.\, g$$

Step 2: Exchanging the public keys
2-1. Alice sends public key Q_A to Bob.
2-2. Bob sends public key Q_B to Alice.

Step 3: Computing the session key
3-1. Alice computes the session key S via the following operation.

$$S = d_A \,.\, Q_B = d_A \,.\, d_B \,.\, g$$

3-2. Bob computes the session key S via the following operation.

$$S = d_B \,.\, Q_A = d_B \,.\, d_A \,.\, g$$

3-3. Alice and Bob share the session key *S* and remove ephemeral private keys.

p: a prime number indicating the finite field size.
a, b: the coefficients for the chosen elliptic curve equation.
g: the base point to generate a subgroup.
n, h: the order and cofactor of the subgroup.

4 Proposed Protocol

In this section, we describe the proposed handover authentication protocol used for application to support the secure communication between UE and NF. The proposed protocol consists of registration and initial access, and two kinds of handover authentication phases. Table 4 shows the notations used in this section.

4.1 Registration and Initial Access

In the registration phase, UE and AUSF exchange the secret key K_{AUSF} via the 5G initial authentication procedure and obtain permission for the application use from UDM/ARPF. When the application usage right is granted, UE and AUSF derive K_{AKMA} and A-KID from K_{AUSF}, and AAnF receives K_{AKMA} from AUSF. At this point, UE derives K_{AF} for accessing AF through Ua* reference point. K_{AF} is derived

from K_{AKMA} shared between UE and AAnF to provide secure Ua* reference point communication. The design of the proposed protocol registration and initial access phase is shown in Figure 5.

Table 4. Notations

Notation	Description
UE	User equipment
AF	Application function
AAnF	AKMA anchor function
AUSF	Authentication server function
UDM	Unified data management
ARPF	Authentication credential repository and processing function
ID_A	A's identifier
AID_{A-B}	Anonymous identifier used between A and B
K_A	A's secret key
K_{AKMA}	Intermediate anchor key derived from K_{AUSF} to be used between UE and AUSF/AAnF
K_{AF}	Secret key derived from K_{AKMA} to be used between UE and AF
X, Y	ECDH private key
Seq	Sequence number
n_X	x-th nonce
SK	Session key
HM	Hash-based message authentication code

Step 1-1: AUSF requests UE's subscriber credential and authentication information from UDM/ARPF through Nudm_UEAuthentication_Get Request message during the 5G initial authentication procedure.

Step 1-2: UDM/ARPF determines whether to generate K_{AKMA} and transmits Nudm_UEAuthentication_Get Response message including necessary information to AUSF. AUSF derives K_{AKMA} and A-KID from K_{AUSF} through the routing identifier received from UDM/ARPF. UE also derives K_{AKMA} and A-KID from K_{AUSF} before communicating with AF. A-KID is an identifier for recognizing K_{AKMA} and consists of 'username@realm' according to the format specified in IETF RFC 7542 [21]. The username part consists of the routing identifier and A- TID, and the realm part consists of the Home Network Identifier. The A-TID is derived from K_{AUSF}.

Step 1-3: AUSF selects AAnF to provide the service to UE through the AAnF Selection procedure and transmits the SUPI and K_{AKMA} in Naanf_AKMA_KeyRegistration Request message.

Step 1-4: AAnF stores the SUPI and K_{AKMA} and transmits Naanf_AKMA_KeyRegistration Response message to AUSF. AUSF does not store AKMA key material but generates new AKMA key material in the re-authentication process and sends it to AAnF.

Step 2-1: UE derives K_{AKMA} and A-KID from K_{AUSF} before initiating communication with AF(1) and transmits A-KID to AF(1) with Application Session Establishment Request message.

Step 2-2: AF(1) searches for a Context matching A-KID and, if it does not exist, selects AAnF through AAnF Selection. After AAnF Selection, AF(1) sends A-KID and own ID ($ID_{AF(1)}$) to AAnF with Naanf_AKMA_ ApplicationKey_Get_Request message for requesting $K_{AF(1)}$.

Step 2-3: AAnF checks whether AF(1) can provide service, and identifies K_{AKMA} through the received A-KID. $K_{AF(1)}$ is derived from K_{AKMA}, and the lifetime of $K_{AF(1)}$ is included in Naanf_AKMA_ApplicationKey_Get Response message and transmitted to AF(1).

Step 2-4: AF(1) stores the received $K_{AF(1)}$ and notifies UE that the session establishment has been completed through Application Session Establishment Response message, including a sequence number Seq1. Seq1 is used to generate an anonymous identifier in the subsequent handover phase.

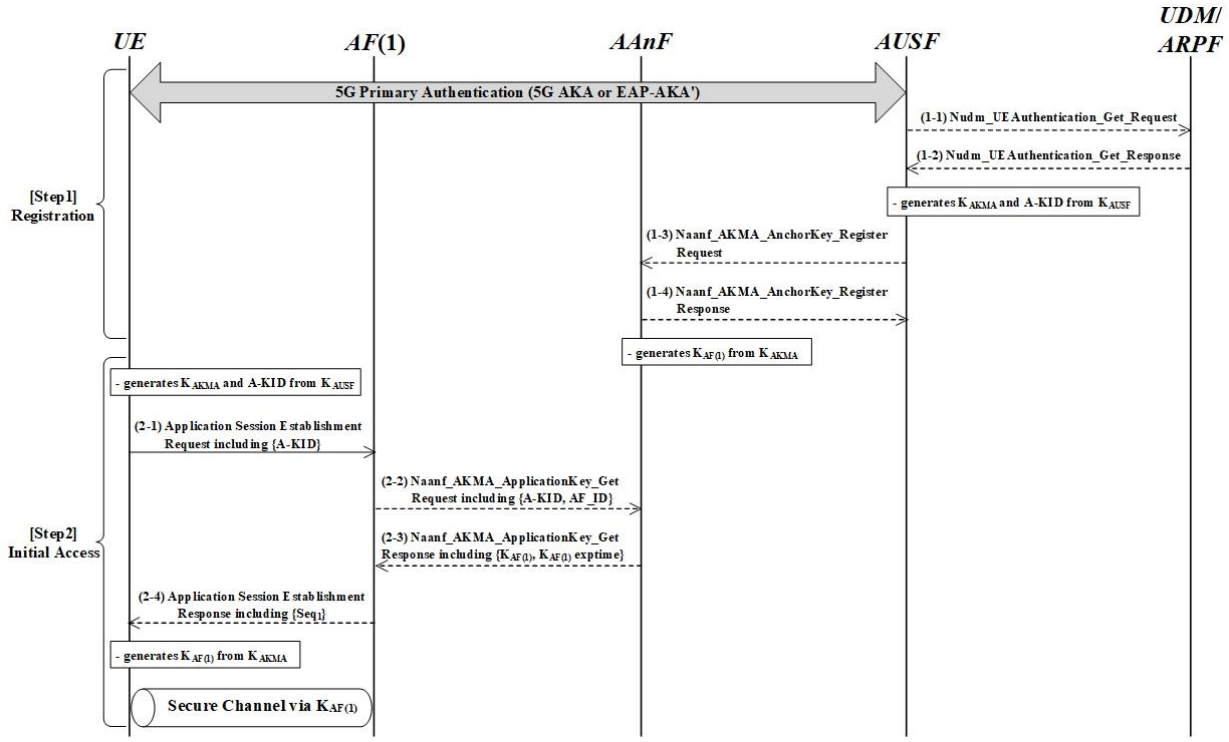


Figure 5. Registration and initial access phase

4.2 Handover Phase with Push Key Option

The handover phase of the proposed protocol is divided into two options: Push key and Pull key. Both options have the same procedure before the handover decision but differ depending on when the UE's context is delivered. In the former option, Push Key, handover is decided faster than UE movement. According to the handover decision, UE requests AF(1) to transmit the context required for handover to AF(2). AF(2) performs authentication and key exchange with the UE through the context received from AF(1). The handover phase with the push key option is shown in Figure 6.

