A Formal Analysis of The Modified 5G EAP-TLS Protocol

DOI:https://doi.org/10.54654/isj.v2i28.6385

Tran Thi Nga*, Nguyen Quoc Hung, Bui Thu Giang

***Abstract*—The 5G EAP-TLS protocol is one of the three protocols standardised by 3GPP for use in 5G networks. Although this protocol inherently ensures security, authentication, and data integrity, recent studies have shown that it still faces several vulnerabilities, including Man-in-The-Middle attacks, user impersonation, and replay attacks. This paper presents a detailed description of the steps in the modified EAP-TLS protocol, which addresses these issues by directly binding the digital certificate to the subscriber's SUCI identity to prevent user impersonation, binding the session key to both the certificate and the identity to ensure resistance against Man-in-The-Middle attacks, and introducing a nonce value to prevent the reuse of old packets in replay attacks. The paper employs the Proverif tool as a formal verification approach to evaluate the security of the modified 5G EAP-TLS protocol. The verification outcomes indicate that the proposed protocol satisfies the specified security properties, including the confidentiality of the session key (KSESSION), the subscriber identity (SUPI), the pre-master key (RPREKEY), as well as other relevant security requirements.**

***Tóm tắt*— Giao thức 5G EAP-TLS là một trong ba giao thức được 3GPP tiêu chuẩn hóa để sử dụng trong mạng 5G. Mặc dù giao thức này vốn đảm bảo tính bảo mật, xác thực và toàn vẹn dữ liệu, các nghiên cứu gần đây đã chỉ ra rằng nó vẫn phải đối mặt với một số lỗ hổng, bao gồm các cuộc tấn công Man-in-The-Middle, mạo danh người dùng và tấn công phát lại. Bài báo này trình bày mô tả chi tiết các bước trong giao thức EAP-TLS được sửa đổi, để giải quyết các vấn đề này bằng cách liên kết trực tiếp chứng chỉ số với định danh SUCI của người đăng ký để ngăn chặn mạo danh người dùng, liên kết khóa phiên với cả chứng chỉ và định danh để đảm bảo khả năng chống lại các cuộc tấn công xen giữa và giới thiệu giá trị nonce để ngăn chặn việc sử dụng lại các gói tin cũ trong các cuộc tấn công phát lại. Bài báo sử dụng công cụ Proverif như một phương pháp xác minh chính thức để đánh giá tính bảo mật của giao thức 5G EAP-TLS sửa đổi. Kết quả xác minh cho thấy giao thức được đề xuất đáp ứng các thuộc tính bảo mật đã được chỉ định, bao gồm tính bảo mật của khóa phiên (KSESSION), định danh người đăng ký (SUPI), the pre-master key (RPREKEY), cũng như các yêu cầu bảo mật liên quan khác.**

Keywords— 5G security; EAP-TLS; proverif; formal verification.

Từ khóa— An ninh mạng 5G; EAP-TLS; proverif; kiểm chứng hình thức.

I. Introduction

In the 5th Generation (5G) network architecture, security is a paramount concern driven by the massive proliferation of connected devices and the emergence of novel service types. The 3rd Generation Partnership Project (3GPP) has standardised three primary authentication protocols: Authentication and Key Agreement (5G-AKA), Extensible Authentication Protocol- Authentication and Key Agreement (EAP-AKA'), and Extensible Authentication Protocol - Transport Layer Security (EAP-TLS) [1 - 3]. Among these, EAP-TLS is a certificate-based authentication protocol that has been extensively adopted in enterprise wireless systems, Virtual Private Network (VPN) remote access, and secure IoT communication environments [4]. Its core purpose is to authenticate communicating parties and block unauthorised access. Nevertheless, when exposed to Man-In-The-Middle (MiTM) threats such as message modification, replay attacks, and identity

This manuscript was received on March 3, 2026. It was reviewed on June 1, 2026, revised on June 16, 2026 and accepted August 20, 2026.

*Corresponding author

spoofing-the entity authentication mechanism of EAP-TLS still reveals notable security weaknesses [5, 6].

