Formalization and evaluation of EAP-AKA' protocol for 5G network access security

Array 16 (2022) 100254

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Formalization and evaluation of EAP-AKA' protocol for 5G network access security

Ed Kamya Kiyemba Edris [a,*], Mahdi Aiash [a], Jonathan Loo [b]

[a] School of Science and Technology, Middlesex University, The Burroughs, Hendon, London, NW4 4BT, United Kingdom
[b] School of Computing and Engineering, University of West London, St Mary's Rd, Ealing, London, W5 5RF, United Kingdom

ARTICLE INFO

Keywords:
5G
EAP-AKA'
Security protocol
Formal methods
Verification
Authentication
ProVerif
Applied pi calculus
Performance evaluation

ABSTRACT

The end user's Quality of Experience (QoE) will be improved while accessing services in Fifth Generation Mobile Network (5G), supported by enhanced security and privacy. The security guarantees offered by the Authentication and Key Agreement (AKA) protocols will be depended upon by end users and network operators. The AKA protocols have been standardized for 5G networks, and the Extensible Authentication Protocol (EAP)-AKA' protocol is one of the main authentication mechanisms that has been specified for User Equipment (UE) and network mutual authentication. This article models the EAP-AKA' protocol and conducts an extensive formal verification of the EAP-AKA' protocol as defined in the 5G security standard to determine whether the protocol is verifiably secure for 5G. It provides a security evaluation of the EAP-AKA' protocol based on the current 5G specifications using ProVerif, a security protocol proof verifier. It also presents security properties that support the security verification, as well as quantitative properties that are used to assess the protocol's performance. Finally, it compares the EAP-AKA' and 5G-AKA protocols' security and performance results.

1. Introduction

The Fifth Generation Mobile Networks (5G) will support applications like the Internet of Things (IoT) and Vehicle to Everything (V2X), as well as user mobility, dense connectivity, and massive Machine Type Communication (mMTC). The immense growth of mobile device and multimedia applications usage has led to the need for seamless connectivity and rapid growth of mobile data traffic. 5G will be supported by technologies such as visualization, edge computing and Device to Device (D2D) communications. Mobile subscribers with their User Equipment (UE) will be able to access 5G network services via the new generation Radio Access Network (ngRAN), therefore secure access drive 5G security standard requirements as specified by Third Generation Partnership Project (3GPP) in [1]. Users and Mobile Network Operators (MNOs) should be able to rely on stated security properties such as authentication, secrecy, and integrity provided by 5G security. To access the network, the UE must be authenticated, and additional authentication and authorization are essential to access services provided by the MNO or other Data Networks (DN). The UE, Serving Network (SN), and Home Networks (HN) use the Authentication and Key Agreement (AKA) protocols for security assurance.

The 5G standard [1] covers the most significant security needs in 5G and specifies Extensible Authentication protocol (EAP)-AKA' as one of the methods used in the primary authentication. There has been significant research conducted on primary authentication but mostly covers 5G-AKA protocol, hence why this paper is focussing on verifying EAP-AKA' protocol with ProVerif [2] proof verifier and evaluating its performance using analytical and simulation methods.

The main contribution of this paper is summarized as follows:

- It interprets security properties and models the EAP-AKA' protocol as described in the 3GPP standard.
- It conducts a formal analysis and verification of the protocol to automatically identify the security properties.
- It presents our security consideration on EAP-AKA', to provide the basis for future formal analysis and verification of next-generation AKA protocols.
- It evaluates EAP-AKA' protocol's performance using two models and compares with 5G-AKA.

The rest of the paper is structured as follows. In Section 2, related work on the EAP framework, 5G security, and formal methods are presented. In Section 3, EAP–AKA' protocol based on 3GPP standard is presented. Section 4 presents the protocol modelling and discusses the security requirements. The formalization of the protocol is discussed in 5. In Section 6 a formal security analysis of EAP-AKA' and security

* Corresponding author.
E-mail addresses: ee351@live.mdx.ac.uk (E.K.K. Edris), m.aiash@live.mdx.ac.uk (M. Aiash), jonathan.loo@uwl.ac.uk (J. Loo).

https://doi.org/10.1016/j.array.2022.100254
Received 1 September 2022; Received in revised form 10 October 2022; Accepted 25 October 2022
Available online 28 October 2022

considerations are discussed. Section 7 presents the performance evaluation of the protocol. The concluding remarks and future work are summarized in Section 8.

2. Related work

The authors in [3–5] reviewed the EAP concept that was specified under Request for Comment (RFC) 3748 as an authentication framework. [6]. It can be used on dedicated links, wired and wireless networks, and runs directly on data link layers without requiring an IP address. 3GPP designed the EAP-AKA protocol, which was then confirmed by the EAP WG in RFC 4187 [7]. It was later specified as an EAP method for authentication and session key distribution for Universal Mobile Telecommunications System (UMTS). Identity privacy support, result indications, and a fast re-authentication procedure were all added to the EAP-AKA. This made the use of AKA method for primary authentication possible within the EAP framework, later improved in RFC 9048 [8] with a new EAP method, EAP-AKA'. A new key derivation function was added, which binds the derived keys with the name of the access network, preventing binding attacks [4].

In addition, the EAP-AKA' can be used to authenticate the UE to network access in a 5G, non-3GPP networks and integrated in security frameworks [9]. The protocol was specified in TS 33.501 [1] as a 5G primary authentication method. The EAP-AKA' uses cipher key (CK') and integrity key (IK') as specified in TS33.402 [10] with an updated hash function Secure Hashing Algorithm (SHA)-256 and Hash based Message Authentication Code (HMAC)-SHA-256 [4].

2.1. Authentication procedure

The AKA is an authentication process where parties involved exchange messages to verify one another through mutual authentication and session key agreement. The 5G system supports AKA between the UE and SN, authorized by the HN. It provides ciphering, integrity and replay protection and privacy within the 5G, enabling the UE to access the HN via SN securely. 5G support primary authentication and secondary authentication methods for accessing the network in 5G and services from external DN, respectively. The EAP framework can be used for both methods, as specified in 5G standard [1].

2.2. Protocol verification using formal methods and automated proof verifier

Due to the use of strong abstractions, simplifications, and constraints in the properties analysis, formal methods and automated verification have been applied to authentication protocols like AKA in the past with weak guarantees. To give solid guarantees, formal approaches have already been used to examine security protocols in [3,4,11]. Most verification approaches and tools struggle with AKA protocol properties like in EAP-AKA', because of cryptographic primitives applications such as Sequence Number (SQN) and Exclusive-OR (XOR), whose algebraic features make symbolic reasoning difficult [12]. As a result, manual proof evaluations are not suitable for certain tools.

Many automated verification tools, such as Tamarin [13] Automated Validation of Internet Security Protocols and Applications (AVISPA) [14] and ProVerif [2], can be used for this analysis. ProVerif assesses the security of cryptographic protocols, and it uses Dolev–Yao models to facilitate user-defined equational theories and the verification of a wide range of security features. ProVerif also recognizes cryptographic primitives specified by rewrite rules and equations that satisfy the finite variant property. To support protocol reasoning, the syntax is combined with a formal semantics. As a result, we consider ProVerif to be an appropriate tool for our analysis. In [3,5], it was applied to formally check security characteristics assurances of AKA protocols.

The authors in [5] modelled the entire architecture of the 5G EAP-AK'A protocol through symbolic model verification with analysis. They also formally verified the security of the protocol. With a thorough formal investigation of the security-related characteristics of the 5G EAP-TLS authentication protocol, the authors in [3] reviewed EAP using ProVerif. The authors discussed the EAP framework and analysed the security features based on 5G-AKA in [12,15]. With security analysis based on Lowe's taxonomy, the authors in [16] formally modelled and examined the EAP-AKA' using Tamarin. The related work in officially analysed the EAP-AKA' protocol using formal methods with various automated tools, but they did not evaluate the protocol's effectiveness. In contrast, this research formally analysed and evaluated the protocol's performance using the two techniques that are covered in the next sections.

3. EAP-AKA' protocol

3GPP defined the EAP-AKA' protocol as one of the main methods of AKA between a mobile device and its HN in the 5G security standard [1]. The authentication protocol includes a home control feature that enables the HN operator to determine whether the device is authenticated in each network and make the final authentication decision. After a successful run of the protocol, all parties should be able to derive and agree on an anchor key, which is used to generate session keys for communication between UEs and Next Generation Node B (gNodeB) in the local network. Additionally, for secondary authentication between the UE and external DN, EAP method is preferred.

3.1. EAP architecture overview

The 5G system consists of the following:

- UE: A Mobile Equipment (ME) which stores the Universal Subscriber's Identity Module (USIM) with cryptographic capabilities like symmetric encryption algorithms, HMAC, and session counter, Subscriber's Permanent Identifier (SUPI), public key of its HN, key K.
- HN: Comprises security functions and the database that support authentication, generates vectors and stores user subscription data.
- SN: The radio access network to which the UE connects to access the HN.

As illustrated in Fig. 1, the security architecture of 5G comprises the following entities as described in [1,11]. The UE, Security Anchor Function (SEAF), Authentication Server Function (AUSF), Authentication Credentials Repository and Processing Function (ARPF) and Unified Data Management (UDM). The SEAF is deployed in the SN, while the AUSF and ARPF are located in the HN. Additionally, the UE shares the secret key and other information with ARPF which are used during the AKA procedure. The subscriber unique identity SUPI is encrypted as Subscription Concealed Identifier (SUCI) when UE sends it to HN and only decrypted by the HN. The UE and SEAF must achieve mutual authentication and must be in the possession of session key before communicating as this occurs on an unsecure wireless channel while the communication between SEAF, AUSF and ARPF occurs on a presumed secure wired channel.

The EAP framework defines the roles of peer, pass-through authenticator, and back-end authentication server under RFC 3748. The EAP server, which is the back-end authentication server, terminates the EAP authentication mechanism with the peer, the entities are shown in Fig. 2. When EAP-AKA' is utilized in a 5G system, the EAP framework is supported in the following way [1]:

- The peer is represented by the UE.
- The pass-through authenticator is represented by the SEAF.