Step 3-1: When UE detects the handover situation, UE generates the anonymous identifier $AID_{UE\text{-}AF(1)}$ with its identifier and Seq1. $AID_{UE\text{-}AF(1)}$ guarantees the anonymity of UE, preventing the identification of UE by anyone (including attackers) other than the communication participants. UE transmits Application Handover Decision message to AF(1) to transfer its context stored in AF(1) to AF(2). At this time, the integrity of Application Handover Decision message is protected by the message authentication code HM1 generated via $K_{AF(1)}$ between UE and AF(1).

Step 3-2: After receiving Application Handover Decision message from UE, AF(1) verifies the HM1 and restores the identifiers of UE (ID_{UE}) through $AID_{UE\text{-}AF(1)}$ and Seq1. After that, AF(1) transmits Application Handover Context Transfer message, including contexts of UE such as $K_{AF(1)}$, ID_{UE}, and Seq1, to the AF(2) via a secure channel.

Step 3-3: Upon receiving Application Handover Context Transfer message from AF(1), AF(2) stores contexts of UE($K_{AF(1)}$, ID_{UE}, and Seq1) and responds to AF(1) with Application Handover Context Transfer Complete message.

Step 3-4: When UE moves and attaches to AF(2), UE transmits Application Handover Request message to AF(2). Application Handover Request message contains the anonymous identifier ($AID_{UE\text{-}AF(1)}$), the identifier of the receiver ($ID_{AF(2)}$), a randomly generated nonce (n_1), ECDH public key ($X \cdot G$), and the message authentication code (HM2). In addition, the integrity of Application Handover Request message is protected via HM2 generated through $K_{AF(1)}$.

Step 3-5: AF(2) firstly checks $AID_{UE\text{-}AF(1)}$ with Seq1 and verifies HM2 with $K_{AF(1)}$. We suppose $AID_{UE\text{-}AF(1)}$ and HM2 are valid. In that case, AF(2) generates ECDH private key (Y), a sequence number (Seq2), and a randomly generated nonce (n_2) and computes a new anonymous identifier ($AID_{UE\text{-}AF(2)}$), and ECDH public key ($Y \cdot G$). With UE's ECDH public key and freshly generated AF(2)'s ECDH private key (Y), AF(2) computes the session key (SK). Then, AF(2) sends Application Handover Response message including $AID_{UE\text{-}AF(2)}$, $ID_{AF(2)}$, n_1, n_2, and $Y \cdot G$. The integrity of Application Handover Response message is protected via hash-based message authentication codes (HM3 and HM4) generated through the session key SK and $K_{AF(1)}$, respectively.

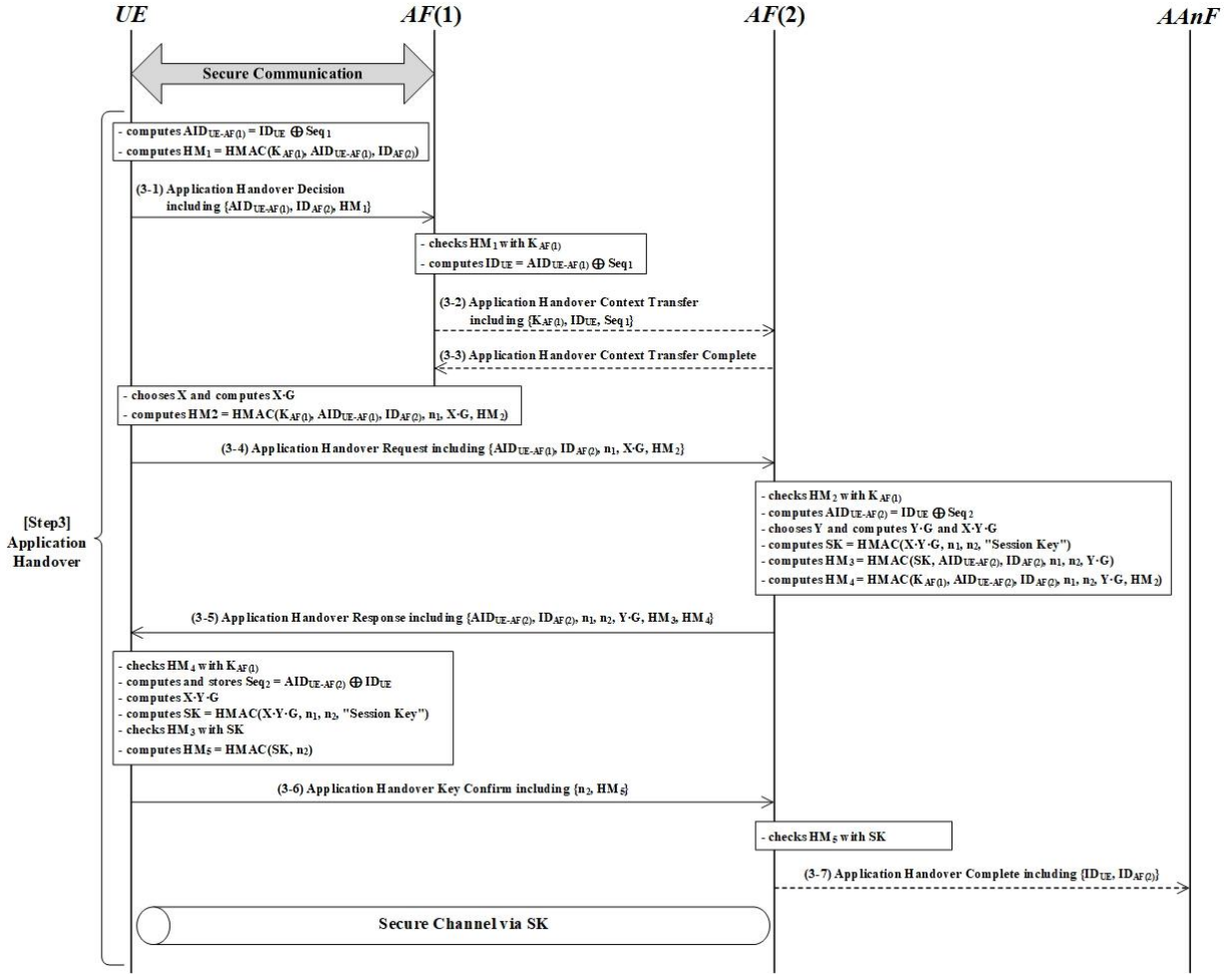


Figure 6. Handover phase with push key option

Step 3-6: After receiving Application Handover Response message, UE verifies HM4 and $AID_{UE\text{-}AF(2)}$ through $K_{AF(1)}$ and ID_{UE}. If HM4 and ID_{UE} are valid, UE can trust Application Handover Response message and stores the subsequent sequence number Seq2. Then, UE calculates the session key (SK) via its ECDH private key (X), AF(2)'s ECDH public key (Y·G), and nonces generated by both participants. With session key SK, UE verifies HM3 and realizes that AF(2) believes in SK in this session. UE includes n_2 and HM5 protected by SK in Application Handover Key Confirm message and transmits it to AF(2).
Step 3-7: AF(2) verifies HM5 included in Application Key Confirm message to confirm whether the session key is securely exchanged with UE. If HM5 is valid, AF(2) reports to AAnF that handover authentication between UE and AF(2) is successful.

