Formal verification of secondary authentication protocol for 5G secondary authentication

Edris, Ed Kamya Kiyemba, Aiash, Mahdi, Loo, Jonathan ORCID logoORCID: https://orcid.org/0000-0002-2197-8126 and Alhakeem, Mohammad Shadi (2021) Formal verification of secondary authentication protocol for 5G secondary authentication. International Journal of Security and Networks, 16 (4). pp. 223-234. ISSN 1747-8405

http://dx.doi.org/10.1504/IJSN.2021.10043015

Formal verification of secondary authentication protocol for 5G secondary authentication

Ed Kamya Kiyemba Edris* and Mahdi Aiash

School of Science and Technology,
Middlesex University,
London, UK
Email: ee351@live.mdx.ac.uk
Email: m.aiash@mdx.ac.uk
*Corresponding author

Jonathan Kok-Keong Loo

School of Computer and Engineering,
University of West London,
London, UK
Email: jonathan.loo@uwl.ac.uk

Mohammad Shadi Alhakeem

Naif Arab University for Security Sciences,
Riyadh, Saudi Arabia
Email: malhakeem@nauss.edu.sa

Abstract: The fifth-generation mobile network (5G) will enable interconnectivity between the home network (HN) and data network (DN) whereby mobile users with their user equipment (UE) will be able to access services provided by external service providers (SP) seamlessly. The mobile user and SP will rely on security assurances provided by authentication protocols used. For 5G, primary authentication between the UE and the HN has been defined and specified by the Third Generation Partnership Project (3GPP) while the secondary authentication has also been defined but not specified. 3GPP recommends the extensible authentication protocol (EAP) framework for secondary authentication between the UE and the SP. However, the secondary authentication methods have not been formally verified, so this paper proposes a secondary authentication protocol (SAP) for service authentication and provides a comprehensive formal analysis using ProVerif a security protocol verifier. Finally, it conducts a security analysis on the protocol's security properties.

Keywords: 5G; secondary authentication; security protocol; services; formal methods; ProVerif; applied Pi calculus.

Reference to this paper should be made as follows: Edris, E.K.K., Aiash, M., Loo, J.K-K. and Alhakeem, M.S. (2021) 'Formal verification of secondary authentication protocol for 5G secondary authentication', *Int. J. Security and Networks*, Vol. 16, No. 4, pp.223–234.

Biographical notes: Ed Kamya Kiyemba Edris received his Diploma in Network Engineering and Telecommunication Systems from the St. Patrick College in London, UK in 2015, BSc in Computer Networks and MSc in Network Security and Penetration Testing from the Middlesex University in London, UK in 2016 and 2017, respectively. He is currently working towards his PhD in Computer Communication and Engineering from the Middlesex University London, UK, also teaches undergraduate and postgraduate programs at the Middlesex University and University of Hertfordshire. He has published in IEEE and ISCT. His research interests include computer networks, future internet, information centric networking, cloud computing, wireless/mobile communications, cyber-physical systems, IoT, AI network, network security, cyber and information security. He is a student member of IEEE and IET.

Mahdi Aiash is a computer security researcher and practitioner with industrial, academic, and research experience. He has over ten years of experience in pentesting, R&D, and incident response. He has produced in excess of 75 research items and he currently maintains nearly a dozen certifications in system and network security. He is a recipient of three research grants to develop new cyber security solutions and protocols. In addition to his academic activities, he has delivered cyber projects nationally and internationally for the private and public sectors and conducted workshops and corporate training around the world apart from his speaking engagements.

Jonathan Kok-Keong Loo received his MSc in Electronics and PhD in Electronics and Communications from the University of Hertfordshire, Hertfordshire, UK, in 1998 and 2003, respectively. Between 2003 and 2010, he was a Lecturer of Multimedia Communications in the School of Engineering and Design, Brunel University, Uxbridge, UK. Between June 2010 and May 2017, he was an Associate Professor of Communication Networks with the School of Science and Technology, Middlesex University, London, UK. Since June 2017, he is the Chair Professor of Computing and Communication Engineering in the School of Computing and Engineering, University of West London, London. His research interests include information centric networking, cloud computing, mobile networks and protocols, network security, wireless communications, IoT/cyber-physical systems, embedded systems, cybersecurity, forensic and applied AI. He has co-authored more than 240 journal and conference papers. He has been an associate editor for the *Wiley International Journal of Communication*.

Mohammad Shadi Alhakeem received his PhD in Computer Science from the Technical University of Berlin, Germany, in 2010. Between 2010 and 2014, he was a Lecturer of Informatics and Communication Engineering at Syrian universities. Between 2014 and 2017 he was a postdoctoral researcher at the Department of Operating and Communication Systems, Technical University of Berlin, Germany. Since September 2017, he is an Assistant Professor of Cyber Security and Digital Forensics at the Naif Arab University for Security Sciences. He received many grants from universities in UK and the European Union for his research in cloud computing and cyber-physical systems, and he published many peer reviewed papers in IEEE and Springer. His areas of research include computer and network security, cloud computing, internet of things and digital forensics.

1 Introduction

Service provisioning from tactile internet, internet of things (IoT) and multiple service providers (SPs) will be supported in fifth generation mobile network (5G) in form network slices. 5G will also enable interconnectivity between the home network (HN) and data network (DN) which will be providing services that are not available from the mobile network operator (MNO), also referred to as third-party SP. The mobile end users with their user equipment (UE) will be able to initiate network access and service requests through new generation radio access network (ngRAN) as the access point (AP) seamlessly and securely. The services provided by third-party SP are accessed via DN function as defined in 5G system architecture (3GPP, 2020b). Security mechanisms are required to secure the access of network and services at all levels of the network. The Third Generation Partnership Project (3GPP) defined primary and secondary authentication procedures in security architecture (3GPP, 2020a) to support 5G objectives. Primary authentication will be used to authenticate the UE to the HN, while secondary authentication will be used to authenticate the UE to the SP. Even though 3GPP has defined secondary authentication but has not specified the details and how this authentication method should be implemented.

Like in primary authentication the end user and MNO expect security assurances from the secondary authentication method properties such as trust, authentication, data confidentiality and data integrity when communicating to the third-party SP. After a successful primary authentication, the UE will be able to perform an optional secondary authentication if required by the SP but 5G security context such as keys and ID should not be shared with DN, so there is need of a security procedure that can provide security for UE, HN and SP without exposing the 5G security context from primary authentication procedure. 5G security standard (3GPP, 2020a) does not specify the security parameters but states that extensible authentication protocol (EAP) framework is the preferred method for the secondary authentication between the UE and the SP, an external DN.

This paper proposes secondary authentication protocol (SAP)-AKA to provide security guarantees for service authentication between the UE and SP. The protocol uses EAP framework (Vollbrecht et al., 2004) aligning it with 5G standard. We formally analyse and verify the proposed protocol using ProVerif (Blanchet et al., 2020). To the best of our knowledge, there is no related work on 5G service authentication in a non-3GPP system and no formal analysis of the secondary authentication method for 5G network.