In [7], the authors were the first to employ the Proverif tool to formally verify the 5G EAP-TLS 1.2 protocol. Their analysis revealed that the current protocol suffers from several vulnerabilities related to authentication. In addition, they proposed several potential countermeasures. However, no explicit verification results were provided to validate the effectiveness of these proposed solutions. Following a similar formal verification approach using Proverif, Naiguang Zhu and colleagues [8] modelled the 5G EAP-TLS 1.3 protocol. This work represents the first formal analysis of EAP-TLS 1.3 within 5G networks. Nevertheless, the study did not propose a concrete mechanism capable of comprehensively mitigating the identified vulnerabilities, but mainly focused on detecting weaknesses associated with privacy attacks, replay attacks, and authentication flaws, as well as several issues related to the TLS handshake and session authentication processes. Furthermore, the proposed approach was still unable to effectively resist certain attacks, including user impersonation attacks, Man-in-the-Middle (MITM) attacks, and session race-condition attacks. Min Shi and colleagues [9] subsequently utilised the Tamarin prover to conduct a formal security analysis of the 5G EAP-TLS protocol.

Compared with previous studies, the proposed work introduces a modification to the 5G EAP-TLS protocol that aims to strengthen the association between user identity and session key establishment. Specifically, the proposed approach incorporates the UE-generated nonce R_{UE} into both the initial SUCI message and the encrypted key transport phase, thereby establishing an additional cryptographic linkage between the subscriber identity and the resulting session key. This design is expected to enhance the protocol's resilience against replay attacks and user impersonation. Moreover, by reinforcing the binding between authentication messages and session context, the proposed mechanism may contribute to mitigating session confusion attack which have not been explicitly addressed in [7].

In this study, to illustrate how the proposed protocol modifications are secure and resistant to several identified attacks, advanced tools now employ formal methods, mathematical techniques, and automated analysis to rigorously verify the safety and robustness of protocols. The tools pinpoint potential vulnerabilities, such as the Proverif tool, which has the capability to model protocols using symbolic models and verify security properties [10 - 14]. We model a modified version of the 5G EAP-TLS protocol aimed at rectifying the aforementioned security property violations. Specifically, we restructure the messaging within the key exchange phase; rather than relying solely on server-side random parameters or retaining the default structure.

Verification results comparing the modified protocol model on Proverif with the original 5G EAP-TLS show that modifying this parameter contributes to mitigating the weak authentication issue, mitigates identity-session misbinding attacks, and ensures UE non-repudiation, thereby enhancing the security framework for the EAP-TLS protocol in 5G networks.

The remainder of the paper is structured as follows. Section II presents the preliminaries used throughout the subsequent sections. Section III introduces the formal security analysis conducted with Proverif. Section IV describes the verification results and provides related discussions. Finally, Section V concludes the paper.

II. PRELIMINARIES

In this section, we introduce the architecture of a private 5G network and give a detailed description of the modified 5G EAP-TLS protocol based on TS 33.501 [1-3, 5].

To facilitate understanding and reference, we summarise the notations in TABLE I.

A. 5G Network Structure

The 5G architecture is fundamentally composed of three primary entities: the UE

(User Equipment), the SN (Serving Network), and the HN (Home Network). Figure 1 illustrates the key components of the 5G network architecture.

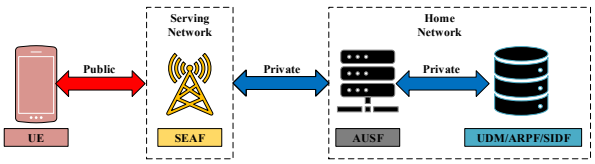

Figure 1. Key components of the 5G network architecture

The UE functions as the mobile device that establishes a connection with the HN through a radio interface. Embedded within each UE is a Universal Integrated Circuit Card (UICC), which houses at least one Universal Subscriber Identity Module (USIM). This module is critical for security, as it securely stores the Subscription Permanent Identifier (SUPI), the pre-shared cryptographic key established between the subscriber and the HN, and the HN's public key.

The SN comprises two essential elements necessary for connectivity and intermediary security. The first is the gNB, which serves as the next-generation base station, succeeding the eNodeB found in 4G networks. The second component is the SEAF, which acts as an intermediary anchor between the UE and the HN. While the SEAF possesses the authority to reject an authentication request from the UE locally, it ultimately relies on the verification and acceptance decisions made by the HN to successfully authenticate the device.

The HN houses the core functions responsible for subscriber management and final authentication decisions. It contains the following entities:

- AUSF belongs to the home network responsible for authenticating and negotiating communication keys with the UE. AUSF stores a certificate certAUSF, the name of the binding UDM, and a list of available serving networks/SEAFs.

- Unified data management (UDM) is an entity that supplies the main functions corresponding to data management, such as Authentication Credential Repository and Processing Function (ARPF) and Subscription Identifier De-concealing Function (SIDF). The ARPF chooses an authentication method among the three available options: 5G AKA, EAP-AKA, or EAP-TLS. 5G AKA and EAP-AKA' are 5G primary authentication methods, based on a pre-shared key mechanism, and only have a slight difference in the message flow and key derivation ways.