4.3 Handover Phase with Pull Key option

Contrary to the above push key option, the pull key option induces AF(2) to request UE's context from AF(1) during the handover phase. AF(2) executes authentication and key exchange with UE through UE's context received from AF(1). The handover phase with the pull key option is shown in Figure 7.

Step 3-1: UE generates ECDH private key (X) and ECDH public key (X·G), a randomly generated nonce (n_1), and an anonymous identifier $AID_{UE\text{-}AF(1)}$ in advance. When UE moves to the new network, including AF(2), UE transmits Application Handover Request message to AF(2). Application Handover Request message contains $AID_{UE\text{-}AF(1)}$, n_1, X·G, the identifier of the previous application function ($ID_{AF(1)}$), and the message authentication code (HM1) protected by $K_{AF(1)}$.
Step 3-2: AF(2) forwards Application Handover Context Request message from UE to AF(1).
Step 3-3: Upon receiving Application Handover Context Request message from AF(2), AF(1) verifies HM1 with $K_{AF(1)}$ and extracts ID_{UE} from $AID_{UE\text{-}AF(1)}$. Here, AF(1) can check that the previously accessed UE has requested handover and sends $K_{AF(1)}$ and ID_{UE} to AF(2) via a secure channel.
Step 3-4: After acquiring $K_{AF(1)}$ and ID_{UE} from AF(1), AF(2) generates following sequence number (Seq2), ECDH private key (Y), ECDH public key (Y·G), and a randomly generated nonce (n_2). AF(2) calculates a new anonymous identifier $AID_{UE\text{-}AF(2)}$ through IDUE and Seq2, and the session key (SK) through UE's ECDH public key (X·G) obtained in above step 3-1, its own private key (Y), and nonces (n_1, n_2) generated by both participants. Then, AF(2) transmits Application Handover Response message, including $AID_{UE\text{-}AF(2)}$, $ID_{AF(2)}$, n_1, n_2, and Y·G. Application Handover Response message is protected via message authentication codes (HM2, HM3).

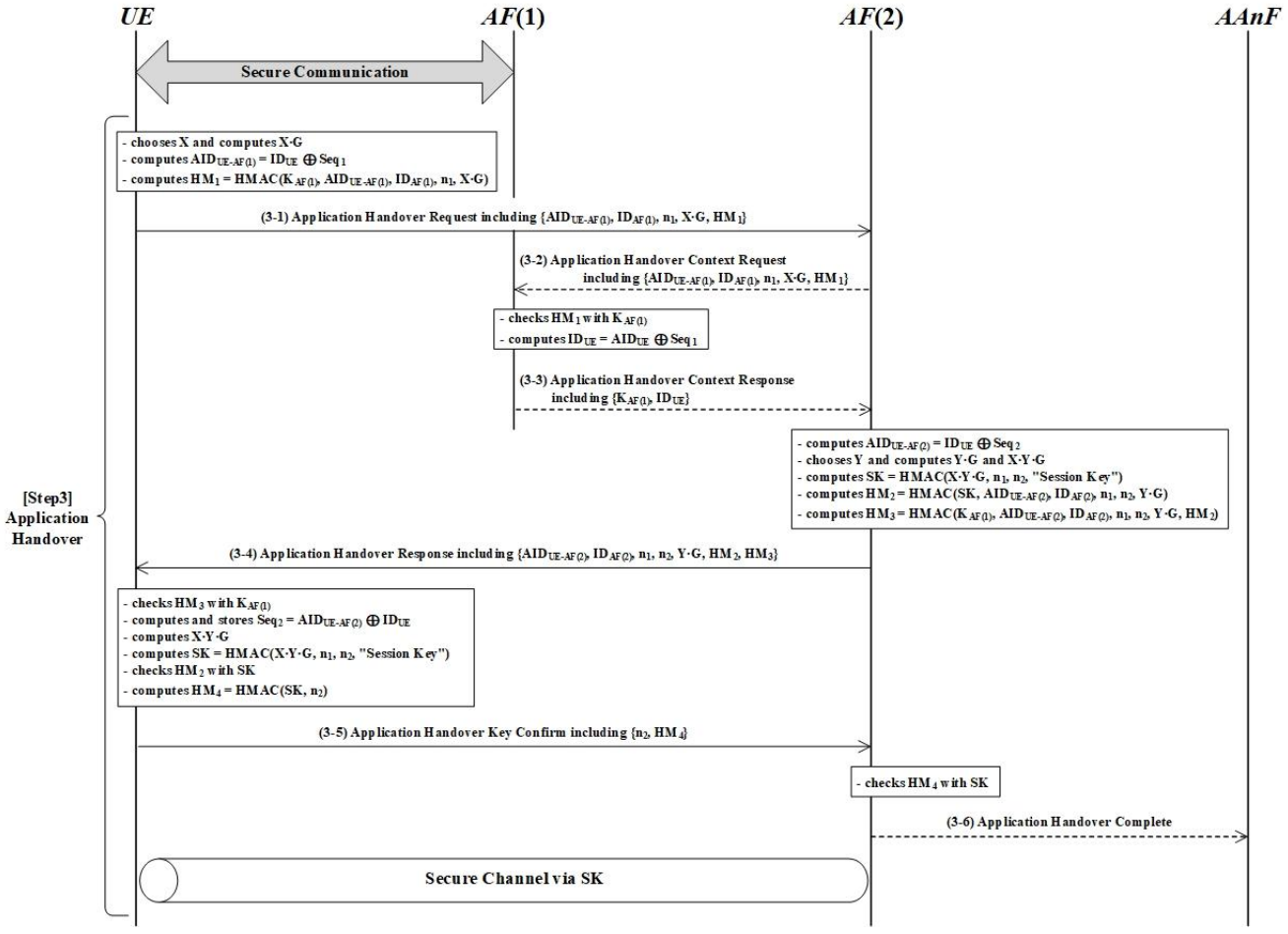


Figure 7. Handover phase with pull key option

Step 3-5: UE firstly checks HM3 and $AID_{UE\text{-}AF(2)}$ through $K_{AF(1)}$ and $AID_{UE\text{-}AF(2)}$, respectively. At this point, UE stores Seq2 for the following handover situation. The session key SK is derived from stored X, n_1, received $Y \cdot G$, n_2. With derived SK, UE verifies HM2 and can trust that SK is successfully exchanged with AF(2). UE sends Application Handover Key Confirm message, including n2 and HM4 protected by SK, to AF(2).

Step 3-6: AF(2) verifies HM4 included in Application Key Confirm message to confirm whether the session key is securely exchanged with UE. If HM5 is valid, AF(2) reports to AAnF that handover authentication between UE and AF(2) is successful.

5 Formal Security Analysis

This section analyzes the proposed handover authentication protocol using BAN logic [9] and Scyther [10]. Both are well-known tools in security analysis, and many researchers have used them to prove the security of the security protocols. The former is proposed by Michael Burrows, Martin Abadi, and Roger Michael Needham in 1989. The latter is an automated analysis tool proposed by Cas JF. Cremers. The proposed protocol consists of the registration and initial access phase and the handover phase with two options. However, the registration and initial access phase performs 5G initial authentication, identifier exchange, and key derivation according to the key hierarchy. Therefore, this section verifies the push and pull options in the handover phase.