In this paper, we interpret the specification of secondary authentication and set the security properties based on 3GPP standard. We propose a service authentication protocol based on EAP framework that provides the UE with external ID (EID) and a session key that can be used in service authentication and authorisation of the UE to the SP, from 3GPP to non-3GPP system. We model the SAP-AKA protocol with symbolic modelling using ProVerif and applied calculus. Furthermore, we conduct a formal and comprehensive security evaluation of SAP-AKA security properties to identify the security requirements of the protocol based on two sets of security taxonomies. We finally present our security consideration, as our protocol modelling can serve as a basis for modelling and analysing for next generation service authentication protocols.

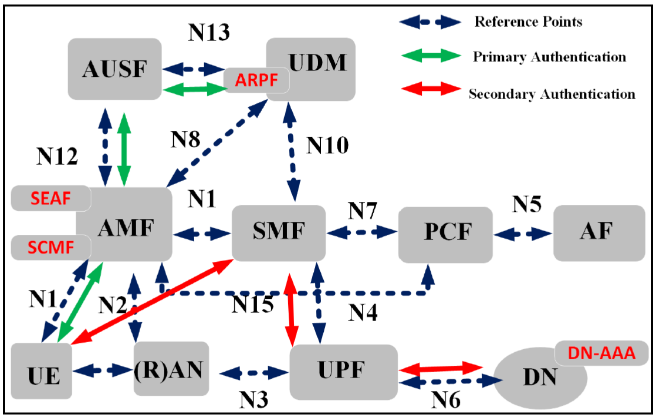
The rest of the paper is structured as follows. Section 2 discusses the related work on secondary authentication procedure. While Section 3 presents our proposed SAPAKA protocol. Section 4 presents the modelling of SAPAKA protocol. The verification of SAP-AKA protocol is presented in Section 5. In Section 6, a security analysis is presented. Finally, this paper concludes in Section 7.

2 Related work

As mentioned in the introduction, 3GPP specified the primary authentication in detail but not the secondary authentication and most of the related work focuses on the 5G network access security. In this work, we focus on secondary authentication and service security between the UE and SP.

The 5G standard (3GPP, 2020a) recommends that secondary authentication should be based on EAP framework which is an authentication framework that supports authentication method defined under RFC 3748 (Vollbrecht et al., 2004). It runs directly over data link layers without requiring an Internet protocol (IP) address and used on dedicated links, wired and wireless links for flexibility. An EAP-AKA protocol was developed by 3GPP and verified by the EAP WG in RFC 4187 (Arkko et al., 2009). It was later specified as an EAP mechanism for authentication and session key distribution that uses AKA mechanism for 3rd generation mobile networks Universal Mobile Telecommunications System (UMTS).

Figure 1 5G security architecture (see online version for colours)



The AKA was based on symmetric keys, and typically runs in a universal subscriber identity module (USIM). EAP-AKA included options for identity (ID) privacy support, result indications, and fast re-authentication procedure. The RFC 4187 made the use of AKA method for primary authentication possible within EAP framework, later improved in 5448 (Arkko et al., 2018) with a new EAP method, EAP-AKA'. The changes included a new key derivation function that binds the derived keys with name of the access network hence protection from binding down attacks. In addition, the EAP-AKA' can be used for primary authentication to gain network access to 5G and non-3GPP access specified in TS 33.501 (3GPP, 2020a). The EAP-AKA' uses 'cipher key (CK)' and 'integrity key (IK)' as specified in TS 33.402 (3GPP, 2018b) and updated the hash function from secure hash algorithm (SHA)-1 to SHA-256 and hash message authentication code (HMAC) to HMAC-SHA-256.

2.1 Architecture overview

The 5G security architecture (3GPP, 2020a) illustrates the functions that participate in primary and secondary authentication as shown in Figure 1, the SP plays the role of DN. Some of the 5G security entitles are:

- Security anchor function (SEAF): A security anchor that acts as middleman during primary authentication. It interacts with the authentication security function (AUSF) to authenticate the UE to the HN.
- AUSF: An authentication server residing in MNO's HN. It interacts with the SEAF to authenticate UE.
- Authentication credential repository and processing function (ARPF): A credential repository residing in a secure environment in an MNO's HN. It stores the long-term security credentials for UE authentication and executes any cryptographic algorithms that use those security credentials as input.
- SMF: Is responsible for interacting with the data plane, creating updating, and removing protocol data unit (PDU) sessions and managing session context with the user plane function (UPF). It acts as middleman during the secondary authentication between UE and SP.
- Access management function (AMF): It manages connection or mobility management and then forwards session management requirements to the SMF.
- Unified data management (UDM): It manages user data, together with ARPF, they support the build-up of a unified authentication framework for different access technologies and enable security context sharing.
- UPF: It provides the interconnectivity between the HN mobile infrastructure and the DN.

2.2 Formal methods

Formal methods and automated verification have been applied to authentication protocols for mobile networks to assess security properties (Basin et al., 2018; Aiash, 2013; Edris et al., 2020a), to provide strong security guarantees. Security protocols properties are very challenging for most verification techniques and tools. This is due to the use of cryptographic primitive and its algebraic properties are tricky for symbolic reasoning (Basin et al., 2018) hence certain tools and manual proof checks are not suitable. There are many automated verification tools that can be used for protocol analysis such as automated validation of internet security protocols and applications (AVISPA) (Armando et al., 2005) and ProVerif (Blanchet et al., 2020).

Table 1 Core language: syntax and informal semantics

a, b, c, k, s	Name
x, y, z	Variable
M, N ::=	Terms
h(D1, …, Dn)	Function application
f(M1, …, Mn)	Constructor application
D ::=	Expressions
Fail	Failure
P, Q ::=	Processes
Out(N, M); P	Output
In(N, x: T); P	Input
!P	!P replication
0	Nil
P|Q	Parallel composition
New a: T; P	Restriction
Let x: T = D in P else Q	Expression evaluation
If M then P else Q	Conditional

ProVerif (Blanchet et al., 2020) is an automatic tool for analysing security protocols, with Dolev-Yao (DY) as the adversarial model and it supports equational theories defined by users and as well as enabling the verification of security properties. It supports underlying theory of abstraction, but it may also lead to false attacks. The equational theories that ProVerif can handle are defined by the user and are enough to model exclusive OR (XOR) (Küsters and Truderung, 2009). It uses applied Pi calculus (Ryan and Smyth, 2011) as a formal language for describing and modelling security protocols. The syntax is paired with a formal semantics to enable reasoning about protocols. It also supports a variety of cryptographic primitives, modelled by equations and rewrite rules. In addition, it also takes the security properties such as authentication, secrecy along with observational equivalence properties to be proved as input. The information is translated into internal representation of the protocol which makes some abstraction that are crucial to an unbounded number of sessions. Cryptographic primitives are modelled as functions, while messages are represented by terms built over an infinite set of names like *a*, *b*, *c*, …, then an infinite set of variables like *x*, *y*, *z*, … and a finite set of function symbols like *f*1, …, *fn*. A set of reduction rules describes how applying function symbols to terms is affected. The syntax and grammar of ProVerif process language is shown in Table 1 and more details can found in Blanchet et al. (2020). For those reasons, we find ProVerif a suitable tool for our analysis. It has been used to formally check security properties guarantees of authentication protocols in Edris et al. (2020a) and Zhang et al. (2020).