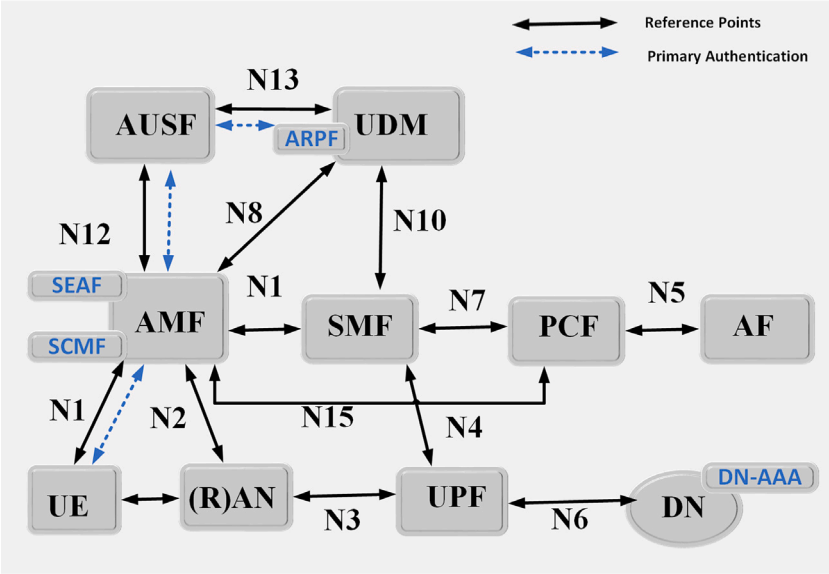


Fig. 1. 5G system architecture.

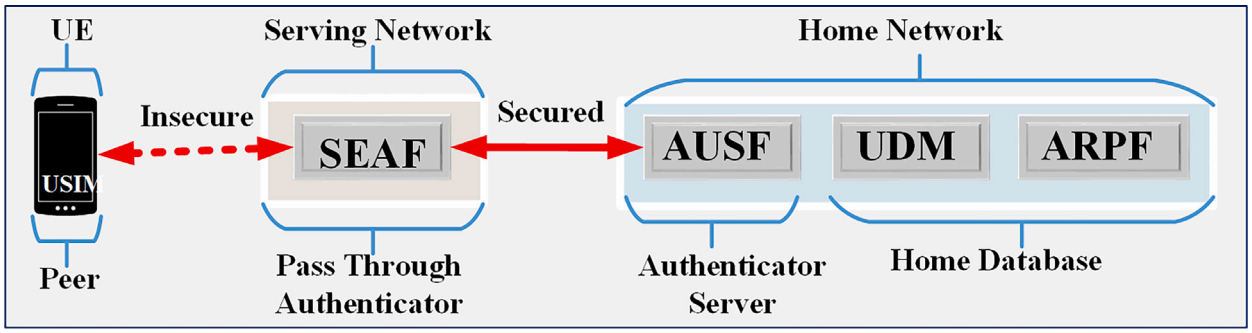


Fig. 2. 5G EAP entities.

- The backend authentication server is represented by the AUSF with the support of ARPF and UDM.

The main EAP-AKA' Attributes in the EAP-request/response are AT_RAND, AT_AUTN, AT_RES, AT_MAC, AT_KDF, AT_KDF_INPUT, AT_MAC, AT_AUTS. There are some significant changes that have been made in EAP-AKA', as per 5G security specifications [8] as follows:

- Network name field: Both the UE and the SEAF must generate the SNN, setting the service code to "5G" and the network identifier to "SNID" for the network to which the UE is authenticating and for the SEAF to SN to which the AUSF is delivering authentication data.
- The identifiers: SUPI, SUCI, SNID.
- Inputs in key derivation: It is critical for identity privacy, privacy-friendly and non trackable identifiers in 5G.
- Session identifiers: = EAP Type code || RAND || AUTN. It carries the AT_KDF_INPUT attribute, which ensures that the UE and AUSF are both aware of the access network's name. To enable future extensions, it offers key derivation function negotiation via the AT_KDF parameter.

3.2. Keys derivation

During the derivation of K_{SEAF} from the K_{AUSF} following a successful authentication procedure between the UE and the HN, the UE's identity and the access network ID are used as inputs in the key derivation using the at_kdf_input parameters. The keys related to authentication include keys: K, CK, IK, CK', and IK' which are used to generate the EMSK (K_{AUSF}) then K_{SEAF} and later used to derive other keys to secure communication between the UE and other network entities. In addition, the KDF input parameters for CK' and IK' are the same only separated by the return of a 256-bit output, with CK's 128 most significant bits and IK's 128 least significant bits [10].

4. Modelling of EAP-AKA' protocol

In this modelling, the values used for Authentication Vectors (AV) are defined in Table 1: They include a random nonce RAND as a challenge, AUTN as an authentication token to validate the challenge's freshness and authenticity, and an expected reply as XRES to the challenge. The EAP-AKA' protocol is modelled with four entities (UE, SEAF, AUSF, and ARPF/UDM). The focus is on authentication and authentication failure, not re-authentication. XOR and HN public keys with Elliptic Curve Integrated Encryption Scheme (ECIES) profiles are used to conceal the SQN and SUPI, respectively. By including "SNN" in the chain of key derivations parameters, the anchor key K_{SEAF} binding with SN is enforced, guaranteeing that the anchor key is specific for an authentication process between a network and a UE. 5G authentication that utilizes the ARPF and USIM directly provides a stronger guarantee, similar to fast re-authentication in EAP-AKA'.

Table 1
5G EAP-AKA' notation and description.

Notation	Description
SNname/SNN	Service code:SNID
K	Symmetric key (UE, HN)
PK_{HN}	Public key (HN)
SK_{HN}	Private key (HN)
SIDF	Decrypt SUCI function
RAND	Random nonce challenge
SUCI/SUPI	User's Network Access Identifier (NAI)
AUTN	(SQNHN ⊕ AK,MAC,AMF)
MAC,MAC2	f1(K, (SQNHN, Rand,AMF)
RES, XRES	f2(K, Rand)
CK	f3(K, Rand)
IK	f4(K, Rand)
AK	f5(K, Rand)
CK'	ik, ck, snn, (sqn⊕ak)
IK'	ik, ck, snn, (sqn ⊕ak)
K_{AUSF} /EMSK	KDF((CK', IK'), (SNN, SQN ⊕ AK))
K_{SEAF}	KDF(K_{AUSF}, SNN)
SQN	Sequence Number
MACS	f1* (AMF, RAND, K, SQNUE)
AK*	f5* (K, Rand)
AUTS	(SQNUE ⊕ AK) ‖ MACS
h(x)	Hash value of message x
{x}{k}	Encrypted message x

4.1. Security assumptions and requirements

If the channel between the SN and HN is presumed to be secure, it should provide confidentiality, integrity, authenticity, and replay protection using cryptographic primitives, keys and HMAC, according to the standards in TS 33.501 [1]. In case where the SN-HN channel is not secure, it is exposed to the same attacks as the UE-SN channel [11].

The entities which execute the diameter protocol, as well as the AKA itself, could be compromised if attacker capabilities grow. This assumption is supported by 5G characteristics that increase the attacker's vector [11]. It is also possible that the attacker might have the actual USIM under his or her control, in which case the attacker might have access to all secret values contained in the USIM, including SUPI, K, and SQN. Although EAP-AKA' lacks cipher suite negotiation capabilities, it does have a mechanism for determining key derivation functions. Mutual authentication, secrecy, cryptographic binding, and session independence are all security qualities provided by SHA-256, which are comparable to those supplied by EAP-AKA [8]. It is also assumed that because SHA-256 is a pseudo-random function, an attacker will not be able to deduce the pre-shared secret from any keys in any practically feasible way. Different identifiers are used by EAP-AKA' to identify the authenticating UE. Even though the protocol key strength precludes brute force assaults, it does not provide channel binding.

Secrecy, confidentiality, integrity, authenticity, and privacy are the desired security features for the EAP-AKA' protocol [1]. The UE must safely assume only SN authorized by their HN can be used for authentication. The UE must use implicit key authentication and confirmation to authenticate SN with the SNN. After key confirmation, a UE must get weak agreement on SNN with its HN.

In the process of AKA between UE and HN, the SN must be able to verify the UE by authenticating SUPI. EAP-AKA' is responsible for maintaining the secrecy of K_{SEAF} and since it is the anchor key. It emphasizes that the same K_{SEAF} should never be generated twice. Privacy in 5G has been strongly defined, subscription privacy is concerned with subscribers' personal data to ensure confidentiality, anonymity, and untraceability and avoid attacks [17]. 5G has precise requirements on privacy, the SUPI and SQN must remain secret in presence of passive attacks to avoid attack such as data leakage. Over communication channels, these security features are utilized to construct and maintain long-term IPSec, (D)TLS, or DIAMETER sessions. SEAF and AUSF are Diameter-based systems that run over IPSec or TLS.

3GPP tries to address security issues of fake base station and non-repudiation by increasing home control. This is achieved by AUSF sending authentication confirmation to ARPF, HN confirming the UE's identity and authorizing the SN sending the SNN to UE [1,11]. The assumption is that EAP-AKA' provides the same security as EAP-AKA or better. However, the EAP-AKA' can be affected by same attacks as 5G-AKA protocol like monitoring, location, desynchronization and linkability attacks, which affects the privacy properties [18].

There have been no revealed attacks that compromise the AKA security properties defined under the originally assumed trust model and that of EAP-AKA' [8]. Despite this, the diameter protocol is still vulnerable to man-in-the-middle, malware, and DDoS attacks [11,19]. Diameter dependent on the peer-to-peer principle rather than end-to-end encryption. Furthermore, interception and information gathering is possible due to diameter's use of same route for request/response message exchange.

4.2. Protocol message exchange

The EAP-AKA' protocol process is divided into the following three stages using authentication vectors (AV) and exchange messages (msg), as shown in Table 1.

Stage 1: Initiation and Method Selection

It involves the initialization and authentication method selection. As shown in Fig. 3, the SEAF initiates authentication process with the UE.

`Msg1. SEAF→UE: (EAP-Request/Identity)`

1. The SEAF send Identity request to UE.

`Msg2. UE→ SEAF: (SUCI)`

2. The UE sends authentication request in msg2 which includes SUCI and HNID.

`Msg3. SEAF → AUSF: (SUCI, SNN)`

3. The SEAF receives msg2 sends SUCI and SN name to AUSF in msg3.

`Msg4. AUSF → ARPF: (SUCI, SNN)`

The AUSF sends msg4 to UDM/ARPF in HN. Before using the SNN, AUSF checks and verifies that the SEAF is authorized. When the ARPF receives msg4, SUCI is de-concealed into SUPI using SIDF and chooses an authentication method.

Phase 2: The Protocol

The EAP-AKA' protocol flows in a form of EAP Challenge-Response between entities as illustrated in Fig. 4.

`Msg5.ARPF → AUSF:EAP-AKA' AV (RAND, AUTN, XRES, SNN, CK' ‖ IK', SUPI)`

UDM/ARPF produces authentication vectors AV with AMF* after receiving SUPI. RAND and SQN are generated first, followed by XRES and AUTN. Calculates CK and IK, as well as CK' and IK'. In msg4, the ARPF transmits EAP-Response/AKA'AV to the AUSF, signalling that EAP-AKA' would be utilized.