5.1 BAN Logic

Security analysis through BAN Logic is sequentially performed by four procedures: (1) Idealization, (2) Assumption, (3) Goal, and (4) Derivation. Information processed by various methods such as encryption, digital signature, and message authentication code is modeled in Idealization through BAN Logic's unique notations and rules [9]. Assumption defines the proper environmental conditions for the target security protocol, such as network environments and secure channels. In Goal, the goals match the security requirements of the target security protocol. Finally, in Derivation, the security analysis result is derived from Idealization, Assumption, and Goal.

5.1.1 Push Key Option

- Idealization

The idealized forms of the push key option are as the following equations (I1)-(I5). The unprotected plaintext between message transmissions is excluded in Idealization.

(I1) $UE \rightarrow AF(1)$:
$$\langle AID_{UE-AF(1)}, ID_{AF(2)}, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)\rangle_{K_{AF(1)}}$$
(I2) $AF(1) \rightarrow AF(2)$:
$$\langle UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2), ID_{UE}, Seq_1\rangle_K$$
(I3) $UE \rightarrow AF(2)$:
$$\langle AID_{UE-AF(1)}, ID_{AF(2)}, n_1, X \cdot G, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\rangle_{K_{AF(1)}}$$
(I4) $AF(2) \rightarrow UE$:
$$\langle AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\rangle_{K_{AF(1)}}$$
(I5) $UE \rightarrow AF(2)$
$$\langle n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\rangle_{SK}$$

- Assumption

(A1) $AF(1) \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)$
(A2) $AF(1) \mid\equiv \#(AID_{UE-AF(1)})$
(A3) $AF(2) \mid\equiv AF(1) \overset{K}{\Leftrightarrow} AF(2)$
(A4) $AF(2) \mid\equiv \#(Seq_1)$
(A5) $AF(2) \mid\equiv AF(1) \mid\Rightarrow UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)$
(A6) $AF(2) \mid\equiv \#(Seq_1)$
(A7) $AF(2) \mid\equiv \overset{Y\cdot G}{\longrightarrow} AF(2)$
(A8) $AF(2) \mid\equiv \#(n_2)$
(A9) $UE \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)$
(A10) $UE \mid\equiv \#(n_1)$
(A11) $UE \mid\equiv \overset{X\cdot G}{\longrightarrow} UE$

- Goal

(G1) $AF(1) \mid\equiv UE \mid\equiv AID_{UE-AF(1)}$
(G2) $AF(1) \mid\equiv UE \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)$
(G3) $AF(2) \mid\equiv UE \mid\equiv AID_{UE-AF(1)}$
(G4) $AF(2) \mid\equiv UE \overset{SK}{\Leftrightarrow} AF(2)$
(G5) $UE \mid\equiv AF(2) \mid\equiv ID_{AF(2)}$
(G6) $UE \mid\equiv AF(2) \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)$
(G7) $UE \mid\equiv AF(2) \mid\equiv UE \overset{SK}{\Leftrightarrow} AF(2)$
(G8) $UE \mid\equiv UE \overset{SK}{\Leftrightarrow} AF(2)$
(G9) $AF(2) \mid\equiv UE \mid\equiv UE \overset{SK}{\Leftrightarrow} AF(2)$

- Derivation

(D1) $AF(1) \triangleleft \langle AID_{UE-AF(1)}, ID_{AF(2)}, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)\rangle_K$
(D2) $AF(1) \mid\equiv UE \mid\sim \left[AID_{UE-AF(1)}, ID_{AF(2)}, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)\right]$ $by\ (D1), (A1), MM$
(D3) $AF(1) \mid\equiv UE \mid\equiv \left[AID_{UE-AF(1)}, ID_{AF(2)}, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)\right]$ $by\ (D2), (A2), FR, NV$
(D4) $\boldsymbol{AF(1) \mid\equiv UE \mid\equiv AID_{UE-AF(1)}}$ $\boldsymbol{by\ (D3), BC}$
(D5) $\boldsymbol{AF(1) \mid\equiv UE \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(1)}$ $\boldsymbol{by\ (D3), BC}$
(D6) $AF(2) \triangleleft \langle UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2), ID_{UE}, Seq_1\rangle_K$
(D7) $AF(2) \mid\equiv AF(1) \mid\sim \left[UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2), ID_{UE}, Seq_1\right] by\ (D6), (A3), MM$
(D8) $AF(2) \mid\equiv AF(1) \mid\equiv \left[UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2), ID_{UE}, Seq_1\right] by\ (D7), (A4), FR, NV$
(D9) $AF(2) \mid\equiv AF(1) \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)$ $by\ (D8), BC$
(D10) $AF(2) \mid\equiv UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)$ $by\ (D9), (A5), JR$
(D11) $AF(2) \triangleleft \langle AID_{UE-AF(1)}, ID_{AF(2)}, Seq_1, n_1, X \cdot G, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\rangle_{K_{AF(1)}}$
(D12) $AF(2) \mid\equiv UE \mid\sim \left[AID_{UE-AF(1)}, ID_{AF(2)}, Seq_1, n_1, X \cdot G, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\right]$ $by\ (D11), (D10), MM$
(D13) $AF(2) \mid\equiv UE \mid\equiv \left[AID_{UE-AF(1)}, ID_{AF(2)}, Seq_1, n_1, X \cdot G, UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\right]$

$by\ (D11), (A6), FR, NV$

(D14) $\boldsymbol{AF(2) \mid\equiv\ UE \mid\equiv\ AID_{UE-AF(1)}}$ $\boldsymbol{by\ (D12), BC}$

(D15) $AF(2) \mid\equiv\ UE \mid\equiv$
$UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)$ $by\ (D12), BC$

(D16) $AF(2) \mid\equiv\ X \cdot Y \cdot G$ $by\ (D11), (A7), BC, DH$

(D17) $\boldsymbol{AF(2) \mid\equiv UE \overset{SK}{\Leftrightarrow} AF(2)}$
$\boldsymbol{by\ (D16), (D12), (A8), BC}$

(D18) $UE \lhd \langle AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\rangle_{K_{AF(1)}}$

(D19) $UE \mid\equiv\ AF(2) \mid\sim \left[AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\right]$
$by\ (D18), (A9), MM$

(D20) $UE \mid\equiv AF(2) \mid\equiv \left[AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)\right]$
$by\ (D19), (A10), FR, NV$

(D21) $\boldsymbol{UE \mid\equiv AF(2) \mid\equiv\ ID_{AF(2)}}$ $\boldsymbol{by\ (D20), BC}$

(D22) $\boldsymbol{UE \mid\equiv AF(2) \mid\equiv}$
$\boldsymbol{UE \overset{K_{AF(1)}}{\Longleftrightarrow} AF(2)}$ $\boldsymbol{by\ (D20), BC}$

(D23) $\boldsymbol{UE \mid\equiv AF(2) \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D20), BC}$

(D24) $UE \mid\equiv\ X \cdot Y \cdot G$ $by\ (D19), (A11), BC, DH$

(D25) $\boldsymbol{UE \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D20), (D24), BC}$

(D26) $AF(2) \lhd \langle n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\rangle_{SK}$

(D27) $AF(2) \mid\equiv UE \mid\sim \left[n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\right]$ $by\ (D26), (D17), MM$

(D28) $AF(2) \mid\equiv UE \mid\equiv \left[n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\right]$ $by\ (D27), (A8), FR, NV$

(D29) $\boldsymbol{AF(2) \mid\equiv UE \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D28), BC}$

5.1.2 Pull Key option

- Idealization

The idealized forms of the pull key option are as the following equations (I6)-(I8).