As mentioned earlier, for network access security, the UE will use primary authentication to authenticate to the HN and while secondary authentication will be used for service authentication to the SP as defined by 3GPP. Our proposed SAP is explained in the next section.

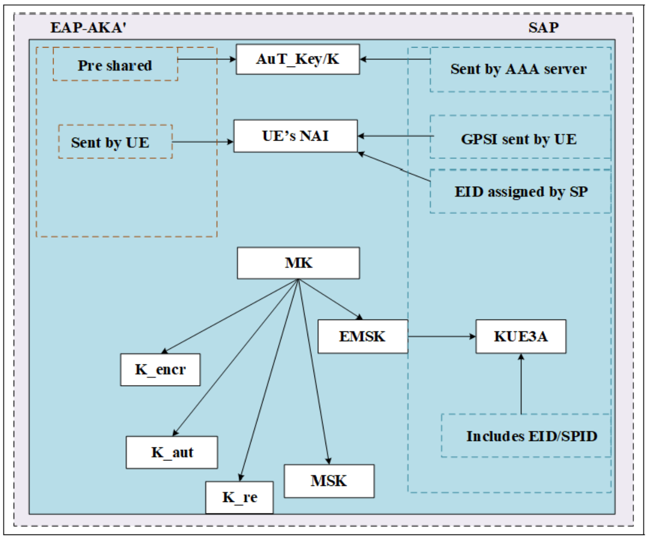
3 Our proposed security protocol

In this section, we give an overview of the proposed protocol, security properties, and how it aligns with 3GPP standard. We propose a SAP-AKA that leverages on EAP framework (Vollbrecht et al., 2004) as recommended in by 3GPP (2020a). This protocol uses the security parameters and EAP-AKA key derivation function (3GPP, 2018b). It is an optional authentication that must be initiated by third-party SP when UE requests its services. To access the services from the third-party SP the UE must get authenticated by SP via session management function (SMF) of the HN which acts as pass through authenticator. This protocol intends to provide authentication and key derivation between the UE and the SP. It achieves mutual authentication and key agreement and implicit authentication with HN.

3.1 Problem definition

As mentioned earlier, 3GPP recommends that EAP framework should be used as the secondary authentication method in a fully active exposure scenario to external networks, however the EAP has some limitations in achieving this objective. There is a restriction on using 5G security context such as keys and IDs outside the HN with non-3GPP access networks and the EAP' framework requires the authentication key *AUT_Key* to be pre-shared between the UE and AAA server before the run of the protocol which raise security problems. The *AUT_Key* is used to derive CK'/IK' and other following keys.

Figure 2 5G EAP-SAP problem definition (see online version for colours)



The IDs used in 5G primary authentication are not allowed to be used outside the HN that is why the SAP protocol uses a generic public subscription identifier (*GPSI*) for the UE a publicly know ID which is later replaced by the *EID* created by SP and securely sent to the UE. The derived keys as part of EAP are used in the following way. The *K_encr* is used to encrypt *AT_ENCR_DATA* attribute such pseudonym IDs (identity privacy), *K_aut* is used to encrypt the *AT_MAC* attribute and *K_re* only used in re-authentication process as per 3GPP and EAP specifications. While the MSK is used to protect the EAP-AKA packets for non-3GPP access interworking function (N3IWF) and the EMSK is used in derivation of 3GPP related access keys to secure the HN.

Therefore, SAP protocol is intended to provide mutual authentication, a session key and EID to secure communication between the UE and SP. The SAP solves the issue of not sharing primary authentication keys and security context with an external AAA server as per 5G specification shown in Figure 2 by using symmetric and asymmetric cryptography. After a successful run the SAP protocol K_{UE3A} is generated to be used by UE and SP to secure their communication and *EID* is created and assigned to the UE as its permanent ID. The EID and SPID are used as inputs in the derivation of the session key to bind both the UE and SP to the session.

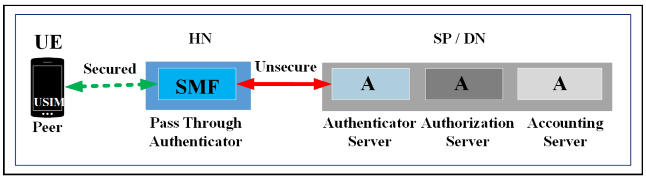
3.2 Security assumption

Most of the assumptions are based on the specifications in TS 33.501 (3GPP, 2020a), TS 33.402 (3GPP, 2018b) and RFC 5448 (Arkko et al., 2018). If the channel between the UE and the SP is assumed to be secure it should provide confidentiality, integrity, authenticity, and replay attack protection. This channel is subject to eavesdropping by passive attackers and manipulation, interception, and injection of messages by active attackers. It is also assumed that cryptographic primitives such as the functions $f1$, $f2$ provide integrity as message authentication code (MAC) and $f3$, $f4$, $f5$ provide integrity and confidentiality as cipher, integrity and anonymity respectively as defined in 3GPP (2018a) and hash functions provide integrity and confidentiality using derivation of keys and HMAC (Vollbrecht et al., 2004) while the individual messages use their own cryptographic protections directly. We also assume that the attacker may have genuine USIMs under its control, hence the attacker can access all secret values stored in the USIM, such as ID and keys. 'EAPAKA' has no ciphersuite negotiation mechanisms but it has a negotiation mechanism for selecting the key derivation functions. The security properties provided by SHA-256 such as mutual authentication, confidentiality, cryptographic binding, and session independence are as good as those of EAP-AKA. We also assume that SHA-256 behaves as a pseudo-random function, an attacker also cannot calculate the pre-shared secret from any keys by any practically feasible means. EAP-AKA' uses different identifiers to identify the authenticating UE. The protocol key strength prevents brute force attacks but does not provide channel binding.

The proposed protocol security entities as shown in Figure 3 are:

- UE: A mobile terminal containing the USIM that has cryptographic capabilities such as algorithms, encryption, MAC. It acts as the peer.
- H-SMF: The HN SMF is a 5G function that communicates with the HN-AAA and EN entities such SP authenticator, it acts as pass through authenticator.
- SP-AAA: It hosts the authentication authorisation accounting (AAA) servers owned by SP. The SP is also part of the transaction; it grants authority, issues access/fresh tokens to be used by the UE to access the service and exchanges *GPSI* with *EID*. It acts as the authentication server.