`Msg6. AUSF → SEAF: (RAND, AUTN, SNN)`

When the AUSF receives msg5, it stores XRES and SUPI before sending EAP-Request/AKA'-Challenge to the SEAF in msg6.

`Msg7. SEAF → UE: (RAND, AUTN, ngKSI, ABBA)`

In Auth-Request, msg7, the SEAF sends RAND and AUTN to the UE. The ABBA parameter must be included in this message to enable binding down protection.

`Msg8. UE → SEAF: (RES, MAC)`

When the UE receives msg7, it forwards RAND and AUTN to the USIM, which checks if the AUTN may be accepted to verify the AV's freshness. It generates AK and obtains SQN. Then it generates MAC2, checks to see (i) if MAC2 = MAC and (ii) if SQN is in the range, if SQNUE < SQNHN. USIM calculates RES if (i) and (ii) are the predicted responses. Finally, it calculates CK and IK. The UE receives RES, CK, and IK from the USIM, then compute CK', IK' and returns RES with MAC2.

`Msg9. SEAF → AUSF: (RES, MAC)`

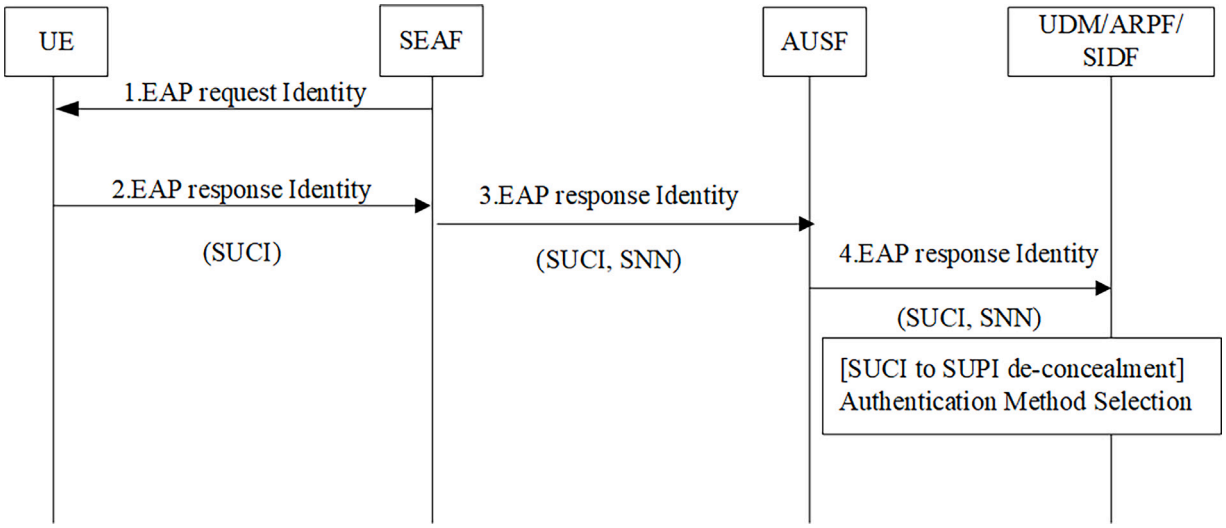


Fig. 3. EAP- AKA' phase 1.

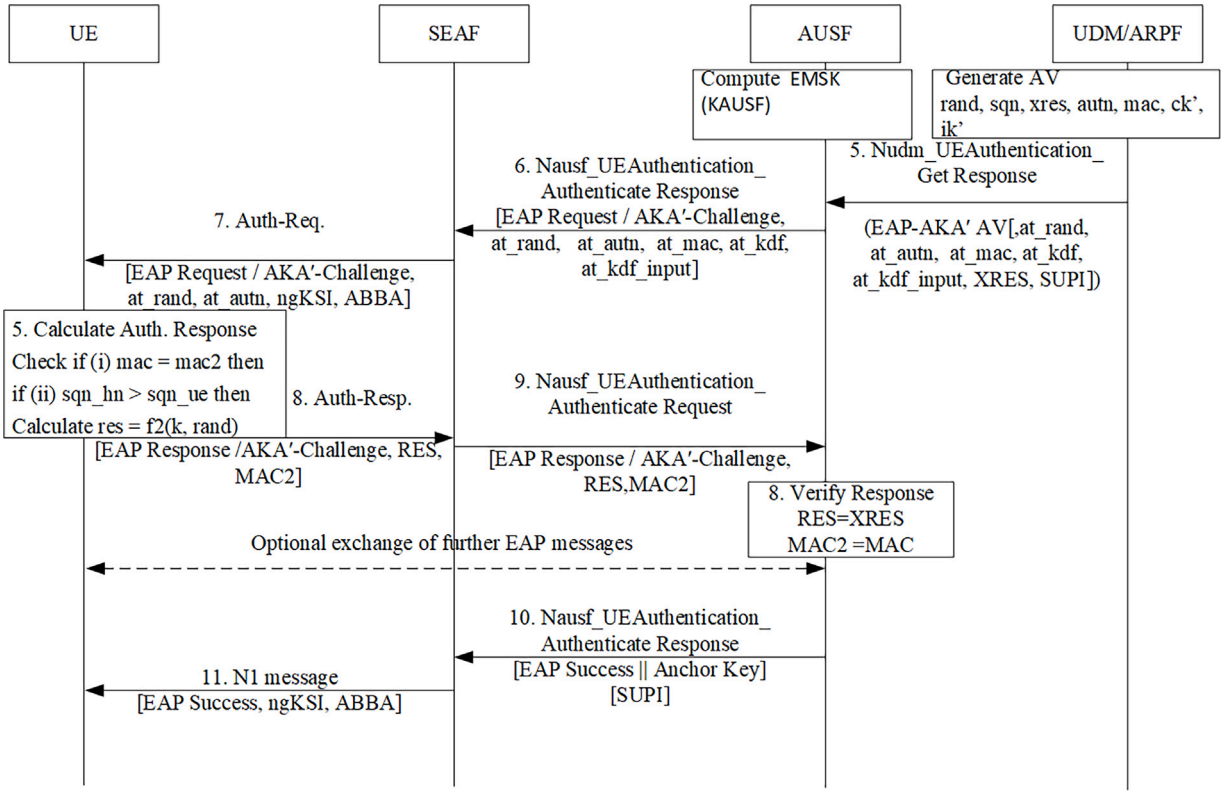


Fig. 4. EAP-AKA' phase 2.

The SEAF transparently forwards msg8 as msg9 to AUSF. There is an optional exchange of further EAP messages after msg9.

`Msg10. AUSF → SEAF:EAP-Success (KSEAF, SUPI)`

When AUSF receives RES and MAC2, it verifies them by comparing RES with XRES; if the two are equal, AUSF consider the authentication successful and informs the UDM. If not, it sends error message to SEAF. Otherwise it derives EMSK (K_{AUSF}) from CK' and IK', the calculates K_{SEAF} from K_{AUSF}. Then sends K_{SEAF} to SEAF in msg10 as the anchor key.

`Msg11. SEAF → UE: EAP-Success (ngKSI,ABBA)`

SEAF sends Success message in msg11 with ngKSI and the ABBA parameter. The UE drives EMSK (K_{AUSF}) from CK'and IK' and generates K_{SEAF} using the same methods as the AUSF.

Phase 3: Re-synchronization

If the SQN is out of sync, the re-synchronization technique is used to update it on the HN side. This is initiated by an AUTN verification fail, whereby the USIM lets the UE know whether it is a MAC or synchronization failure, and sends the AUTS parameter to the UE, as shown in Fig. 5.

`Msg12. UE → SEAF: (Mac_failure, Synch_failure, AUTS)`

In msg 12 with AUTS, the UE sends mac_failure and synch_failure to SEAF.

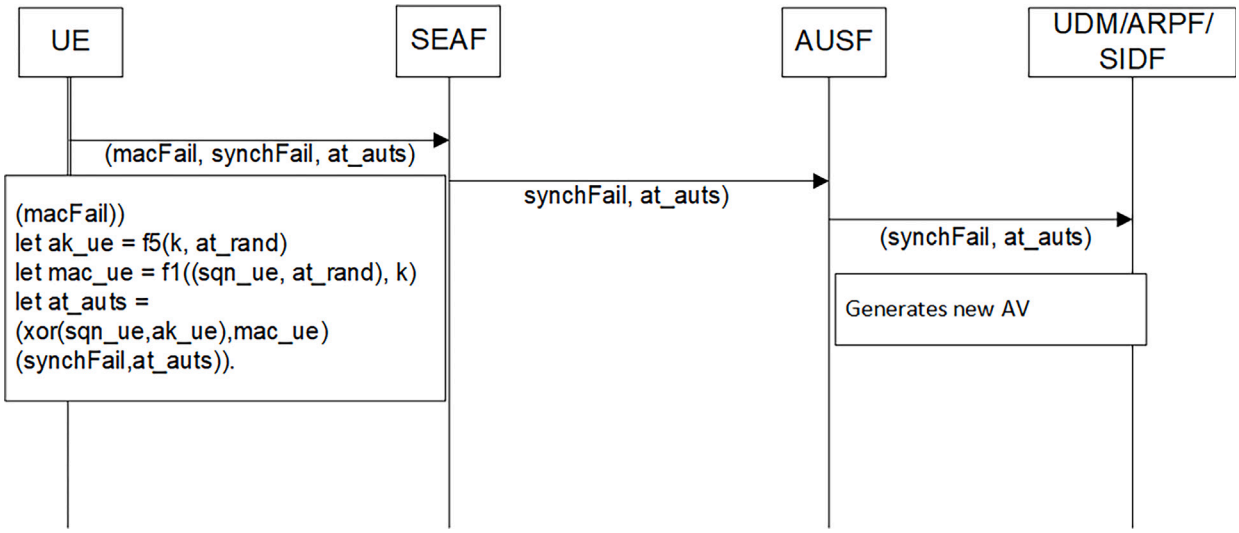


Fig. 5. EAP-AKA' phase 3.

`Msg 13. SEAF → AUSF: (Synch_failure,AUTS)`
SEAF may request the UE to re-identify its self in case of a mac_failure or launch a new authentication session if it is a sync_failure and sends msg13 to AUSF. `Msg 14. AUSF → ARPF: (Synch_failure,AUTS, Rand)`
With RAND transmitted to the UE in msg6, and AUTS received in msg 13, the AUSF sends msg14 to the UDM/ARPF. The ARPF obtains SQNUE from AUTS, verifies that SQNHN's range, and determines whether the SQN created with SQNHN will be accepted by the USIM. As a result, the UDM/ARPF computes fresh AV, checks AUTS, and resets the value of the counter SQNHN to SQNUE after a successful verification. The UDM/ARPF transmits new AV to the AUSF for the UE to perform a new authentication protocol run [1,11].