(I6) $UE \rightarrow AF(2)$
$\langle AID_{UE}, ID_{AF(1)}, n_1, X \cdot G, UE \overset{K}{\Leftrightarrow} AF(2)\rangle_K$

(I7) $AF(2) \rightarrow UE$
$\langle AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K}{\Leftrightarrow} AF(2)\rangle_K$

(I8) $UE \rightarrow AF(2)$
$\langle n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\rangle_{SK}$

- Assumption

(A12) $AF(2) \mid\equiv\ UE \overset{K}{\Leftrightarrow} AF(2)$
(A13) $AF(2) \mid\equiv\ \#(AID_{UE})$
(A14) $AF(2) \mid\equiv \overset{Y \cdot G}{\longrightarrow} AF(2)$
(A15) $AF(2) \mid\equiv\ \#(n_2)$
(A16) $UE \mid\equiv\ UE \overset{K}{\Leftrightarrow} AF(2)$
(A17) $UE \mid\equiv\ \#(n_1)$
(A18) $UE \mid\equiv \overset{X \cdot G}{\longrightarrow} UE$

- Goal

(G10) $AF(2) \mid\equiv\ UE \mid\equiv\ AID_{UE}$
(G11) $AF(2) \mid\equiv\ UE \mid\equiv\ UE \overset{K}{\Leftrightarrow} AF(2)$
(G12) $AF(2) \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)$
(G13) $UE \mid\equiv AF(2) \mid\equiv\ ID_{AF(2)}$
(G14) $UE \mid\equiv AF(2) \mid\equiv\ UE \overset{K}{\Leftrightarrow} AF(2)$
(G15) $UE \mid\equiv AF(2) \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)$
(G16) $UE \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)$
(G17) $AF(2) \mid\equiv\ UE \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)$

- Derivation

(D30) $AF(2) \lhd \langle AID_{UE}, ID_{AF(2)}, n_1, X \cdot G, UE \overset{K}{\Leftrightarrow} AF(2)\rangle_K$

(D31) $AF(2) \mid\equiv\ UE \mid\sim \left[AID_{UE}, ID_{AF(2)}, n_1, X \cdot G, UE \overset{K}{\Leftrightarrow} AF(2)\right]$
$by\ (D30), (A12), MM$

(D32) $AF(2) \mid\equiv\ UE \mid\equiv \left[AID_{UE}, ID_{AF(2)}, n_1, X \cdot G, UE \overset{K}{\Leftrightarrow} AF(2)\right]$
$by\ (D31), (A13), FR, NV$

(D33) $\boldsymbol{AF(2) \mid\equiv\ UE \mid\equiv\ AID_{UE}}$ $\boldsymbol{by\ (D32), BC}$

(D34) $\boldsymbol{AF(2) \mid\equiv\ UE \mid\equiv\ UE \overset{K}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D33), BC}$

(D35) $AF(2) \mid\equiv\ X \cdot Y \cdot G$ $by\ (D31), (A14), BC, DH$

(D36) $\boldsymbol{AF(2) \mid\equiv}$
$\boldsymbol{UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D32), (D35), (A15), BC}$

(D37) $UE \lhd \langle AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K}{\Leftrightarrow} AF(2)\rangle_K$

(D38) $UE \mid\equiv\ AF(2) \mid\sim \left[AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K}{\Leftrightarrow} AF(2)\right]$
$by\ (D37), (A16), MM$

(D39) $UE \mid\equiv AF(2) \mid\equiv \left[AID_{UE-AF(2)}, ID_{AF(2)}, n_1, n_2, Y \cdot G, UE \overset{SK}{\Leftrightarrow} AF(2), UE \overset{K}{\Leftrightarrow} AF(2)\right]$
$by\ (D38), (A17), FR, NV$

(D40) $\boldsymbol{UE \mid\equiv AF(2) \mid\equiv\ ID_{AF(2)}}$ $\boldsymbol{by\ (D39), BC}$

(D41) $\boldsymbol{UE \mid\equiv AF(2) \mid\equiv\ UE \overset{K}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D39), BC}$

(D42) $\boldsymbol{UE \mid\equiv AF(2) \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D39), BC}$

(D43) $UE \mid\equiv\ X \cdot Y \cdot G$ $by\ (D38), (A18), BC, DH$

(D44) $\boldsymbol{UE \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D39), (D43), BC}$

(D45) $AF(2) \lhd \langle n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\rangle_{SK}$

(D46) $AF(2) \mid\equiv UE \mid\sim \left[n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\right]$ $by\ (D45), (D36), MM$

(D47) $AF(2) \mid\equiv UE \mid\equiv \left[n_2, UE \overset{SK}{\Leftrightarrow} AF(2)\right]$ $by\ (D46), (A15), FR, NV$

(D48) $\boldsymbol{AF(2) \mid\equiv UE \mid\equiv\ UE \overset{SK}{\Leftrightarrow} AF(2)}$ $\boldsymbol{by\ (D47), BC}$

5.1.3 BAN Logic Summary

This section verifies whether the proposed protocol satisfies the security requirements by checking the above derivations of push key option and pull key option.

Lemma 1-1. Push key option in handover phase of the proposed protocol can provide mutual authentication.

Proof. The derived belief (D4) shows that AF(1) authenticates UE, and (D14) shows the AF(2) authenticates UE. UE can authenticate AF(2) in (D21). From this, push key option in handover phase of the proposed protocol can provide mutual authentication. □

Lemma 1-2. The session key SK is successfully exchanged between UE and AF(2).

Proof. According to (D23) and (D25), UE can trust the session key SK, directly and indirectly. On the other hand, AF(2) has a direct belief for the session key SK via (D17) and an indirect belief via (D29). The session key SK exchanged in push key option in handover phase of the proposed protocol is successfully exchanged between UE and AF(2). □

Lemma 1-3. Push key option in handover phase of the proposed protocol can satisfy the perfect forward secrecy.

Proof. From the above beliefs (D17) and (D25), the session key SK is generated with the ECDH ephemeral key. This ECDH ephemeral key is newly exchanged for each session, the current SK is independent of the previous SK. Thus, push key option in handover phase of the proposed protocol can satisfy the perfect forward secrecy. □

Lemma 1-4. Push key option in handover phase of the proposed protocol can support confidentiality and integrity.

Proof. The security protocol can guarantee confidentiality through the secure exchange of keys and the safety of the key itself. Based on Lemma 1-2, UE and AF(2) successfully exchange the secret key SK. Also, the secret key SK can satisfy the perfect forward secrecy as Lemma 1-3. On the other hand, the security protocol can guarantee integrity by believing the message has not been altered in transit. The derived beliefs (D17), (D23), (D25), and (D29) show that the secret key SK shared between UE and AF(2) guarantees the integrity of the message. □

Lemma 1-5. Push key option in handover phase of the proposed protocol can guarantee the guarantee the anonymity of UE.

Proof. The proposed protocol should not disclose the user identifier in the open channel. According to the derived beliefs (D4) and (D14), AF(1) and AF(2) can identify UE via anonymous identifiers $AID_{UE\text{-}AF(1)}$ and $AID_{UE\text{-}AF(2)}$, respectively. □

Theorem 1: Push key option in handover phase of the proposed protocol is secure.

Proof. According to the above Lemma 1-1 to Lemma 1-5, push key option in handover phase of the proposed protocol can provide mutual authentication, secure key exchange, perfect forward secrecy, confidentiality, and integrity. As a result, push key option in handover phase of the proposed protocol is secure since it satisfies all of the security requirements defined in '3 Preliminaries'. □

Lemma 2-1. Pull key option in handover phase of the proposed protocol can provide mutual authentication.