Figure 3 5G EAP-AKA' entities (see online version for colours)



3.3 Overview of SAP-AKA

In 5G, the UE must register with the network, perform primary authentication via AUSF and establishes a NAS security context with the AMF as precondition. After a successful primary authentication, the UE will have agreed with HN a session key K_{SEAF} that is used to communicate with SEAF in the serving network (SN). K_{SEAF} is also used to derive other keys that are used by UE to communicate securely other functions in 5G. One of the derived keys is K_{AMF} used to secure communication between the UE and AMF. With its network access credentials and non-access stratum (NAS) security context, the user via the UE initiates to establishment of PDU service session with a SP for particular service based on content name, domain name of SP or single network slice selection assistance information (S-NSSAI) in 5G network, by sending service session ID (*SID*) and SP identifier (*SPID*) as the packet data network (PDN). The SMF obtains subscription data from the UDM for the given subscriber permanent identity (SUPI) obtained from the AMF, the UE is assigned a *GPSI* as its ID for outside HN use. The communication between the UE and the SMF via AMF is protected by $K_{AMF.}$ The SMF checks whether the UE request is compliant with the user subscription, local policies, and external policies in relations to the SP. If not, the SMF may reject UE's request. The SMF may also check whether the UE has been authenticated or authorised by the same SP before hence updating the UDM/ARPF database.

The SMF redirects the UE to the SP to initiate the secondary authentication as initial step for service authorisation, the assigned SP entity will check if the UE is registered and authenticated with SP before, if not then it will initiate a secondary authentication. If the UE is already registered, it will continue with service authorisation procedure (Edris et al., 2020b). In this protocol, for the authentication between the UE and the SP, we adopt the EAP framework (Vollbrecht et al., 2004) and 3GPP TS TS33.501 specification (3GPP, 2020a) with a few modifications to suit service level authentication in a non-3GPP system. This AKA was specified to be used as secondary authentication for external DN. We assume that UE would have been registered with HN's MNO, while the SP and MNO would have service subscription of the UE. If the MNO and the SP are different then they should have interoperator service agreement for the UE.

After a successful authentication, the UE will be assigned a permanent user ID *EID* by SP-AAA, different from the one used in primary authentication that will be used in the request of access to services. The ID will either be derived from IP address or pseudo randomised name given by the authenticator. At the same time session key K_{UE3A} is generated to secure service authorisation process as one described in Edris et al. (2020b). The HN supports the procedure but it is controlled by the SP-AAA, it provides mutual authentication and key agreement between UE and SP without using the security context from primary authentication.

3.4 Security requirements

The desired security properties for SAP-AKA protocol are secrecy, confidentiality, integrity, authenticity, and privacy (3GPP, 2020a). The UE must have the assurance that authentication can only be successful with SP authorised by their HN. The UE shall authenticate SP with the network access identifier (NAI) through mutual authentication and key confirmation. Formally, a HN must obtain weak agreement on SP with its UE after key confirmation. The SP shall be able to authenticate the UE with *GPSI* and pre-shared information with HN in the registration process. SAP-AKA should ensure the secrecy of K_{UE3A} and ensure that no other party has knowledge of the session key. It should ensure that even without using the security context from the primary authentication, the SAP-AKA can secure communication between UE and SP. In addition, the same key K_{UE3A} should never be established twice. Since no security context is shared with third-party SP compromising primary authentication should not compromise the secondary authentication. The subscription privacy should also be ensured by providing confidentiality, anonymity, and untraceability.

Table 2 *AT_KDF* and *AT_KDF_INPUT* parameters

Key	*Input*
MK	*KDF*(*PRF′*(*IK′*|*CK′*; ‘*SAP*’’|*Identity*))
K_encr	*KDF*(*MK*[0…127])
K_aut	*KDF*(*MK*[128…383])
K_re	*KDF*(*MK*[384…639])
MSK	*KDF*(*MK*[640…1,151])
EMSK	*KDF*(*MK*[1,152…1,663])
K_{SEAF}	*KDF*(*EMSK*, *SNN*)
K_{AMF}	*KDF*(*KSEAF*, *SUPI*, *ABBA*)
K_{UE3A}	*KDF*(*EMSK*, (*EID*, *SPID*))

The security properties are informally defined before the formalisation of the protocol, we adopt the taxonomies in Lowe (1997) and Menezes et al. (2001) for precise formal analysis, referred to as set 1 and set 2 respectively in this paper. In set 1, the security properties are specified from an agent A's point of view, with four levels defined between two agents A and B, aliveness, weak agreement, non-injective agreement, and injective agreement. While in set 2 the security protocol should meet the following security properties; mutual entity authentication, mutual key Authentication, mutual key confirmation, key freshness, unknown-key share, and key compromise impersonation resilience.

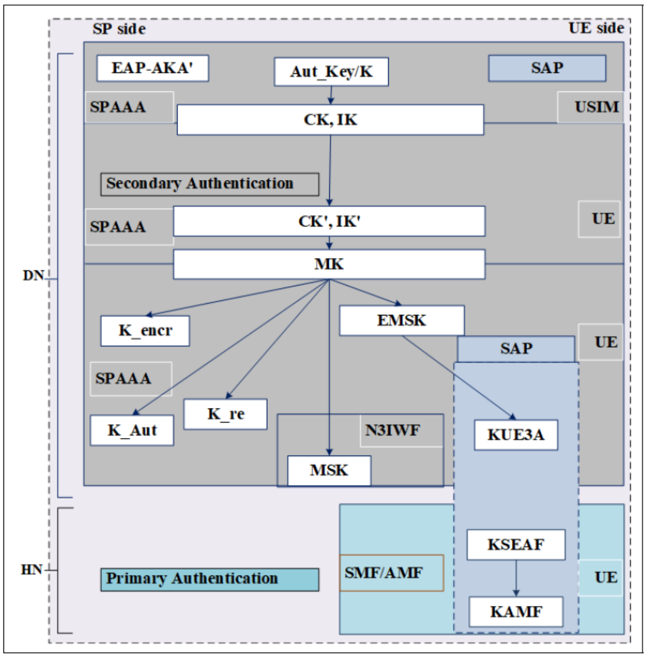
3.5 Keys derivation and hierarchy

The key derivation is performed according to EAP framework (Arkko et al., 2018) with the *at_kdf* input parameters as inputs (3GPP, 2018b). The UE and the authentication server compute CK', IK' keys which are used together with PRF', SAP and identity as key derivation inputs *at_kdf_input*. PRF' is a pseudorandom function, SAP is string indicating the type of protocol and identity is the UE identity used derive a master key (MK). The *MK* is used to derive *K_encr*, *K_aut*, *K_re*, master session key (*MSK*) and extended master session key (*EMSK*) as shown in Table 2. The *K_encr* is used for *AT_ENCR_DATA* and *K_aut* for *AT_MAC* attributes respectively while the *K_re* is used during the re-authentication process if required. The *MSK* and *EMSK* are derived after a successful EAP AKA challenge response run (Arkko et al., 2018) for non-trusted and trusted non-3GPP access networks, respectively. After a successful secondary authentication process using SAP-AKA protocol, the *EMSK* key, UE and SPIDs are used as input parameters *at_kdf_input* with key derivation function *KDF*(*EMSK*, (*EID*, *SPID*)) in deriving K_{UE3A} to secure communication UE and SP in next stage of service authorisation. The MSK is used derived keys for non-trusted N3IWF. In addition, K_{AMF} is used to secure communication between UE and SMF provided by AMF, derived from K_{SEAF} using SUPI associated with NAI and anti-bidding down between architectures (ABBA) parameters for forward compatibility as *at_kdf_input* parameters *KDF*(*KSEAF*, *SUPI*, *ABBA*) (3GPP, 2020a) during primary authentication run. The K_{UE3A} is derived after a successful SAP protocol run between the UE and the SP. The key derivation and hierarchy is shown in Figure 4, where [0…*n*] denotes the substring from bit 0 to *n* used in the key derivation (3GPP, 2020a).