5. Verification of EAP-AKA' protocol

This section formalizes the EAP-AKA' protocol using formal methods and ProVerif.

5.1. Formal verification using ProVerif

In ProVerif, the declaration formalizes the behaviour of cryptographic primitives, which include variables, constants, names, and channels, to show reachability, secrecy, correspondence assertions, and observational equivalence of the protocol. The main process establishes the blueprint of the assessing scheme. Constants and variables, as well as cryptographic functions specified as constructors and equations, can be defined as free (unsecure) or private (secure). The beginning and termination of participating entities are defined by processes, and the execution is kept parallel. Then queries are run to ensure that a protocol's correctness and secrecy are maintained. Cryptographic primitives are represented by terms constructed from an unlimited number of names like (a, b, c, ...) and an infinite number of variables like (x, y, z, ...) and a finite number of function symbols such as $f1, \ldots, fn$. In addition, the application of function symbols to terms is influenced by a set of reduction rules. The syntax and grammar of the ProVerif process language are displayed in Table 2 [2].

Table 2
ProVerif process language.

Term	Grammar
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

5.2. Protocol formal analysis

These are some of queries used in the simulation.

```
free supi:id [private].
query attacker (supi).
free kseaf:key [private].
query attacker (kseaf).

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

The protocol was modelled and run in ProVerif using the secure `pubsec channel`, there was no effect on the protocol, and no attack was found. However, when the protocol was run on the compromised channel, the authentication did not hold as assumed by the 5G standard. This is a similar attack against 5G-AKA protocol [11].

The secrecy of UE's identity, long-term key, anchor key and authentication of UE to SN hold, according to ProVerif results in Fig. 6. However, the authentication of SN to UE does not hold on both non-injective and inject agreements. The UE received msg4 and transmitted msg5, as indicated by the `e1` showing that the SEAF sent msg4. All protocol parameters are taken as arguments in these events.

```
ededris@ededris-VirtualBox:~/proverif2.00$ ./proverif protocols/5G-EAP-AKA.pv |grep RES
RESULT not attacker(secretAUSF[]) is true.
RESULT not attacker(secretUE[]) is true.
RESULT not attacker(supi[]) is true.
RESULT not attacker(kseaf[]) is true.
RESULT not attacker(k[]) is true.
RESULT event(endAUSF(u_110,a_111,r,k_113)) ==> event(beginUE(u_110,a_111,r,k_113)) is true.
RESULT event(endSEAF(supi_114,kseaf_115)) ==> event(beginSEAF(supi_114,kseaf_115)) is false.
RESULT inj-event(endAUSF(u_116,a_117,r_118,k_120)) ==> inj-event(beginUE(u_116,a_117,r_118,k_120))
is true.
RESULT inj-event(endSEAF(supi_121,kseaf_122)) ==> inj-event(beginSEAF(supi_121,kseaf_122)) is false
.
RESULT (even event(endSEAF(supi_22429,kseaf_22430)) ==> event(beginSEAF(supi_22429,kseaf_22430)) is
 false.)
```

Fig. 6. EAP-AKA' unsafe ProVerif results.

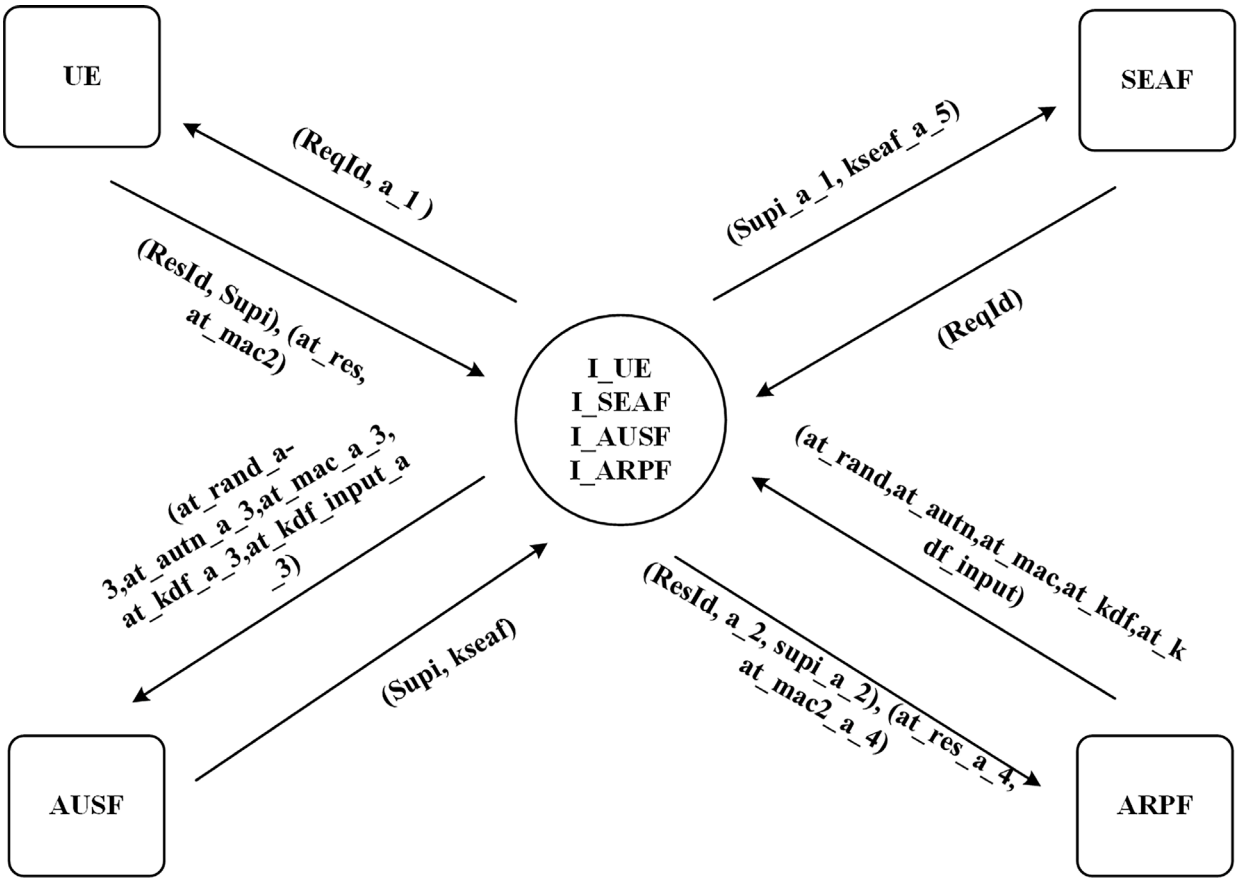


Fig. 7. Attack trace.

Except for `e2`, which checks if `xsqn = xor(xored sqn, ak)`, `xmac = f1((xsqn, xrand), ki)`, `xmac = mac` and if `xsqn = sqnue`. If the inputs are true, a *RES* message is conveyed; otherwise, a `MAC_failure` or `synch_failure` message is sent for authentication failure or re-authentication initiation.

However, because msg4 can be replayed, resulting in many `e2` for a single `e1`, this correspondence cannot be proved directly in ProVerif. The study also discovers that `event e2`, which has `res` as an argument that cannot be executed before `autn` and `rand` have been sent, i.e. before `e1`. Which fails as false in ProVerif.

5.3. The attack against EAP-AKA' protocol

As illustrated in Figs. 6 and 7, the ProVerif results indicate that there was an attack on the protocol. The attacker's behaviour is represented by the attack derivation, but the genuine attack is represented by the attack trace, which is an executable trace of the considered process. The derivation and trace are a set of steps, inputs, and outputs on the communication channel, as well as related events.

5.3.1. Attack derivation and trace

The attacker I intercepts communication between entities, impersonates the UE, and continues the protocol with SEAF, which completes the protocol with the attacker instead of the UE. The attacker's actions are depicted in Fig. 7 and concisely explained in steps below; some content has been excluded for simplicity:

- The `event(endSEAF(x1_80)) ==> event (beginSEAF(x1_80))` is queried, the attacker's goal is achieved, when he gets `suci_4228` using `attacker(suci_4228)`