5G EAP-TLS is based on a public-key certificate mechanism and is dedicated to the private 5G network. The option is selected on the basis of the subscriber identity and configured policy. The SIDF is responsible for the decryption of the Subscription Concealed Identifier (SUCI) to obtain the SUPI [1, 7-9].

B. Modified 5G EAP-TLS Protocol

Prior to the initiation of this protocol session, the UE has already completed the carrier service subscription. Consequently, the UE's unique and private identifier (SUPI) is stored in the UDM's database, and the UDM's public key is provisioned in the UE. Moreover, each subscriber credential is stored in no more than one UDM, while the UDM's public key is embedded in every UE.

Based on the formal analysis results highlighting the lack of binding in the standard specification [7- 9]. In this section, we propose a design for a modified 5G EAP-TLS protocol. The core objective of this proposal is to establish a rigorous identity-key binding mechanism to mitigate the risk of MITM attacks. Our modified protocol leverages the UE nonce R_{UE} as a security anchor. Specifically, R_{UE} (which is generated and protected within the initial SUCI message) is embedded as part of the encrypted payload in the Key Transport message. This modification cryptographically binds the Session Key to the user's initial identity. Consequently, the HN can definitively verify that the entity transmitting the key is the same entity that initiated the identity request, thereby ensuring strong mutual authentication and session integrity. Furthermore, encrypting the TLS-

START message together with the UE generated random value R_{UE} may provide an additional mechanism for preserving session context consistency. Although the primary mitigation against session confusion is achieved through the integration of R_{UE} into the session key derivation process, returning R_{UE} within the encrypted TLS-START message (step 2 Figure 2) allows the UE to verify that the received response corresponds to the same authentication instance before initiating the EAP-TLS handshake. Consequently, any mismatch between authentication sessions may be detected at an earlier stage. Meanwhile, the protocol of [8] will continue to perform part or all of the TLS handshake process until it is detected.

Figure 2 illustrates the authentication flow of the revised 5G EAP-TLS protocol. The components highlighted in red denote modified steps, thereby clearly distinguishing the proposed scheme from previous approaches.

TABLE I .NOTATIONS

Notations	Descriptions
SUPI	Subscription Permanent Identifier
R_{UE}	A random nonce generated by User Equitment
R_{AUSF}	A random nonce generated by AUSF
R_{PREKEY}	A random nonce generated by UE
pk_{UDM}	UDM's Public Key
pk_{UE}	UE's Public Key
pk_{AUSF}	AUSF's Public Key
SSK_{UE}	UE's Private Signing Key
$K_{SESSION}$	Session Key
K_{SEAF}	Key secure between UE and SEAF
SEAFN	Name of SEAF

1. The UE initiates the session by generating a fresh binding nonce R_{UE}. Instead of sending the identity in isolation, the UE constructs a SUCI by encrypting the tuple $\{SUPI, R_{UE}\}$ using the HN public key pk_{UDM} and transmits it to the SN (SEAF).

2. Upon receiving the initial request, the SEAF appends its identifier SEAFN to the packet and forwards the authentication request to the AUSF in the HN.

3. The AUSF verifies the SN's legitimacy. If valid, it relays the message to the UDM. The UDM invokes its private key to decrypt the SUCI, extracting both the subscriber identity SUPI and the binding nonce R_{UE}.

4. After validating the subscriber's identity, the UDM responds to the AUSF. Crucially, it transfers the decrypted R_{UE} to the AUSF, establishing the necessary context for the subsequent binding verification.

5. The AUSF sends TLS_START a message via the SEAF to the UE, formally triggering the TLS handshake.

6. The UE generates a new nonce R_{UE1} and proposes supported cryptographic algorithms, encrypting the tuple $\{R_{UE1}, TLS\_START\}$ using the HN public key pk_{AUSF}. This message along with the information about its supported algorithms Methods_UE is routed through the SEAF to the AUSF.

7. The AUSF responds with its own nonce R_{AUSF}, encrypting the tuple $\{R_{AUSF}, R_{UE1}\}$ using the UE public key pk_{UE}, the server certificate Certificate_AUSF, and the selected cipher suite Methods_AUSF.

8. The UE validates the AUSF certificate. Upon success, it generates a new nonce R_{PREKEY}, which is called pre-master key, and derives a session key $K_{SESSION}$ using R_{PREKEY}, R_{UE1} and R_{AUSF}. To enforce the binding mechanism, the UE concatenates the pre-master key with the initial nonce R_{UE} and encrypts this pair using the AUSF's public key pk_{AUSF}. This encrypted payload, along with the session signature and hash, is sent to the AUSF.