Proof. The derived belief (D33) shows that AF(2) authenticates UE, and (D40) shows the UE authenticates AF(2). Also, UE and AF(2) can authenticate the message via shared secret. From this, pull key option in handover phase of the proposed protocol can provide mutual authentication. □

Lemma 2-2. The session key SK is successfully exchanged between UE and AF(2).

Proof. According to (D42) and (D44), UE can trust the session key SK, directly and indirectly. On the other hand, AF(2) has a direct belief for the session key SK via (D36) and an indirect belief via (D48). The session key SK exchanged in pull key option in handover phase of the proposed protocol is successfully exchanged between UE and AF(2). □

Lemma 2-3. Pull key option in handover phase of the proposed protocol can satisfy the perfect forward secrecy.

Proof. From the above beliefs (D35) and (D43), the session key SK is generated with the ECDH ephemeral key. This ECDH ephemeral key is newly exchanged for each session, the current SK is independent of the previous SK. Thus, pull key option in handover phase of the proposed protocol can satisfy the perfect forward secrecy. □

Lemma 2-4. Pull key option in handover phase of the proposed protocol can support confidentiality and integrity.

Proof. The security protocol can guarantee confidentiality through the secure exchange of keys and the safety of the key itself. Based on Lemma 2-2, UE and AF(2) successfully exchange the secret key SK. Also, the secret key SK can satisfy the perfect forward secrecy as Lemma 2-3. On the other hand, the security protocol can guarantee integrity by believing the message has not been altered in transit. The derived beliefs (D39), and (D47) show that the secret key SK shared between UE and AF(2) guarantees the integrity of the message. □

Lemma 2-5. Pull key option in handover phase of the proposed protocol can guarantee the guarantee the anonymity of UE.

Proof. The proposed protocol should not disclose the user identifier in the open channel. From the derived belief (D33), AF(2) can identify UE via anonymous identifier $AID_{UE\text{-}AF(2)}$. □

Theorem 2: Pull key option in handover phase of the proposed protocol is secure.

Proof. Similar to the Theorem 1, pull key option in handover phase of the proposed protocol is secure since it can satisfy all of the security requirements through Lemma 2-1 to Lemma 2-5. □

5.2 Scyther

BAN Logic is a valuable tool to express and analyze authentication protocols through modal logic. However, several studies have pointed out deficiencies in security analysis using BAN Logic, such as inaccurate message representation in the Idealization step and lack of inference rules related to hash functions [22-23]. Thus, the proposed protocol is verified not only by BAN Logic but also by Scyther, an automated verification tool.

Scyther was proposed by Cas J. F. Cremers as an automated formal verification tool. It has the following verification procedure consisting of three steps: (1) modeling, (2) verification, and (3) result. (1) The target security protocol is modeled with SPDL (Security Protocol Description Language). Scyther defines a role for each communication participant in the target security protocol and specifies all messages exchanged in the protocol with SPDL expression. The protocol model written in SPDL consists of global variable declaration, protocol definition, and individual role definition. In the global variable declaration, agents, user-defined functions, macros, etc., commonly used in protocols, are declared, and protocol actions, including individual roles, are defined in the protocol definition. The protocol definition can determine not only a single protocol but also a parallel protocol as needed. The individual role definition defines the behavior of participants. It declares local variables and

includes a communication message composed of send and recv and a claim event to verify security protocols. (2) The protocol model written in SPDL is verified based on the claim events (*i.e.*, 'Alive,' 'Niagree,' 'Nisynch,' 'Weakagree,' 'Running/Commit,' and 'Secret'). Each claim checks the security attributes of the protocol model, such as authentication and confidentiality. (3) If the claim events found attacks against the security model, Scyther provides the attack flow chart. If not, Scyther displays the status of the corresponding claim events as 'OK' in the result screen. The verification results are shown in Figure 8 and Figure 9. Since the result screen displays 'OK,' both options in the proposed protocol handover phase can be seen as secure against known attacks.

Claim				Status	Comments
Push	UE	Push,UE2	Alive	Ok	No attacks within bounds.
		Push,UE3	Nisynch	Ok	No attacks within bounds.
		Push,UE4	Niagree	Ok	No attacks within bounds.
		Push,UE5	Weakagree	Ok	No attacks within bounds.
		Push,UE6	Commit AF2,g(x),YG,n1,n2	Ok	No attacks within bounds.
		Push,UE7	SKR k(UE,AF1)	Ok	No attacks within bounds.
		Push,UE8	SKR hm(h(YG,x),n1,n2)	Ok	No attacks within bounds.
	AF2	Push,AF22	Alive	Ok	No attacks within bounds.
		Push,AF23	Nisynch	Ok	No attacks within bounds.
		Push,AF24	Niagree	Ok	No attacks within bounds.
		Push,AF25	Weakagree	Ok	No attacks within bounds.
		Push,AF26	Commit UE,XG,g(y),n1,n2	Ok	No attacks within bounds.
		Push,AF27	SKR k(UE,AF1)	Ok	No attacks within bounds.
		Push,AF28	SKR hm(h(XG,y),n1,n2)	Ok	No attacks within bounds.

Figure 8. Scyther result (push key option)

Claim				Status	Comments
Pull	UE	Pull,UE2	Alive	Ok	No attacks within bounds.
		Pull,UE3	Nisynch	Ok	No attacks within bounds.
		Pull,UE4	Niagree	Ok	No attacks within bounds.
		Pull,UE5	Weakagree	Ok	No attacks within bounds.
		Pull,UE6	Commit AF2,g(x),YG,n1,n2	Ok	No attacks within bounds.
		Pull,UE7	SKR k(UE,AF1)	Ok	No attacks within bounds.
		Pull,UE8	SKR hm(h(YG,x),n1,n2)	Ok	No attacks within bounds.
	AF2	Pull,AF22	Alive	Ok	No attacks within bounds.
		Pull,AF23	Nisynch	Ok	No attacks within bounds.
		Pull,AF24	Niagree	Ok	No attacks within bounds.
		Pull,AF25	Weakagree	Ok	No attacks within bounds.
		Pull,AF26	Commit UE,XG,g(y),n1,n2	Ok	No attacks within bounds.
		Pull,AF27	SKR k(UE,AF1)	Ok	No attacks within bounds.
		Pull,AF28	SKR hm(h(XG,y),n1,n2)	Ok	No attacks within bounds.

Figure 9. Scyther result (pull key option)

6 Comparative Analysis

In this section, we compare both options in the proposed handover authentication protocol with several variants of the EAP protocol widely used in mobile networks, such as EAP-AKA [24], EAP-AKA' [25], EAP-TLS [26], and EAP-IKEv2 [27], based on the security properties (Table 5), computation overhead (Table 6), and handover latency (Figure 7).