```
ededris@ededris-VirtualBox:~/proverif2.00$ ./proverif protocols/5G-EAP-AKA.pv |grep RES
RESULT not attacker(secretAUSF[]) is true.
RESULT not attacker(secretUE[]) is true.
RESULT not attacker(supi[]) is true.
RESULT not attacker(kseaf[]) is true.
RESULT not attacker(k[]) is true.
RESULT event(endAUSF(u_110,a_111,r,k_113)) ==> event(beginUE(u_110,a_111,r,k_113)) is true.
RESULT event(endSEAF(supi_114,kseaf_115)) ==> event(beginSEAF(supi_114,kseaf_115)) is true.
RESULT inj-event(endAUSF(u_116,a_117,r_118,k_120)) ==> inj-event(beginUE(u_116,a_117,r_118,k_120))
is true.
RESULT inj-event(endSEAF(supi_121,kseaf_122)) ==> inj-event(beginSEAF(supi_121,kseaf_122)) is true.
```

Fig. 8. EAP-AKA' safe results.

and `attacker (rand_4227)` for `rand_4227` together with function `SHA256` to obtain `SHA256(rand_4227, x1_4230)`.
- The attacker's goal is also achieved when `event endSEAF(a)` is executed in session `copy a_4234`, SEAF completes a session with the attacker at event {57}. The injective agreement fails when `event endSEAF(a_5801)` is completed during session `a_5800`.

6. Protocol security analysis

This section analyses the EAP-AKA' protocol's security properties using taxonomies defined in [20,21] for security analysis 1 and 2, respectively, utilized to check formal methods outcome.

6.1. Security analysis 1

- Mutual Entity Authentication: If RES = XRES, the UE is implicitly authenticated to the HN and SN, as SNN is included in the successful authentication and K_{SEAF} confirmation. This is enforced when SUPI and SNN are transmitted to the HN and are proven to hold.
- Mutual Key Authentication: The UE and HN authentication is predicated on the secrecy of K_{SEAF}, it is implicitly authenticated by incorporating K_{AUSF} and SNN in its derivation parameters.
- Mutual Key Confirmation: This condition is enforced by a successful AKA roundtrip between the entities and with K_{SEAF} confirmation.
- Key Freshness: Although there is no function in ProVerif to check key freshness, the UE validates the AUTN freshness during the authentication process by checking if SQNUE > SQNHN. Additionally, every K_{SEAF} is linked to SN by SNN, which guarantees the key freshness, K_{SEAF} from prior sessions cannot be reused in new sessions.
- Unknown-Key Share: In ProVerif, the reachability property is utilized to check for aliveness. This attack is prevented by the entities' identity and key binding. SUPI, HNID and SNN in the derivation of K_{SEAF}, and the dependence on a preshared key K between UE and HN also prove this condition. Moreover, K_{SEAF} is only provided to SEAF after AUSF has verified the RES and MAC2.
- Key Compromise Impersonation Resilience: K_{SEAF} is implicitly authenticated and stays a secret. However, even if the attacker discovers K_{SEAF} keys used in all previous sessions and the key K material is compromised, the current session remains confidential. It does not, however, hold for forward secrecy or post-compromise secrecy. EAP-AKA' fails to meet these standards because knowing key K allows an attacker to deduce all previous and future keys.

6.2. Security analysis 2

- Secrecy: Since SUPI's secrecy is maintained, then this requirement is met. Key derivations parameters are protected in transit and storage by employing XOR and anonymity keys. F1 and F* give SQN privacy protection. The protocol's confidentiality maintained and privacy are preserved.
- Aliveness: HN obtains the aliveness of UE from the SN, with the UE agreeing to a non-injective agreement on SNN. Furthermore, the HN acquires fresh aliveness as a result of the injective agreement on K_{SEAF} with the UE.
- Weak Agreement: This is met when SN obtains non-injective agreement with UE on $SUPI$ and the key confirmation using SNN as parameter. However, as the ProVerif results show, the weak agreement does not hold.
- Non-injective Agreement: After K_{SEAF} confirmation and $HNID$ being part of SUPI, the UE gets non-injective agreement on SNN with its HN. The UE receives injective agreements on K_{SEAF} from both the SN and the UE. This due to K_{SEAF}'s derivation involving AT_RAND from HN and SNN from SN, any agreement on K_{SEAF} between the HN and the UE ensures that the UE is attached to an authorized SN. But the SN-UE authentication fails due to a change in the channel security assumption.
- Injective Agreement: The injective agreement on K_{SEAF} between the UE and the SN is critical to the protocol's goal, and establishing this for various pairs of parties means K_{SEAF} cannot be derived twice in the same session. As a result, using AT_RAND in the K_{SEAF} derivation ensures injective agreement on K_{SEAF} between the HN and the UE. It is worth noting that any agreement with HN on K_{SEAF} based on SNN informs the UE that SN is trustworthy. To ensure that the session with SN was authorized by the HN, the UE obtains an injective agreement on K_{SEAF} from the HN. The SN, on the other hand, is unable to gain the same level of trust from the UE since the SN-UE injective agreement does not occur.

It must be considered that key K could be leaked through eavesdropping on the communication channel, hacking the USIM card, an insider attack via the USIM vendor, mobile provider or side channels [11]. This would allow an attacker to impersonate a user to the SN, compromising the UE's privacy. Misuse of keys, on the other hand, could cause the SN to send traffic on behalf of the UE. However, knowing the K_{SEAF} established in one session is not enough to derive K_{SEAF} from a previous session or a future session in 5G. The network name binding in the EAP-AKA' protocol can also help to mitigate some of the attacks that affect the old EAP-AKA protocol, such as privacy attacks, but its configuration should not be based on the location of where a request originates unless the location information can be confirmed using cryptographic methods. In case, the SN requests a large number of authentication runs for a UE from a HN to induce a DoS, resynchronization and tracking/monitoring mechanisms should prevent this type of attack by limiting the number of authentication tries [8].

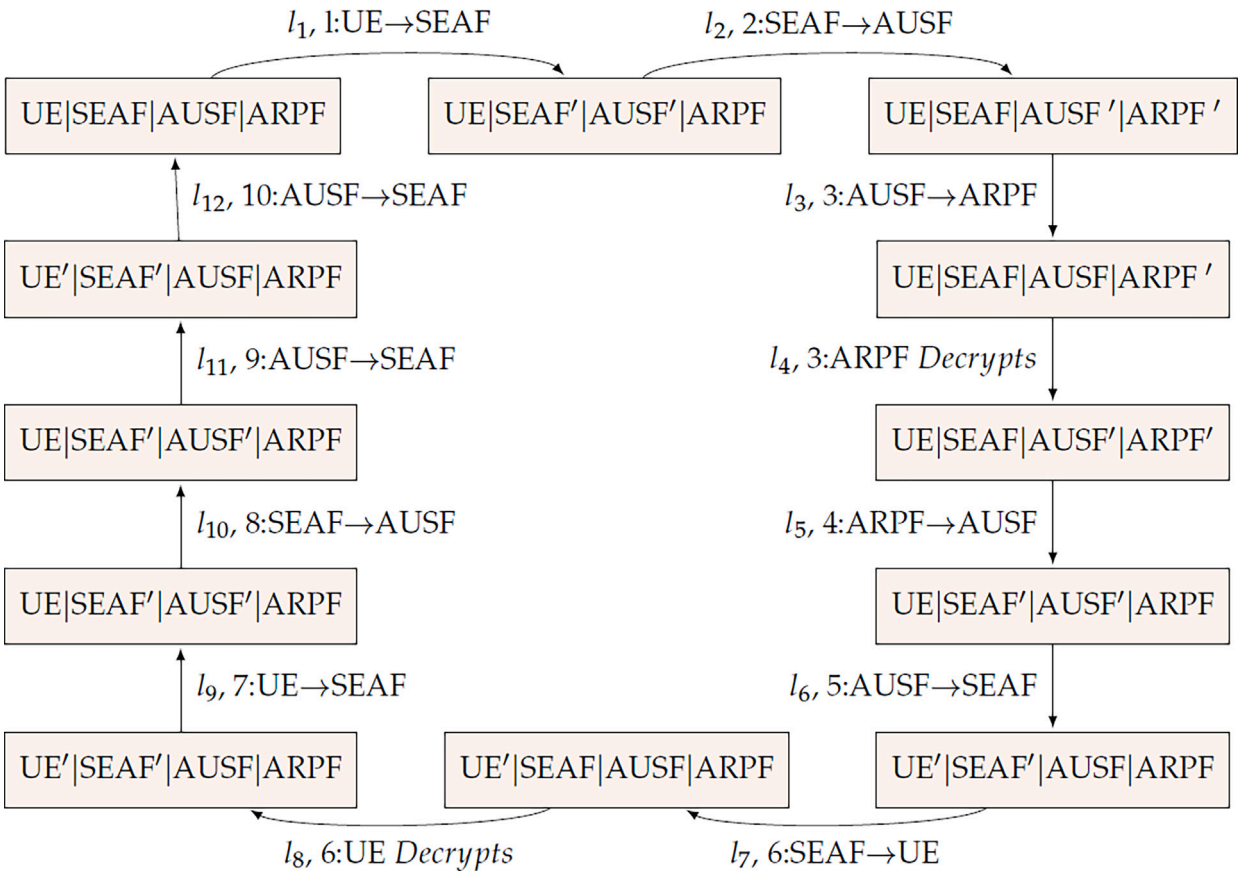


Fig. 9. EAP-AKA state transition system.

Table 3
Process extensions.

Term	Grammar
$A, B.C ::=$	extended processes
P	plain process
$A \mid B$	parallel composition
$vx.A$	variable restriction
$vn.A$	name restriction
$\{^M/_x\}$	active substitution

Furthermore, [12,18,22] ignored the sophistication of today's attackers. They assumed that security mechanisms in place would protect the diameter protocol, the channel, and HN entities, but they did not consider attacks in [19,23,24], that even a secure network can be compromised, resulting in a larger attack vector [11]. Because the non-injective and injective agreements between SN and UE fails, a replay attack is possible. However, after the replayed message has made a roundtrip to the HN, the SQN and unlink-ability problems [18] can be solved. The resynchronization is accomplished by either checking if the SQNUE sent in AUTS is greater than SQNHN, or setting SQNHN to SQNUE.

Therefore, if the standard is underspecified, the protocol vulnerability could allow the multiple attacks in 5G. Another additional measure is the use of Diffie–Hellman key exchange for perfect forward secrecy, but the computation cost is too much in terms of mobile device resources. Also, the authentication relying on the K_{AUSF} in AUSF is not as strong as direct authentication between the ARPF and the USIM [11].

When the protocol was simulated again with a secure communication channel in HN environment and using more robust mechanisms such as cryptographic techniques, and randomness, it was found that the authentication on UE and SN holds as assumed by the 5G standard on both non-injective and injective agreements a shown in Fig. 8. The security mechanism that protects the diameter sessions should also be enhanced.

7. Protocol performance evaluation

This section evaluates the protocol's performance using analytical and simulation approaches in [25]. EAP-AKA' protocol is assessed and compared with 5G-AKA protocol in [11].