4 Modelling of SAP-AKA protocol

In this section, we model the proposed protocol and present the message exchange between the involved parties. The notations and values used for authentication vectors (AV) includes random nonce *RAND* as challenge, *AUTN* as authentication token to prove the challenge's freshness and authenticity, illustrated in Table 3. The authentication and verification of the AV is controlled by external AAA servers. The SAP-AKA protocol consists of three entities, i.e., UE, SMF and SP-AAA. Whereby, the SMF acts as pass through authenticator and it also processes the UE initial services request to check with HN UDM via AMF if the UE subscription credentials are valid and which SP should the UE be redirected to. It relies on the external SPAAA server to authenticate and authorise the UE's request for the establishment of service sessions. The cryptographic function and scheme are based on elliptic curve integrated encryption scheme (ECIES) as per 5G standard. The session key binding with SP shall be achieved by including '*SPID*' and '*EID*' into the chain of key derivations parameters, it makes sure that the session key is specific for particular authentication process between a SP and a UE.

Figure 4 SAP-AKA key derivation (see online version for colours)



4.1 *Protocol message exchange and execution*

We now give an overview of the SAP-AKA protocol execution and message exchange, to illustrate the full execution of the protocol. It consists of service request and authentication phases. The protocol messages (msg) between the parties is illustrated in Figure 5, with reference to notations in Table 3.

4.1.1 *Phase 1: service request*

- Msg1.UE!SMF: ({ServSsReq}, {KAMF})

  After the primary authentication the UE sends a service session request message *ServSsReq* via AMF to SMF in 5G HN encrypted with key K_{AMF}, that includes the service name *Servname* and *SID* to request for a service and session establishment for a service request. The *Servname* is the identifier of the service, while *SID* is used for session management by SMF.

- Msg2.SMF!UE: ({ServSsResp}, {KAMF})

  The SMF checks user's subscription data, the primary authentication security status and context of the UE in HN database. It checks if the SP provisioning the service for the user resides inside or outside HN and security context available. If it is an external SP, then SMF retrieves the UE global generic identifier *GPSI* that corresponds to the UE's permanent identifier SUPI and sends it to the UE along with the *SPID* and SP's public key PK_{SP} in service session response message *ServSsResp* encrypted with K_{AMF}. SMF redirects the UE to SP for authentication and service authorisation.

- Msg3.UE!SP-AAA: ({ServReq}, {PKSP})

  Then UE sends a service request message *ServReq* to SP-AAA, it includes service name *Servname*, SID encrypted with SP public key PK_{SP}.

Table 3 SAP-AKA protocol notation and description

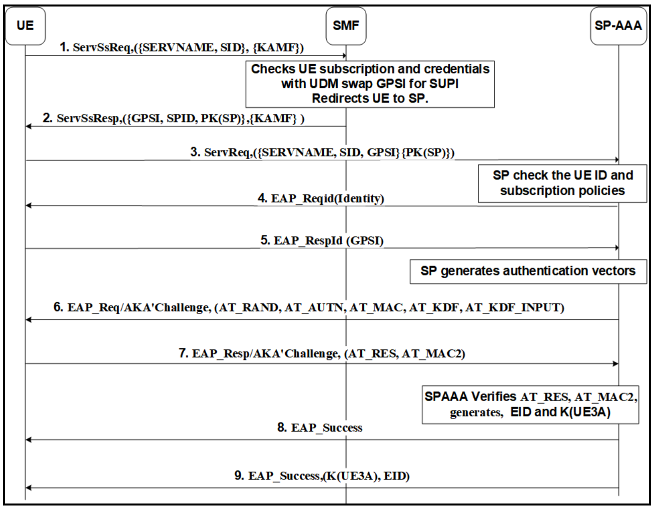
Notation	*Description*
SPID	SP identity
SID	Service/session identity
DNN	Service code: SPID (NAI)
NAI	(SID, SPID)
Aut_Key/K	Pre-shared key shared (UE, SP)
K_{AMF}	Session key for (UE,AMF)
RAND	Random nonce challenge
EID	UE Permanent Identity
GPSI	UE Generic Identity
AUTN	(SQNSP ▯ AK,MAC)
MAC, MAC2	f1(K, (SQNSP, Rand))
RES, XRES	f2(K, Rand)
CK	f3(K, Rand)
IK	f4(K, Rand)
AK	f5(K, Rand)
CK'	ik, ck, dnn, (sqn ⊕ ak))
IK'	ik, ck, dnn, (sqn ⊕ ak))
EMSK	KDF((CK', IK'), (SNN, SQN ⊕ AK))
SQN	Sequence number
PK_{SP}	SP public key
K_{UE3A}	KDF(EMSK, (EID, SPID))
h(x)	Hash value of message x
{x}{k}	Message encrypted with key K

4.1.2 *Phase 2: authentication*

- Msg4.SP-AAA!UE: (EAP_Reqid)

  The SP-AAA verifies the ServReq by checking the details of the *Servname* and *SID* in its database which include the services and agreement policies with HN, if they are valid then it sends an EAP request message requesting the UE to identify itself, starts an EAP AKA message exchange.

Figure 5 SAP-AKA protocol message exchange flow


UE
SMF
SP-AAA
1. ServSsReq,({SERVNAME, SID}, {KAMF})
Checks UE subscription and credentials with UDM swap GPSI for SUPI Redirects UE to SP.
2. ServSsResp,({GPSI, SPID, PK(SP)},{KAMF})
3. ServReq,({SERVNAME, SID, GPSI}{PK(SP)})
SP check the UE ID and subscription policies
4. EAP_Reqid(Identity)
5. EAP_RespId (GPSI)
SP generates authentication vectors
6. EAP_Req/AKA'Challenge, (AT_RAND, AT_AUTN, AT_MAC, AT_KDF, AT_KDF_INPUT)
7. EAP_Resp/AKA'Challenge, (AT_RES, AT_MAC2)
SPAAA Verifies AT_RES, AT_MAC2, generates, EID and K(UE3A)
8. EAP_Success
9. EAP_Success,(K(UE3A), EID)

- `Msg5.UE!SP-AAA: (EAP_RespId)`

  When the UE receives the messages, it sends an EAP response identity message to the SP-AAA and including its generic identity `GPSI`.

- `Msg6.SP-AAA!UE: (EAP_Req/AKA'Challenge)`

  After receiving msg 5, the SP-AAA checks UE's ID *GPSI* and its service policies to ensure that UE is authorised by the HN. The SP-AAA then generates the AVs which includes random number `RAND` authentication challenge token `AUTN` then computes MAC which are sent as `AT_RAND`, `AT_AUTN`, `AT_MAC` attributes together with `AT_KDF` key derivation functions for generating keys with `AT_KDF_INPUT` with the key input and `AUT_Key` authentication key for deriving CK'/IK'. The key derivations relies on AUT_Key which is normally a pre-shared secret key between the UE and AAA server, due to 5G's no security context sharing policy and secondary authentication being independent of HN, the `AUT_Key` has to be sent to the UE during the AKA challenge. So, the SP-AAA responds with EAP request sending AV to start EAP-AKA' challenge with the UE.