Table 5. Comparison in terms of security properties with existing works

Security property	[24]	[25]	[26]	[27]	Push	Pull
SP1	O	O	O	O	O	O
SP2	O	O	O	O	O	O
SP3	O	O	O	O	O	O
SP4	O	O	O	O	O	O
SP5	X	X	O	O	O	O
SP6	X	X	X	X	O	O
SP7	X	X	X	X	O	O

SP1: Confidentiality; SP2: Integrity;
SP3: Mutual authentication; SP4: Secure key exchange;
SP5: Perfect forward secrecy;
SP6: Optimized handover; SP7: Anonymity;
O: Support; X: Not support

Table 6. Comparison in terms of computational overhead with existing works

Protocol	Computational overhead			
	UE	AF(1)	AF(2)	AAnF /AAAA
[24]	$9C_5$	-	-	$9C_5$
[25]	$9C_5$	-	$1C_5$	$8C_5$
[26]	$1C_1+1C_2+4C_5+1C_6$	-	$1C_1+1C_2+1C_3+1C_4+1C_5$	-
[27]	$3C_1+1C_3+1C_4+1C_5+1C_6+1C_7$	-	$3C_1+1C_3+1C_4+1C_5+1C_6+1C_7$	-
Push	$6C_5+1C_7$	$1C_5$	$5C_5+1C_7$	-
Pull	$5C_5+1C_7$	$1C_5$	$4C_5+1C_7$	-

C_1: Symmetric encryption/decryption overhead;
C_2: Asymmetric encryption/decryption overhead;
C_3: Digital signature overhead;
C_4: Signature validation overhead;
C_5: One-way HMAC overhead;
C_6: Certificate validation overhead;
C_7: ECDH operational overhead

According to Table 5, all protocols support confidentiality, integrity, mutual authentication, and secure key exchange. [24] and [25] cannot provide perfect forward secrecy. In contrast, only the proposed protocols (Push key and Pull key option) can support optimized handover and anonymity. In other

words, the proposed protocols can support all security properties from SP1 to SP7 and have a relative advantage over different protocols.

As shown in Table 6, push key and pull key options do not require symmetric encryption/decryption, asymmetric encryption/decryption, digital signature, signature validation, or certificate validation. Thus, the proposed protocols outperform [26] and [27] in terms of computational overhead. However, [24] and [25] show better performance than the proposed protocols by requiring only one-way HMAC but do not satisfy perfect forward secrecy, optimized handover, and anonymity in Table 5. The comparison results from Tables 5 and 6 show that the proposed protocols can provide secure and efficient handover between UE and AF(2).

The proposed protocol provides optimized handover compared to the existing protocol. Optimized handover minimizes latency as the user moves between AFs. By comparing the handover delay time of the proposed protocol and the EAP protocols, one of the strengths of the proposed protocol, the optimized handover, can be demonstrated. The handover latency means the execution time of the signalling message that occurs until mutual authentication between communication participants is achieved. The handover latency of the proposed protocols can be expressed as follows:

$$L_{Push} = L_{Pull} = 3 * T_{UE-AF(2)} + 2 * T_{AF(1)-AF(2)} + T_{AF(2)-AAnF} + \delta \quad \textbf{(1)}$$

$T_{AF-AAnF}$ is the transmission delay between communication participants, and δ is the processing delay for the received message. The transmission delay is given as $T_{AF-AAnF} = d * \varsigma$ where d is the distance between AF and AAnF, and ς is the average transmission delay per distance. For smooth comparison, peer, authenticator, and authentication server in EAP protocols take on UE, AF, and AAnF role in AKMA scenario, respectively. Accordingly, the handover latency of EAP protocols are given as follows:

$$L_{EAP-AKA} = L_{EAP-AKA'} = 4 * T_{UE-AF(2)} + 2 * T_{AF(1)-AAnF} + 3 * T_{AF(2)-AAnF} + \delta \quad \textbf{(2)}$$

$$L_{EAP-TLS} = 8 * T_{UE-AF(2)} + 2 * T_{AF(1)-AAnF} + 9 * T_{AF(2)-AAnF} + \delta \quad \textbf{(3)}$$

$$L_{EAP-IKEv2} = 6 * T_{UE-AF(2)} + 2 * T_{AF(1)-AAnF} + 7 * T_{AF(2)-AAnF} + \delta \quad \textbf{(4)}$$

Table 7 shows the numerical simulation parameters proposed in [28] and [29] to calculate the handover latency. Figure 10 shows the handover latency of EAP protocols and the proposed protocol with the numerical simulation parameters of Table 7. Since the EAP protocols execute the full authentication procedure for every authentication, the handover latency is higher than the proposed protocol. [24] and [25] have the same signaling message sequence, so they incur the same cost, and [26], which includes a relatively large number of procedures, has the highest handover latency.

Table 7. Numerical simulation parameters

Parameters	Values
$T_{UE-AF(2)}$	1 ms
d	20 – 100 km
ς	0.05 ms/km
δ	9.5 ms

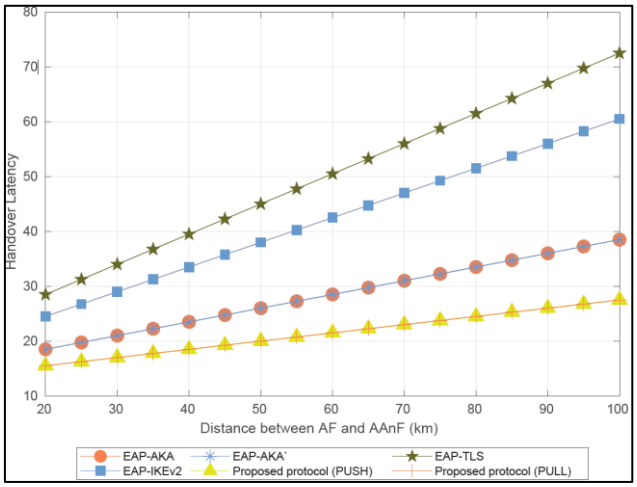


Figure 10. Handover latency vs Distance

7 Conclusion

With the advent of 5G advanced technologies, users can use various innovative multimedia services anytime, anywhere, in real-time. Protecting the channel between the user and the AF, a distributed unit for NetApp, is essential to provide a safe application use environment. In particular, it is necessary to consider that UE frequently moves among AFs in 5G MEC environments because its subscribing service, a NetAPP, is composed of distributed AFs. Existing studies, however, have fallen short of taking this into account and so are unable to provide the necessary security measures. Thus, in this article, we put forward a novel security protocol that seamlessly provides a multimedia service in the 5G MEC environments by satisfying a range of security properties such as mutual authentication, secure key exchange, perfect forward secrecy, confidentiality, integrity, and anonymity. In addition, the protocol provides push-key and pull-key options for optimized secure handover mechanisms that existing standards did not provide. The formal security verification of both options of the handover scheme proved the protocol indeed satisfies the stated requirements. The comparative analysis of the proposed protocol with EAP variants further confirmed that the proposed protocol is efficient in terms of computational overhead. The handover procedure can be improved even more with artificial intelligence-based movement route prediction technology, which is reserved for future study.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-01019, Development of eSIM security platform technology for edge devices to expand the eSIM ecosystem).

Abbreviations

Abbreviation	Description
AAnF	AKMA Anchor Function
AF	Application Function
AKA	Authentication and Key Aggrement
AKMA	Authentication and Key Management for Application
AMF	Access and Mobility Management
AUSF	AUthentication Server Function
A-KID	AKMA Key Identifier
BEST	Battery Efficient Security for very low Throughput MTC devices
BSF	Bootstrapping Server Function
CAGR	Compound Annual Growth Rate
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman key exchange
ECDHE	ECDH Ephemeral
GBA	Generic Bootstrapping Architecture
HPLMN	Home Public Land Mobile Network
HSS	Home Subscriber Server
HTTP	Hyper-Text Transfer Protocol
IPSec	Internet Protocol Security
JSON	JavaScript Object Notation
MEC	Multi-access Edge Computing
MTC	Machine Type Communication
MVNO	Mobile Virtual Network Operator
NAF	Network Application Function
NEF	Network Exposure Function
NetApp	Network Application
NF	Network Function
NRF	Network function Repository Function
NSSF	Network Slice Selection Function
PCF	Policy Control Function
PCRF	Policy and Charging Rule Function
REST	REpresentational State Transfer
SBA	Service-Based Architecture
SBI	Service-Based Interface
SCP	Service Communication Proxy
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
TLS	Transport Layer Security
UDM	Unified Data Management
UDR	Unified Data Repository
UPF	User Plane Function
3GPP	Third Generation Partnership Project
5G	Fifth-Generation technology standard for broadband cellular network

References

[1] Maximize Market Research, *IP Multimedia Subsystem Market: Global Industry Analysis and Forecast (2021-2027) Trends, Statistics, Dynamics, Segmentation by Component, Operator and Region*, Maximize Market Research Report ID 12949, December, 2021.