7.1. Analytical performance evaluation

The analytical modelling in [25,26] associates an enhanced label to each communication and each decryption based on the ProVerif and Applied pi-calculus processes used in the protocol verification. It is supported with enhanced operational semantics based on labelled bisimilarity [27], processes that build finite state spaces, and maintains communication output and input components with their grammatical contexts. Each prefix of a particular process is given a context label ϑ and the processes' parallel composition (|) defines the entire system. As indicated in Table 3, the new names are formed using the restriction operator vnP, which operates as a static binder in the procedure P for the variable n. The transition of systems can be used to demonstrate communication, illustrated in Figs. 9 and 10 for EAP-AKA' and 5G-AKA protocols, respectively.

To evaluate quantitative aspects of transitions like cryptographic processes that conduct encryption and decryption, the cost is assigned to individual transitions derived from their labels [26]. The protocol's cost in terms of the time overhead of the primitives' actions, transition costs, which are determined by examining enhanced labels are given.

It is assumed that each entity has its processing unit, assigned a cost of 1 to each tag $\| i$, output and input are given the same cost. In a

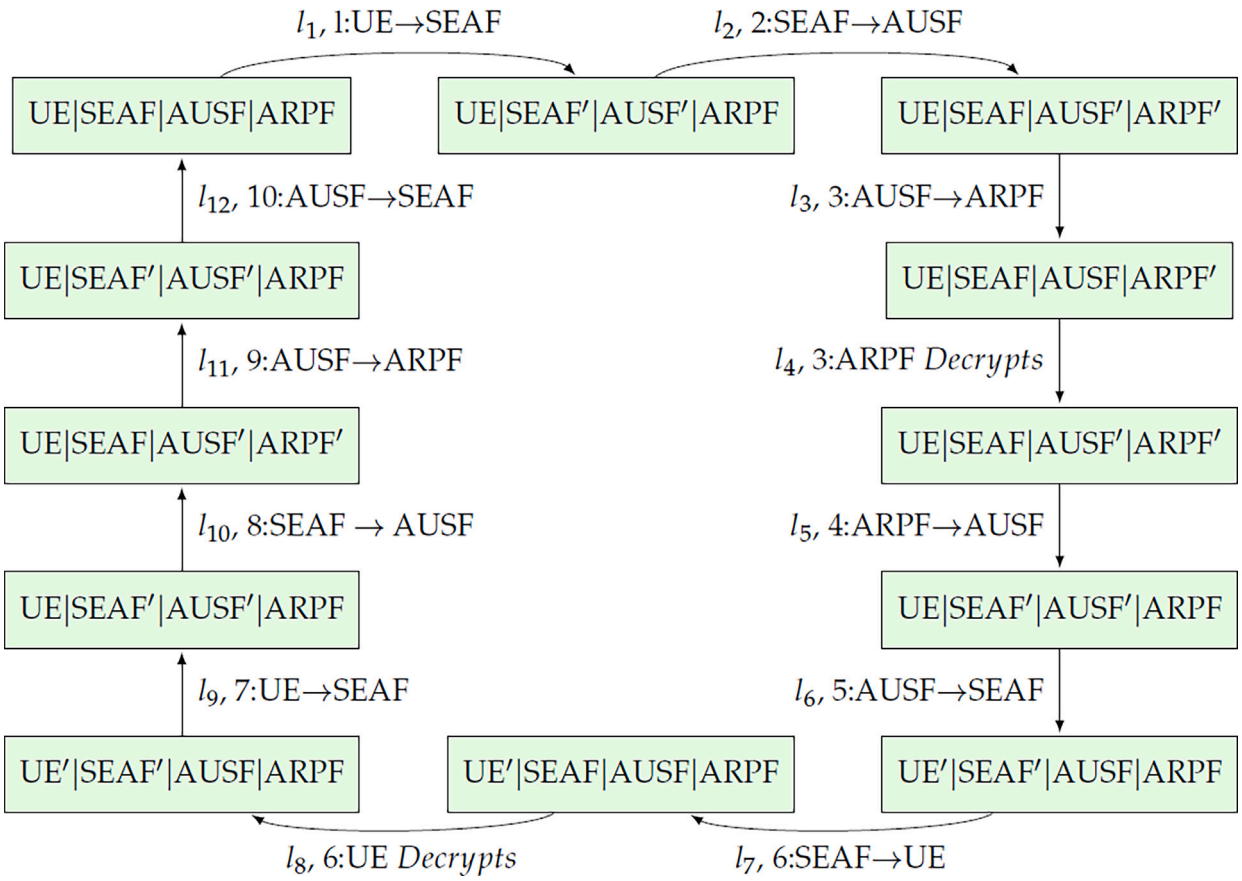


Fig. 10. 5G-AKA state transition system.

Table 4
Cost description.

Term	Description
n	message size
m_i	ith encryption size
e	unitary encryption cost
d	unitary decryption cost
s	unitary output cost
l_i	label of the state
c_i	cost of the label

Table 5
Cost labels for the protocols.

EAP-AKA'	5G-AKA
c1 = 2s + e	c1 = 2s+e
c2 = 3s	c2 = 3s
c3 = 3s	c3 = 3s
c4 = d	c4 = d
c5 = 5s + 8e	c5 = 5s+7e
c6 = 5s	c6 = 5s
c7 = 5s	c7 = 5s
c8 = 5d	c8 = 4d
c9 = 2s + e	c9 = s+e
c10 = 2s	c10 = s
c11 = d	c11 = 2s
C12 = 2s	c12 = 2s
C13 = s	c13 = d

transition, the cost of transmission is equal to $n * s + \sum_{i=1}^{l} m_i * e$ the cost of decryption is equal to $n * d$, and the terms are specified in Table 4. The cost c_i of the protocol in relation to its label l_i are presented in Section 7.1.2. Cryptography primitives, system architecture, protocols, and encryption algorithms such as ECC, SQN, AKA challenge, and XOR influence the cost.

7.1.1. Quantitative measurement

The enhanced operational semantics are used to obtain the quantifiable data required to generate the Continuous-time Markov chain (CTMC) process [28]. CTMC is made up of a number of states, labelled transitions in those states, and a sequence of random values, the probabilities of which depend on the values of previous states [29]. In [26], the costs are regarded as exponential distributions' parameters, when the transitions' exponential distributions are calculated, the arcs that share a source and a target are collapsed, resulting in the numerical process P.

Additionally, the parameter rate r is connected to a transition to estimate the rate, or transition probabilities at which a system switches from binding with process Pi to Pj. As a result, it is equivalent to the sum of all viable transition costs from Pi to Pj, and rates correlate with individual costs inside a transition system, as stated in [26]. A CTMC C is represented as a directed graph, where the nodes are the states of C and states reachable from one another are connected by arcs, as shown in Figs. 9 and 10. Because of this, the rates at which the process switches between states can be organized in a square/generator matrix represented as Q. The graph's adjacency matrix contains a representation of the CTMC for the process ($CTMC(P)$). Moreover, the elements of Q are used to illustrate the instantaneous transition rates [26]. The transition rate between two states Pi and Pj, denoted by the symbol $q(P_i, P_j)$, is the rate at which transitions between these states take place using Eq. (1).

$$\mathbf{Q1} = \begin{array}{c} \\ l1 \\ l2 \\ l3 \\ l4 \\ l5 \\ l6 \\ l7 \\ l8 \\ l9 \\ l10 \\ l11 \\ l12 \\ l13 \end{array} \begin{array}{c} \begin{array}{ccccccccccccc} l1 & l2 & l3 & l4 & l5 & l6 & l7 & l8 & l9 & l10 & l11 & l12 & l13 \end{array} \\ \begin{pmatrix} -x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3s & 3s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3s & 3s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -d & d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -z & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -5s & 5s & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -5s & 5s & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5d & 5d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2s & 2s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -d & d & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2s & 2s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -s \end{pmatrix} \end{array}$$

Box I.

$$\mathbf{Q2} = \begin{array}{c} \\ l1 \\ l2 \\ l3 \\ l4 \\ l5 \\ l6 \\ l7 \\ l8 \\ l9 \\ l10 \\ l11 \\ l12 \\ l13 \end{array} \begin{array}{c} \begin{array}{ccccccccccccc} l1 & l2 & l3 & l4 & l5 & l6 & l7 & l8 & l9 & l10 & l11 & l12 & l13 \end{array} \\ \begin{pmatrix} -x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3s & 3s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3s & 3s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -d & d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -y & y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -5s & 5s & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -5s & 5s & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4d & 4d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w & w & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -s & s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2s & -2s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2s & 2s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -d \end{pmatrix} \end{array}$$

Box II.

Table 6
Metrics variables.

Variable	Description
w	s + e
x	2s+e
y	5s+7e
z	5s+8e

$$q_{ij} = \begin{cases} q(P_i, P_j) = \sum\limits_{P_i \xrightarrow{\theta_k} P_j} \$(\theta_k) & \text{if } i \neq j \\ -\sum\limits_{\substack{j=1 \\ j \neq i}}^{n} q_{ij} & \text{if } i = j \end{cases} \quad (1)$$

System performance metrics become understandable since they are finite and cyclic in Eq. (2), these measurements of process P are obtained by utilizing the stationary probability distribution Π for the CTMC and bounding it with P.

$$\Pi Q = 0 \; and \; \sum_{i=0}^{n} \Pi(x_i) = 1. \quad (2)$$

The answers to the systems' linear equations are the stationary distributions for each system. Associating a reward structure, allows one to evaluate the performance of a procedure P [29]. The reward structure of a process P is a vector of rewards with the same number of components as the number of derivatives of P. Performance measurements for a process P were computed from information and the stationary distribution Π of the $CTMC$. Eq. (3) is used to calculate the total reward of a process P [26].

$$R(P) = \sum_{P_i \in d(P)} \rho P_i \; X \; \Pi(P_i). \quad (3)$$

Therefore, add the values of Pi, multiply by the matching reward structure with the encryption algorithm used and the time spent for summation. Whereas comparing the non-zero reward value against the rate of the relevant transition [30], one can determine the system's throughput in terms of the quantity of work completed per unit time [25].

7.1.2. Parameters and metrics

To assess the performance of 5G protocols, Markov chains and other mathematical techniques are used after specifying the protocols, describing the cost function and the semantics of the labelled enhanced operation. The stationary Markov chain distributions $\Pi_i = (X_0, \ldots, X_{n-1})(i = 1, 2,$ and $n = 6, 8)$ for the protocols, using the following linear equation to serve as the solution for each protocol as Eq. (4) [26].

$$\Pi Q = 0 \; and \; \sum_{i=0}^{n-1} X_i = 1. \quad (4)$$