- `Msg7.UE!SP-AAA: (EAP_Resp/AKA'Challenge)`

  The UE verifies the AUTN the MAC, checks the token challenge and MAC values, for freshness and message integrity. The UE also retrieves the details in `AT_KDF_INPUT`, `AT_KDF`, `AUT_Key`, then derives CK'/IK' using *AUT_Key* then other necessary keys such as *MK*, *K_enc*, *K_re*, *MSK*, and *EMSK*. Then UE responds with EAP response message that includes (`AT_RES`, `AT_MAC2`) a response to the challenge sent in msg 6 with a new *MAC*2.

- `Msg8.SP-AAA!UE: (EAP_Success_UE)`

  The SP checks *MAC*2 and *RES* values from UE received in message 7, if they correct it generates session key K_{UE3A} for UE and SP-AAA, using key derivation function *KDF*(*MSK*, (*eid*, *spid*)) and a permanent identifier *EID* for use during service authorisation procedure. The session key K_{UE3A} and *EID* encrypted with *MSK* key in EAP success message.

- `Msg9.SP-AAA!SMF: (EAP_Success_SMF)`

  The SP also sends an EAP success message to the SMF to inform the HN that authentication was successful. That SMF sends message to UDM to update the UE's profile.

5 Verification of SAP-AKA protocol

In this section, we formally model and verify the SAPAKA protocol using ProVerif and applied Pi calculus. We formalise the protocol security properties with ProVerif results.

5.1 Formal verification of the SAP-AKA protocol using ProVerif

The modelling of a protocol in ProVerif is composed of declaration, process macros and main processes. The queries are carried out to rectify the correctness and secrecy of a protocol. The ProVerif code is used to specify the protocol concisely using declaration of types, functions, queries, and events. Free names are free variables that are known to the public, globally known whereas bound names are locally known by the process like the public channel for communication [private] excludes names from the attacker (Blanchet et al., 2020).

Specification include the following:

- Functions: `fun PRF(key, key, bitstring, id):key.`
- Key: `type key.`
- Private and public names:

  `free secretUE:bitstring [private]`

  `free eid:id [private]`

  `free kue3a:key [private] free pubChannel:channel.`

- Queries: Queries on secrecy, reachability, and authentication. A secrecy property is specified as a query of the attacker's knowledge `attacker(M)`. When the fact `attacker(M)` is derivable from the horn clauses, the attacker may have the knowledge of `M`. When the fact `attacker(M)` is not derivable from the clauses, there is no way that the attacker can gain the knowledge of `M`. With reachability, the `query` `attacker(K)` is also used to debug the model of the protocol to check a particular branch is reachable or not. `query k: bitstring; event(endServer(k))`. The authentication properties are specified as correspondence assertions in the form of `event(e1(M)) event(e2(M))`. If all clauses that conclude `event e1` contain `event e2` in their hypotheses, then `event e1` is derivable only when `event e2` holds, so the correspondence assertion is proven. In case of the SAP protocol the following is queried: `query attacker(Secret) query attacker(eid).query attacker(kue3a)` are used to test the secrecy of message, *EID* and key K_{UE3A}, respectively. While query `u:host, a:host, r:nonce, kue3a:key, k:key; event(endAAA(u, a, r, k)) ==> event (beginUE(u, a, r, k))` is used to test events relationships (authentication).

- Events: Querying events using correspondence assertion to test the relationship between events.
  1. Event correspondence uses syntax to query a basic correspondence assertion, `query x1: t1, …, xn: tn; event(e(M1, …, Mj)) ==> event(e′(N1, …,Nk))`. Where `M1, …, Mj, N1, …, Nk` are terms built by the application of constructors to the variables `x1, …, xn` of types `t1, …, tn` and `e, e′` are declared as events.
  2. While the injective correspondence assertions capture the one-to-one relationship and are denoted, `query x1: t1, …, xn: tn; inj-event(e(M1, …, Mj)) ==> inj-event(e′(N1, …, Nk))`. The correspondence asserts that, for each occurrence of the event `e(M1, …, Mj)`, there is a distinct earlier occurrence of the event `e′(N1, …, Nk)`.
- Process: The protocol encoded using the main process and the process macros for the participating entities to allow sub-process being defined: `((!procUE(hostU))` from the UE `(!procAAA(hostA))` for the SP-AAA and `(!procSMF(hostS))` for SMF. The main process also starts of several copies of the system entities (UE, SMF, SP-AAA) with the relevant parameters representing several sessions of the roles as explained in the message exchange.

Table 4 Proverif query checks

Properties output	*Query*	*Expected output*	*ProVerif*
EID	Secrecy	True	True
K_{UE3A}	Secrecy	True	True
UE-SP	Non-injective	True	True
SP-UE	Injective agreement	True	True

5.2 *Formal analysis of SAP-AKA protocol*

We simulate the SAP-AKA protocol in ProVerif with the following processes:

- The three parties are:
  `((!procUE(hostU)) for UE`
  `(!procAAA(hostA))for SP-AAA`
  `(!procSMF (hostS)) for SMF.`

The following queries are used:

1. Secrecy:

   `free secretAAA, secretUE: bitstring [private].`

   `query attacker(secretAAA); attacker (secretUE).`

   `free eid:id [private]. query attacker (eid).`

   `free kue3a:key [private]. query attacker (kue3a).`

   `free k:key [private]. query attacker (k).`

2. Authentication:

   `query u:host, a:host, r:nonce, kue3a:key, k:key;`

   `event(endAAA(u, a, r, k))==> event(beginUE(u, a, r, k)).`

   `query u:host, a:host, r:nonce, kue3a:key, k:key;`

   `inj-event(endAAA(u, a, r, k))==> inj-event(beginUE(u, a, r, k)).`

When we modelled the protocol, we found that the authentication of UE and SP holds on both authorisation injective and injective agreements. The SP and HN implicit authentication is checked in the process. All the security properties we are interested are with respect to the K_{UE3A}, the mutual authentication for the UE and SP, the privacy of communication between entities. The results also indicate that the secrecy of *Secret*, *EID*, K_{UE3A} holds as shown in Table 4.