[2] V. A. Cunha, E. Silva, M. B. Carvalho, D. Corujo, J. P. Barraca, D. Gomes, L. Z. Granville, R. L. Aguiar, Network slicing security: Challenges and directions, *Internet Technology Letters*, Vol. 2, No. 5, Article No. e125, September/October, 2019.

[3] X. Huang, V. Tsiatsis, A. Palanigounder, L. Su, B. Yang, 5G Authentication and Key Management for Applications, *IEEE Communications Standards Magazine*, Vol. 5, No. 2, pp. 142-148, June, 2021.

[4] 3GPP, *Authentication and Key Management for Applications (AKMA) based on 3GPP credentials in the 5G System (5GS)*, 3GPP TS 33.535 v17.6.0, July, 2022.

[5] 3GPP, *Generic Bootstrapping Architecture (GBA) (Release 17)*, 3GPP TS 33.220, Version 17.3.0, July, 2022.

[6] 3GPP, *Battery Efficient Security for very low Throughput Machine Type Communication (MTC) devices (BEST) (Release 16)*, document 3GPP TS 33.163, Version 17.0.0, December, 2021.

[7] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, C. Su, Blockchain and PUF-Based Lightweight Authentication Protocol for Wireless Medical Sensor Networks, *IEEE Internet of Things Journal*, Vol. 9, No. 11, pp. 8883-8891, June, 2022.

[8] W. Wang, Z. Han, M. Alazab, T.R. Gadekallu, X. Zhou, C. Su, Ultra Super Fast Authentication Protocol for Electric Vehicle Charging Using Extended Chaotic Maps, *IEEE Transactions on Industry Applications*, Vol. 58, No. 5, pp. 5616-5623, September/October, 2022.

[9] M. Burrows, M. Abadi, R. M. Needham, A logic of authentication, *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, Vol. 426, No. 1871, pp. 233-271, Dec. 1989.

[10] C. J. F. Cremers, The Scyther Tool: Verification, falsification, and analysis of security protocols, *International conference on computer aided verification*, Princeton, NJ, USA, 2008, pp. 414-418.

[11] 3GPP, *Study on security aspects of the 5G Service Based Architecture (SBA)*, 3GPP TR 33.855, Version 16.1.0, September, 2020.

[12] C. Jost, *Security for 5G Service-Based Architecture: What you need to know*, Ericsson Blog, August, 2020.

[13] 3GPP, *Security architecture and procedures for 5G system*, 3GPP TS 33.501 v17.6.0, July, 2022.

[14] T. Dierks, E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, IETF RFC 5246, August, 2008.

[15] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, IETF RFC 8446, August, 2018.

[16] D. Hardt, *The OAuth 2.0 Authorization Framework*, IETF RFC 6749, October, 2012.

[17] 3GPP, *Study on authentication and key management for applications; based on 3GPP credential in 5G*, 3GPP TR 33.835, Version 16.1.0, July 2020.

[18] D. Dolev, A. Yao, On the security of public key protocols, *IEEE Transactions on Information Theory*, Vol. 29, No. 2, pp. 198-208, March, 1983.

[19] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, Vol. 48, No. 177, pp. 203-209, January, 1987.

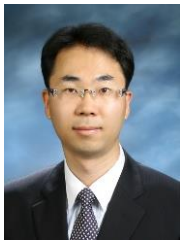
[20] K. Lauter, The advantages of elliptic curve cryptography for wireless security, *IEEE Wireless Communications*, Vol. 11, No. 1, pp. 62-67, February, 2004.

[21] A. DeKok, *The Network Access Identifier*, IETF RFC 7542, May, 2015.

[22] C. Boyd, W. Mao, On a limitation of BAN logic, *Workshop on the Theory and Application of Cryptographic Techniques*, Lofthus, Norway, 1993, pp. 240-247.
[23] C. A. Meadows, Formal verification of cryptographic protocols: A survey, *International Conference on the Theory and Application of Cryptology (ASIACRYOPT)*, Wollongong, Australia, 1994, pp. 133-150.
[24] J. Arkko, H. Haverinen, *Extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA)*, IETF RFC 4187, January, 2006.
[25] J. Arkko, V. Lehtovirta, P. Eronen, *Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')*, IETF RFC 5448. May, 2009.
[26] D. Simon, B. Aboba, R. Hurst, *The EAP-TLS authentication protocol*, IETF RFC 5216, March, 2008.
[27] H. Tschofenig, D. Kroeselberg, A. Pashalidis, Y. Ohba, F. Bersani, *The extensible authentication protocol-Internet key exchange protocol version 2 (EAP-IKEv2) method*, IETF RFC 5106 (Experimental), February, 2008.
[28] G. Brown, *New transport network architecture for 5G RAN*, Fujitsu, Kanagawa, Japan, White Paper, 2018.
[29] Samsung, *4G-5G Interworking: RAN-Level CN-Level Interworking*, June, 2017.

Biographies

Jiyoon Kim received the Ph.D. degrees in information security from Soonchunhyang University, Asan, South Korea, in 2022. He is currently working as an Assistant Professor with the School of Computer Science, Gyeongsang National University, Jinju, South Korea. His main research interests include 5G/6G security and formal security analysis.



Dong-Guk Han received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in engineering in information security from Korea University, Seoul, Republic of Korea, in 1999, 2002, and 2005, respectively. He was a Postdoctoral Researcher at Future University Hakodate, Hokkaido, Japan. After finishing his doctoral course, he was then an Exchange Student with the Department of Computer Science and Communication Engineering, Kyushu University, Japan, from April 2004 to March 2005. From 2006 to 2009, he was a Senior Researcher at the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea. He is currently working as a Professor with the Department of Information Security, Cryptology, Mathematics, Kookmin University, Seoul. He is a member of KIISC, IEEK, and IACR.

Ilsun You received the MS and PhD degrees in computer science from Dankook University, Seoul, Korea, in 1997 and 2002, respectively. He received his second PhD degree from Kyushu University, Japan, in 2012. Now, he is a full professor at the Department of Information Security, Cryptology, and Mathematics, Kookmin University. He has served or is currently serving as a Steering Chair, General Chair or a Program Chair of international conferences and symposiums such as MobiSec'16-22, WISA'19-20, 22, ProvSec'18, ACM MIST'15-17 and so forth. Dr. YOU is the EiC of Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) and Journal of Internet Services and Information Security (JISIS) while serving as an associate EiC of Intelligent Automation & Soft Computing (IASC). He is in the Editorial Board for Information Sciences, International Journal of Intelligent Systems, IEEE Access, International Journal of Ad Hoc and Ubiquitous Computing, Computing and Informatics, and Journal of High Speed Networks. Especially, he has focused on 5/6G security, security for wireless networks & mobile internet, IoT/CPS security and so forth while publishing more than 180 papers in these areas. He is a Fellow of the IET and a Senior member of the IEEE.