$$\Pi_1 = \left[\frac{M}{x}, \frac{M}{3s}, \frac{M}{3s}, \frac{M}{d}, \frac{M}{z}, \frac{M}{5s}, \frac{M}{5s}, \frac{M}{5d}, \frac{M}{x}, \frac{M}{2s}, \frac{M}{d}, \frac{M}{2s}, \frac{M}{s}\right] \tag{5}$$

$$\Pi_2 = \left[\frac{N}{x}, \frac{N}{3s}, \frac{N}{3s}, \frac{N}{d}, \frac{N}{y}, \frac{N}{5s}, \frac{N}{5s}, \frac{N}{4d}, \frac{N}{w}, \frac{N}{2s}, \frac{N}{2s}, \frac{N}{2s}, \frac{N}{d}\right] \tag{6}$$

EAP-AKA' Protocol: Tables 5 and 6 illustrate the cost and transition association, while Fig. 9 displays the transition state and labels. To ensure that the transition system has stationary distributions and has both finite and cyclic beginning states, the following generator matrix $Q1$ = CTMC (EAP-AKA') is generated, and the stationary distribution is Π_1 (see Box I) and Eq. (5), where $M = 30s + 10e + 7d$.

Where:

$$M = \frac{13M}{30s + 10e + 7d} \tag{7}$$

5G-AKA Protocol: Tables 5 and 6 illustrate the cost and transition association while Fig. 10 displays the status of the transition and the labels. The following generator matrix $Q2$ = CTMC (5G-AKA) is generated, and the stationary distribution is $\Pi 2$ (see Box II) and Eq. (6), where $N = 29s + 9e + 6d$.

Where:

$$N = \frac{13N}{29s + 9e + 6d} \tag{8}$$

7.1.3. Analytical results

The cost of each protocol is analysed using efficiency and throughput metrics.

Efficiency: To measure the effect of cryptographic primitives on the performance of the protocol, value 0 is assigned to any other transition and value 1 to any transition in which the decryption is enabled, therefore the following is given value 1:

1. the 4th, 8th, 11th transitions in EAP-AKA'
2. the 4th, 8th, 13th transitions in 5G-AKA

The protocols' performance as measured by the metric R is as follows:

$$R(EAP - AKA') = \frac{M}{7d} \tag{9}$$

$$R(5G - AKA) = \frac{N}{6d} \tag{10}$$

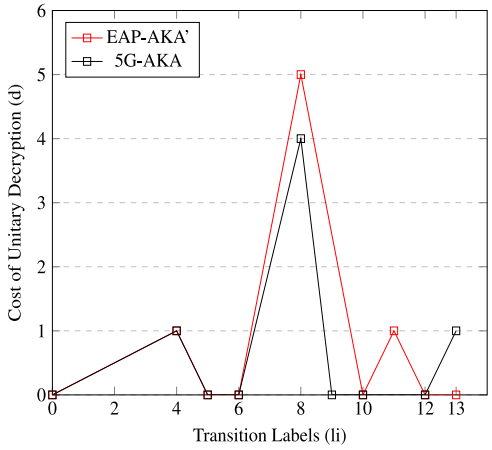
If the encryption algorithm is the same for both protocols and the same quantitative performance evaluation metric is employed, it is easy to demonstrate how one protocol is expensive than the other for every positive using s, d, and e.

Throughput: This is the outcome of linking a transition reward to an activity's rate and transition, the CTMC is cyclic, and each label represents a different transaction. A transition reward is assigned to the rate of the last protocol communication but nothing to all other communications, then throughput of the protocol is calculated. Additionally, the results show a correlation between a cryptographic algorithm's energy usage and its time complexity [25].

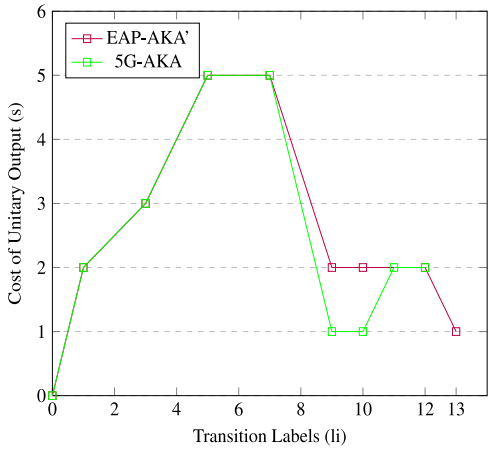
The following are calculations for the reward structure and total rewards:

$$\begin{aligned} \rho 1 &= (0.....C_{12}), \quad (C_{11}) = d, \\ R(EAP - AKA') &= \frac{3d}{30s + 10e + 7d} \end{aligned} \tag{11}$$

$$\begin{aligned} \rho 2 &= (0.....C_{12}), \quad (C_{13}) = d, \\ R(5G - AKA) &= \frac{2d}{29s + 9e + 6d} \end{aligned} \tag{12}$$



(a) Effeciency



(b) Throughput

Fig. 11. Protocols' efficiency and throughput based on analytical approach.

Table 7
Performance evaluation of the protocols.

Protocols	Efficiency	Throughput
EAP-AKA'	$\frac{EAP-AKA'}{7d}$	$\frac{d}{30s+10e+7d}$
5G-AKA	$\frac{5G-AKA}{6d}$	$\frac{d}{29s+9e+6d}$

7.1.4. Performance analysis 1

The results in Table 7 and Fig. 11 show that 5G-AKA protocol is more effective and has greater throughput than EAP-AKA' protocol, it uses more messages between the UE and AUSF. Using the continuous time method allows the assessment of the protocol's performance based on its stationary distribution, if any. Additionally, ProVerif and applied *pi* calculus are applied for security characteristics and bisimilarity labelling, respectively [25,26].

7.2. Simulation performance evaluation approach

For comparability of the protocol effectiveness, a simulation approach is used for further evaluation, whereby the results and compared and analysed. This is achieved by simulating the protocol in NS-3, which is installed and configured on an Ubuntu Linux virtual machine in a VirtualBox environment running on a Windows computer [25].

The setup emulates the 5G non-standalone implementation with 5G radio technology and LTE core network based on the NS-3 5G mmWave module [31,32]. To model 5G communication, the nodes, net

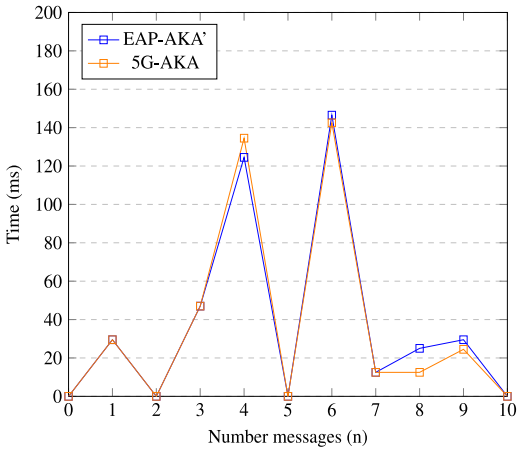


Fig. 12. Protocols computational cost based on simulation approach.

Table 8
Cryptographic operations compute time.

Notation	Description	Computational time (ms)
T_E	execution	21.5
T_{Av}	authentication vector	33.5
T_{KDF}	key generation	12.0
T_h	hash digest function	5
T_{Se}	symmetric encoding	4
T_{Sd}	symmetric decoding	5.5
T_{Ae}	asymmetric encoding	8
T_{Ad}	asymmetric decoding	9.5
T_V	verification	12.5

Table 9
Cryptographic primitive length/size.

Cryptographic primitive	Value
Strings	32 bits
SQN	48 bits
Identity	64 bits
MAC	64 bits
Nonce	128 bits
Symmetric key	128 bits
Asymmetric key	256 bits
RES	256 bits
SHA256	256 bits

Table 10
Performance evaluation metrics.

Parameters	Values
Throughput	bits/milliseconds
Latency	milliseconds
m	messages primitive cost
n	total sum of m

device, applications, and topology helpers were adjusted to represent 5G-AKA and EAP-AKA' protocols entities. In addition, the security attributes, cryptographic operatives were defined using applications with a `sendMessage ()` function for protocol exchange messages [25,33].

7.2.1. Parameter and metrics

The protocols' computational and communication inputs are based on 3GPP recommended 5G cryptographic primitives and algorithms [1, 11].

7.2.2. Computational cost

To approximately measure the computational cost of the protocols, time cost of message attributes processing is defined and given

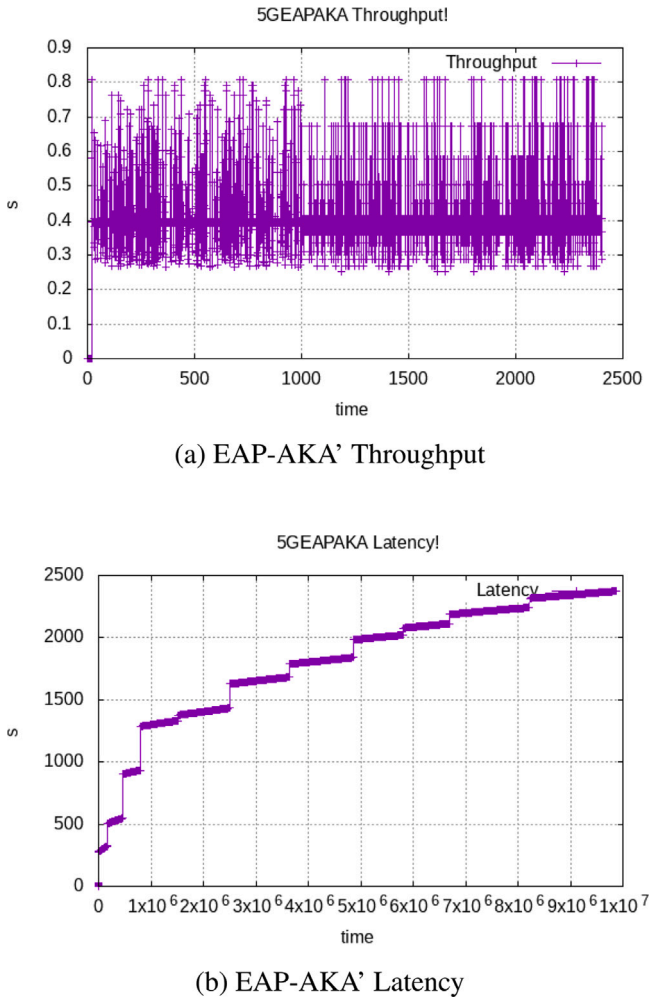


Fig. 13. EAP-AKA' protocol communication cost based on simulation approach.

estimated times in milliseconds (ms) as illustrated in Table 8. The variations in computational times of the protocols are shown in Fig. 12.

7.2.3. Communication cost

The protocol's cryptographic primitives and messages are utilized as parameters to assess the communication cost, defined in Tables 9 and 10. AMF, $synch\_fail$ and mac_fail messages are represented as strings. The term m refers to those cryptographic primitive numerical values part of messages between entities and used as input in NS-3 simulation, while n is the sum of all m in the protocol and $n = (m1, m2, m3, m4....)$ in bits. Therefore, EAP-AKA' = (384, 448, 448, 1082, 1082, 1082, 592, 592, 224, 32) and 5G-AKA = (384, 448, 448, 1738, 928, 672, 256, 256, 192, 576). Throughput $(bits/ms)$ and latency (ms) are performance metrics of n during the protocol simulation to determine the communication cost.

7.2.4. Simulation results

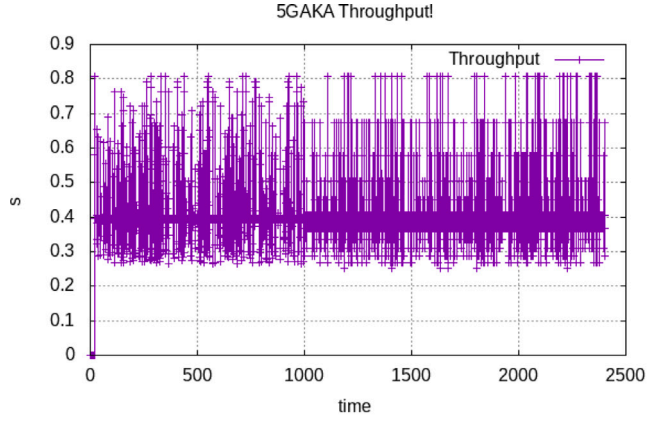
The quantified security protocol findings were extracted from the trace pcap and XML files produced by NS-3 simulation for both protocols, measuring throughput and latency based on metrics in Table 10.

7.2.5. Performance analysis 2

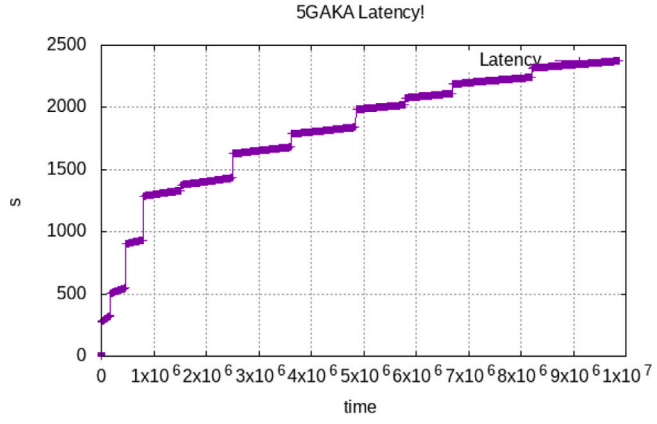
It is assumed that a successful authentication run was completed, hence, the resynchronization phase is not evaluated. Table 11 provides an overview of the computational cost of each security protocol, and Fig. 12 illustrates the comparison. It shows that EAP-AKA' compute

Table 11
Protocols computational cost.

Protocols	Time values	Time (ms)
EAP-AKA'	$T_E + 7T_{Se} + T_{Ae} + 7T_{Sd} + T_{Ad} + 1T_{Av} + 8T_{KDF} + 12T_V$	557.2
5G-AKA	$T_E + 6T_{Se} + T_{Ae} + 6T_{Sd} + T_{Ad} + 1T_{Av} + 8T_{KDF} + 2T_h + 11T_V$	542.5



(a) 5G-AKA Throughput



(b) 5G-AKA Latency

Fig. 14. 5G-AKA protocol communication cost based on simulation approach.

Table 12
Protocols communication cost.

Protocol	Number of messages (m)	Communication cost (bits) (n)
EAP-AKA'	10	5966
5G-AKA	10	5898

takes 557.2 ms compared to 545.6 ms for 5G-AKA. However, there is a slight difference in messages 4, 6, 8 and 9 of the protocols.