ProVerif results:

```
./proverif protocols/SAP-AKA.pv | grep RES
RESULT not attacker(secretAAA[]) is true.
RESULT not attacker(secretUE[]) is true.
RESULT not attacker(eid[]) is true.
RESULT not attacker(kue3a[]) is true.
RESULT not attacker(k[]) is true.
RESULT event(endAAA(u_98, a_99, r, k_101)) ==>
event(beginUE(u_98, a_99, r, k_101)) is true.
RESULT event(endUE(u_102, a_103, r_104, k_106))
==>
```

```
event(beginAAA(u_102, a_103, r_104, k_106)) is
true.
RESULT inj-event(endAAA(u_107, a_108, r_109,
k_111)) ==>
inj-event(beginUE(u_107, a_108, r_109, k_111))
is true.
RESULT inj-event(endUE(u_112, a_113, r_114,
k_116)) ==>
inj-event(beginAAA(u_112, a_113, r_114, k_116))
is true.
```

The event endAAA means that the SP-AAA has completed the protocol, that the UE received message 6 and sent message 7, that the SP-AAA sent message 6. These events take as arguments all parameters of the protocol: `at_rand:nonce`, `at_autn:bitstring`, `at_mac:bitstring`, `at_kdf:bitstring`, `at_kdf_input:bitstring`. The check the sent `at_mac`, `at_autn` and computes the `at_rand`. If the arguments are true then `at_res`, `at_mac2` are sent otherwise it sends authentication failure. We would like to prove the correspondence below.

```
query u:host, a:host, r:nonce, kue3a:key, k:key;
event(endAAA(u, a, r, k)) ==>
event (beginUE(u, a, r, k)).
query u:host, a:host, r:nonce, kue3a:key, k:key;
inj-event(endAAA(u, a, r, k)) ==>
inj-event(beginUE(u, a, r, k)).
```

In this case the direct proof of this correspondence in ProVerif holds because message 7 was sent and message 8 was received hence success of the authentication. We also try to prove the correspondence instead below and conclude the desired correspondence by noticing that msg 7 which has `at_res`, `at_mac2` as argument cannot be executed before `at_rand:nonce`, `at_autn:bitstring`, `at_mac:bitstring`, `at_kdf:bitstring`, `at_kdf_in put:bitstring` has been sent in msg 6, that is, msg 6 has been executed. Which holds in ProVerif with true.

6 Security analysis

This section presents the security analysis of the SAP-AKA security properties based on two taxonomies and discusses the security consideration of the protocol.

6.1 Protocol security analysis

Our threat model assumes a Dolev and Yao (1983) adversary model, it controls the network, can read, intercept, modify and send messages. It is also capable of initiating passive and active attacks such as eavesdropping, manipulation, interception, impersonation, and injection of messages. The adversary can also apply hashing, encryption and sign on values that are known to the attacker. The analysis is based on the symbolic protocol model, assuming that the cryptography is perfect, and the computational strengths of the primitives are not considered. However, the protocol should meet certain security properties and the analysis is based on the following properties in set 1 (Lowe, 1997) and set 2 (Menezes et al., 2001).

6.1.1 Analysis using security properties of set 1

- Secrecy: This is achieved since the K_{UE3A} of subscribers is never revealed to the attacker. By using XOR and anonymity keys protect the parameters used in derivations of keys in transit and in storage. The use of functions $f1, f2, f3, f4, f5$ to provide privacy protection of challenges/response of the data. By achieving this property also covers confidentiality and privacy of the protocol.
- Aliveness: The SP obtain the aliveness of the UE at that SMF, which is non-injective agreement on NAI from the SP's point of view with the subscribers. But also, the SP should have injective agreement on K_{UE3A} with the subscribers, which gives recent aliveness as a result.
- Weak agreement: This is achieved when HN achieve non-injective agreement on *EID* with UE as it is the ID. Also, the SP achieves weak agreement with HN after the key confirmation as the key includes *SPID* and *GPSI*. However, the weak agreement does hold as ProVerif result indicate.
- Non-injective agreement: The UE obtains non-injective agreement on NAI with its SP after key confirmation of K_{UE3A}. Moreover, since *GPSI* also is assigned by HN, an agreement on *EID*, is an agreement on *GPSI*. The HN obtain non-injective agreement on *EID* with the SP after *EID* assigned to UE by HN. The injective agreement on K_{UE3A} from the SP towards the UE, also guarantees that UE is attached to the authorised SP this IS achieved since K_{UE3A} is derivation include rand from SP and NAI. Which assures the UE that SP is trusted the authentication UE-SP holds.
- Injective agreement: The injective agreement on K_{UE3A} between the UE and the SP is central to the protocol's purpose. While the injective agreement on K_{UE3A} for different pairs of parties is achieved when the K_{UE3A} cannot be derived twice for the same session. The K_{UE3A} derivation also includes a `at_rand`, from which it obtains the desired assurance as an injective agreement on K_{UE3A} from the SP towards the subscribers. The injective agreement on K_{UE3A}, which is bound to *SPID* provided with the HNs assures that UE that SP is known and trusted. The UE obtain the injective agreement on K_{UE3A} with the SP to assure that the session was authorised by the HN. However, it achieves the same trust from UE as the event correspondence hold.

6.1.2 *Analysis using security properties of set 2*

- Mutual entity authentication: The UE is authenticated to SP if `at_res` and `at_mac2` are valid and to the HN to SP-AAA implicitly. Since the *SPID* and *GPSI* are included it enforces weak agreement and implicit authentication upon a successful authentication.
- Mutual key authentication: Since the SP-AAA `at_kdf:bitstring`, `at_kdf_input :bitstring` to UE for to key derivation parameters. It fulfils this requirement.
- Mutual key confirmation: After the successful AKA round-trip between the entities ending with SP-AAA sending SUCCESS message and K_{UE3A}, it enforces this requirement.
- Key freshness: ProVerif has no function to check key freshness however during the authentication process the UE checks the `at_autn` freshness and computes `at_rand`. K_{UE3A} is results of the input request that was sent by SP-AAA to UE in during the current protocol session, hence the input and the key are fresh.
- Unknown-key share: The reachability property in ProVerif is used to check aliveness. The entities ID and `at_kdf:bitstring`, `at_kdf_input: bitstring` prevent this attack. The inclusion *GPSI*, *SPID* in the authentication process and the *GPSI* in the derivation of `K_aut`, also proves this requirement. Also, the K_{UE3A} is only sent to UE after the RES and MAC2 verification by SP-AAA.
- Key compromise impersonation resilience: The K_{UE3A} is implicitly authenticated and its secrecy holds. It remains confidential new session even when the attacker learns the K_{UE3A} keys established in all other sessions and Since every key derivation input were sent by SP-AAA in a secure communication exchange as the defined by RFC 5448. However, forward secrecy and post-compromise secrecy might hold. EAP-AKA' does meet these requirements as knowledge of their no other key involved in the derivation of K_{UE3A}, therefore, derive all past and future keys are cannot be known the attacker based on `at_kdf:bitstring`, `at_kdf_input:bitstring`.

6.2 *Security consideration*

The service session establishment procedure is out of scope of this research for IP based procedure refer to 3GPP (2020a) and Ravindran (2017) for ICN-based procedure. Now the UE has been authenticated and authorised to service now we discuss different level of access as well as further authorisation to cache and share the data. When the UE registers with the network it shares some data with HN as per 3GPP standard and with SP as per the contract. Also, the HN gets in agreement with SP if their different entities. Also, the SP register its content with the SP 3A server. After the authentication, the SP-AAA will create a session key for UE and SP-AAA.

The proposed protocol should provide authentication and session for UE trying access services from the external network. Its main purpose is to allow the UE to communicate securely to non-3GPP networks without compromising the security context such as SUPI to an external network. The SMF may also check whether the UE has been authenticated or authorised by the same SP before. If so, they can use the previous keys and trust to generate the new session key. The *GPSI* used as UE's initial ID globally known but it is not used in the AVs as it is swapped with *EID*.

Moreover, there have been no published attacks that violate the AKA security properties defined under the originally assumed trust model and that of EAPAKA' (Arkko et al., 2019). Even though the diameter protocol is still vulnerable to attacks like man in the middle (MITM), Malware, and DDoS attacks that can be used for further attacks on the network (ENISA, 2018). Encryption is enabled in diameter, it is based on the peer-to-peer principle and not end-to-end, in most cases the security is built on trust between operators. Furthermore, interception and information gathering are possible due to diameter's use of same route for request/response message exchange.

7 Conclusions

With 5G full specification of primary authentication provides security for the network access between UE and HN while secondary authentication provide security between UE and SP as the UE tries access services provided by the SP. In this paper, we explored how secondary authentication based on EAP framework. We discussed the role of the SAP-AKA protocol as secondary authentication for service authorisation to the SP. We proposed a services authentication protocol SAPAKA that can be used to secure service access request process between UE and the SP network. The SAPAKA provides EID and session key for service authorisation. We modelled and formally analysed the proposed protocol using formal methods and automated proof verifier ProVerif based on applied Pi calculus. We conducted a security analysis on the protocol security properties based on two taxonomies. In future work, we would like to analyse the protocol using computational modelling and we should also build on this protocol by investigating service authorisation of UE to multiple SPs in 5G network using this proposed protocol in this paper as the cornerstone.

Acknowledgements

This research is in collaboration and partially funded by Naif Arab University for Security Science.

References

3GPP (2018a) *3G Security; Security Architecture*, Technical Report 3GPP TS 33.102 V15.1.0 (2018-12), Third Generation Partnership Project.

3GPP (2018b) *3GPP System Architecture Evolution (SAE) System Aspects, Security Aspects of Non-3GPP Accesses*, Technical Report 3GPP TS 33.402 V15.1.0 (2018-06), Third Generation Partnership Project.

3GPP (2020a) *Security Architecture; Procedures for 5G System*, Technical Report 3GPP TS 33.501 V16.2.0 (2020-03), Third Generation Partnership Project.

3GPP (2020b) *System Architecture for the 5G System*, Technical Report 3GPP TS 23.501 V16.4.0 (2020-03), Third Generation Partnership Project.

Aiash, M. (2013) 'A formally verified initial authentication and key agreement protocol in heterogeneous environments using Casper/FDR', in *International Conference on Network and System Security*, Springer, Berlin, Heidelberg, pp.742–748.

Arkko, J., Eronen, P., Lehtovirta, V. and Torvinen, V. (2018) 'Improved extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA)', *The Internet Engineering Task Force (IETF) Request for Comments* [online] https://tools.ietf.org/html/draft-ietf-emu-rfc5448bis-03 (accessed 1 September 2020).

Arkko, J., Eronen, P., Lehtovirta, V. and Torvinen, V. (2019) 'Improved extensible authentication protocol method for 3GPP mobile network authentication and key agreement (EAP-AKA)', *The Internet Engineering Task Force (IETF) Request for Comments* [online] https://tools.ietf.org/html/draft-ietf-emu-rfc5448bis-05 (accessed 7 October 2020).

Arkko, J., Lehtovirta, V. and Eronen, P. (2009) 'Improved extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA')', *The Internet Engineering Task Force (IETF) Request for Comments* [online] https://tools.ietf.org/html/rfc5448 (accessed 4 October 2020).

Armando, A., Basin, D.A., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J.R., Drielsma, P.H., Heam, P.C., Kouchnarenko, O., Mantovani, J., Modersheim, S.A., Oheimb, D.V., Rusinowitch, M., Santiago, J., Turuani, M., Vigano, L. and Vigneron, L. (2005) 'The AVISPA tool for the automated validation of internet security protocols and applications', *Proceedings Computer Aided Verification*, Vol. 3576, pp.281–285.

Basin, D., Dreier, J., Hirschi, L., Radomirović, S., Sasse, R. and Stettler, V. (2018) 'A formal analysis of 5G authentication', in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp.1383–1396.

Blanchet, B., Smyth, B., Cheval, V. and Sylvestre, M. (2020) *ProVerif 2.01: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial* [online] https://prosecco.gforge.inria.fr/personal/bblanche/-proverif/ (accessed 4 October 2020).

Dolev, D. and Yao, A.C-C. (1983) 'On the security of public key protocols', *IEEE Transactions on Information Theory*, Vol. 30, No. 2, pp.198–208.

Edris, E.K.K., Aiash, M. and Loo, J. (2020a) 'Formal verification and analysis of primary authentication based on 5G-AKA protocol', in *The Third International Symposium on 5G Emerging Technologies (5GET 2020)*, IEEE, Paris, France.

Edris, E.K.K., Aiash, M. and Loo, J. (2020b) 'Network service federated identity (NS-FID) protocol for service authorization in 5G network', in *5th IEEE International Conference on Fog and Mobile Edge Computing (FMEC 2020)*, IEEE, Paris, France.

ENISA (2018) *Signalling Security in Telecom SS7/Diameter/5G*, Technical Report, ENISA [online] https://www.enisa.europa.eu/publications/signallingsecurity-in-telecom-ss7-diameter-5g (accessed 5 October 2020).

Küsters, R. and Truderung, T. (2009) 'Using ProVerif to analyze protocols with Diffe-Hellman exponentiation', in *2009 22nd IEEE Computer Security Foundations Symposium*, IEEE, pp.157–171.

Lowe, G. (1997) 'A hierarchy of authentication specifications', in *Proceedings 10th Computer Security Foundations Workshop*, IEEE, pp.31–43.

Menezes, A.J., Oorschot, P.C.V. and Vanstone, S.A. (2001) *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Fla, USA.

Ravindran, R. (2017) 'Enabling ICN in 3GPP's 5G nextgen core architecture', *The Internet Engineering Task Force (IETF) Request for Comments* [online] https://tools.ietf.org/id/draft-ravi-icnrg-5gc-icn-00.html (accessed 5 October 2020).

Ryan, M.D. and Smyth, B. (2011) 'Applied Pi calculus', *Formal Models and Techniques for Analyzing Security Protocols*, Vol. 5, pp.112–142, DOI: 10.3233/978-1-60750-714-7-112.

Vollbrecht, J.R., Aboba, B., Blunk, L.J., Levkowetz, H. and Carlson, J. (2004) 'Extensible authentication protocol (EAP)', *The Internet Engineering Task Force (IETF) Request for Comments* [online] https://tools.ietf.org/html/rfc3748.

Zhang, J., Yang, L., Cao, W. and Wang, Q. (2020) 'Formal analysis of 5G EAP-TLS authentication protocol using ProVerif', *IEEE Access*, Vol. 8, pp.23674–23688, 10.1109/ACCESS.2020.2969474.