Additionally, the overall communication costs are listed in Table 12. Figs. 13 and 14 illustrate throughput and latency plot graphs that were created from NS-3 simulation with almost identical graphs. Both protocols have the same amount of messages $m = 10$, EAP-AKA' has a higher communication cost than 5G-AKA i.e., 5996 *bits n* compared to 5898 *bits n*. 5G-AKA' has a better performance in terms of throughput and latency but simulation produces comparable results due the similarity in number messages, primitives and entities used. The usage of MAC2 and a success message in EAP-AKA' increases the cost slightly, even though the hash of RES is only used in 5G-AKA.

8. Conclusion

The EAP framework and EAP-AKA' protocol specifications were presented in this article based on the 3GPP standard. All the security presumptions and requirements have been identified through interpretation and analysis of the 5G standard and the EAP-AKA' RFC. It formally analysed and verified the protocol using ProVerif utilizing automated reasoning about the security properties. It analysed the protocol for security guarantees, identified some vulnerabilities and provided recommendations. The performance of the EAP-AKA' protocol performance was evaluated and compared with the 5G-AKA protocol for effectiveness using analytical and simulation methods. These strategies considered the cost variables that may affect the protocol's design and efficiency in security implementation. Using these methods, future work will concentrate on verifying and evaluating future security protocol guarantees and effectiveness.

CRediT authorship contribution statement

Ed Kamya Kiyemba Edris: Conceptualization, Methodology, Investigation, Software, Formal Analysis, Validation, Resources, Writing – original draft, Editing, Visualization. **Mahdi Aiash:** Conceptualization, Methodology, Writing – review, Supervision. **Jonathan Loo:** Conceptualization, Writing – review, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] 3GPP. Security architecture; procedures for 5G system. Technical specification (TS), (3GPP TS 33.501 V17.4.1 (2022-01)). Third Generation Partnership Project; 2022, URL https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169.

[2] Blanchet B, Smyth B, Cheval V, Sylvestre M. ProVerif 2.01: automatic cryptographic protocol verifier, user manual and tutorial. In: Version from. 2020, URL https://prosecco.gforge.inria.fr/personal/bblanche/proverif.

[3] Zhang J, Yang L, Cao W, Wang Q. Formal analysis of 5G EAP-tls authentication protocol using ProVerif. IEEE Access 2020.

[4] Edris EKK, Aiash M, Loo J, Alhakeem MS. Formal verification of secondary authentication protocol for 5G secondary authentication. Int J Secur Netw 2021;16(4):223–34. http://dx.doi.org/10.1504/IJSN.2021.119379.

[5] Ajit M, Sankaran S, Jain K. Formal verification of 5G EAP-aka protocol. In: 2021 31st international telecommunication networks and applications conference. ITNAC, 2021, p. 140–6. http://dx.doi.org/10.1109/ITNAC53136.2021.9652163.

[6] Vollbrecht JR, Aboba B, Blunk LJ, Levkowetz H, Carlson J. Extensible authentication protocol (EAP). RFC Editor, IETF; 2004, URL https://tools.ietf.org/html/rfc3748.

[7] Arkko J, Lehtovirta V, Eronen P. Improved extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA'). RFC Editor,IETF; 2009, URL https://tools.ietf.org/html/rfc5448.

[8] Arkko J, Lehtovirta V, Torvinen V, Eronen P. Improved extensible authentication protocol method for 3GPP mobile network authentication and key agreementsw (EAP-AKA'). Tech. rep, (9048). IETF; 2021, http://dx.doi.org/10.17487/RFC9048, URL https://www.rfc-editor.org/info/rfc9048.

[9] Edris EKK, Aiash M, Loo J. An introduction of a modular framework for securing 5G networks and beyond. Network 2022;2(3):419–39. http://dx.doi.org/10.3390/network2030026.

[10] 3GPP. 3GPP system architecture evolution (SAE) system aspects, security aspects of non-3GPP accesses. Technical specification (TS), (3GPP TS 33.402 V16.0.0(2020-07)). Third Generation Partnership Project; 2020, URL https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2297.

[11] Edris EKK, Aiash M, Loo J. Formal verification and analysis of primary authentication based on 5G-AKA protocol. In: The third international symposium on 5g emerging technologies. 5GET 2020, Paris, France: IEEE; 2020, p. 256–61.

[12] Basin D, Dreier J, Hirschi L, Radomirović S, Sasse R, Stettler V. A formal analysis of 5G authentication. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018, p. 1383–96. http://dx.doi.org/10.1145/3243734.3243846.

[13] Meier S, Schmidt B, Cremers C, Basin D. The TAMARIN prover for the symbolic analysis of security protocols. In: Sharygina N, Veith H, editors. Computer aided verification, Vol. 8044. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013, p. 696–701. http://dx.doi.org/10.1007/978-3-642-39799-8_48, URL https://www.research-collection.ethz.ch/handle/20.500.11850/68108.

[14] Armando A, Basin DA, Boichut Y, Chevalier Y, Compagna L, Cuellar JR, et al. The AVISPA tool for the automated validation of internet security protocols and applications. Comput Aided Verif Proc 2005;3576:281–5, ID: wos000230755800027.

[15] Cremers C, Dehnel-Wild M. Component-based formal analysis of 5G-AKA: Channel assumptions and session confusion. In: Network and distributed system security symposium. NDSS, Internet Society; 2019.

[16] Caixia LIU, Xinxin HU, Shuxin LIU, Wei YOU, Yu ZHAO. Security analysis of 5G network EAP-AKÁprotocol based on Lowe's taxonomy. J Electron Inf Technol 2019;41(8):1800–7.

[17] van DB, Verdult R, de Ruiter J. Defeating IMSI catchers. In: Proceedings of the 22Nd ACM SIGSAC conference on computer and communications security, Vol. 2015. 2015, p. 340–51. http://dx.doi.org/10.1145/2810103.2813615, ID: acm2813615.

[18] Koutsos A. The 5G-AKA authentication protocol privacy. In: 2019 IEEE european symposium on security and privacy. IEEE; 2019, p. 464–79, ID: proquest2135414227.

[19] Enisa. Signalling security in telecom SS7/Diameter/5G. Tech. rep, Enisa; 2018, URL https://www.enisa.europa.eu/publications/signalling-security-in-telecom-ss7-diameter-5g.

[20] Menezes AJ, Oorschot PCV, Vanstone SA. Handbook of applied cryptography. Boca Raton, Florida, USA: CRC Press; 2018, Includes bibliographical references and index. ID: alma991001301199704781.

[21] Lowe G. A hierarchy of authentication specifications. In: Proceedings 10th computer security foundations workshop. IEEE; 1997, p. 31–43. http://dx.doi.org/10.1109/CSFW.1997.596782, ID: ieee_s596782.

[22] Dehnel-Wild M, Cremers C. Security vulnerability in 5G-AKA draft. Tech. rep, Department of Computer Science, University of Oxford; 2018.

[23] Engel T. Ss7: Locate. Track. Manipulate. In: Talk at 31st chaos communication congress. 2014.

[24] RIFS G. Diameter roaming security - proposed permanent reference document. Tech. rep., GSMA; 2016.

[25] Edris EKK, Aiash M, Loo J. Security protocol performance evaluation using applied Pi calculus and Markov chain based on 5G security. Comput Secur 2022, Submitted for Publication.

[26] Bodei C, Curti M, Degano P, Buchholtz M, Nielson F, Nielson HR, et al. Performance evaluation of security protocols specified in Lysa. Electron Notes Theor Comput Sci 2005;112:167–89.

[27] Abadi M, Blanchet B, Fournet C. The applied Pi calculus: Mobile values, new names, and secure communication. J ACM 2017;65(1):1–41, ID: hal_soai_HAL_hal_01423924v1.

[28] Stewart WJ. Introduction to the numerical solution of markov chains. Princeton University Press; 1994.

[29] Hillston J. A compositional approach to performance modelling, Vol. 12. Cambridge University Press; 2005.

[30] Bodei C, Curti M, Degano P, Buchholtz M, Nielson F, Nielson HR, et al. On evaluating the performance of security protocols. In: International conference on parallel computing technologies. Springer; 2005, p. 1–15.

[31] Nsnam. Ns-3 a discrete-event network simulator for internet systems. (NS-3.33). 2021.

[32] Mezzavilla M, Zhang M, Polese M, Ford R, Dutta S, Rangan S, et al. End-to-end simulation of 5G mmwave networks. IEEE Commun Surv Tutor 2018;20(3):2237–63. http://dx.doi.org/10.1109/COMST.2018.2828880.

[33] Banerjee S, Odelu V, Das AK, Chattopadhyay S, Park Y. An efficient, anonymous and robust authentication scheme for smart home environments. Sensors 2020;20(4):1215.