9. The AUSF decrypts the received packet. It extracts the candidate pre-master key and the check-value. The AUSF strictly enforces a comparison between this check-value and the R_{UE} received earlier from the UDM (in Step 4). The authentication proceeds only if these values match, confirming that the key generator is the legitimate identity owner.

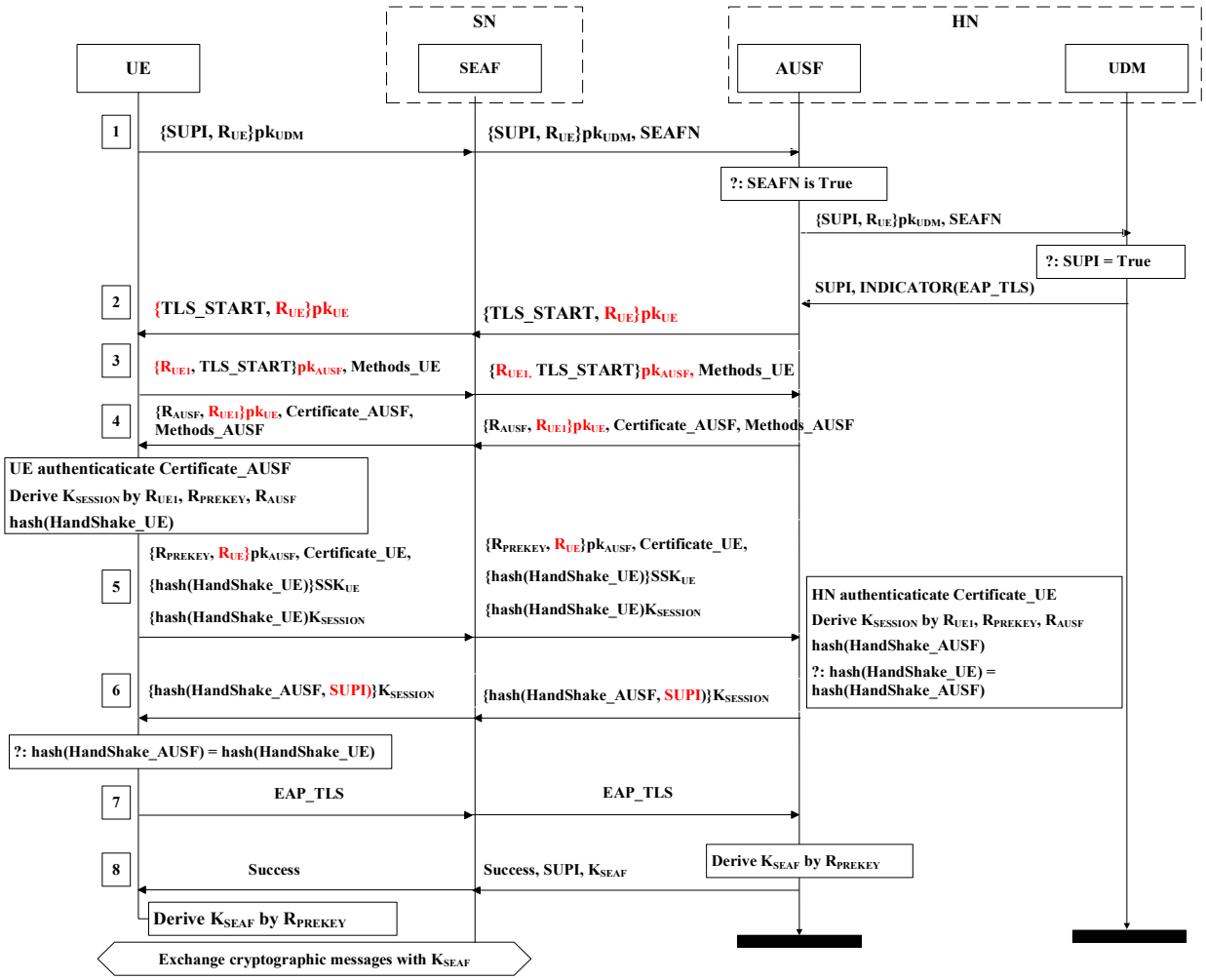

Figure 2. The modified 5G EAP-TLS authentication protocol

10. The AUSF verifies the integrity signature. If correct, it computes the session key $K_{SESSION}$, encrypts the handshake hash, and sends to the UE. The UE verifies the AUSF's message by compare the hash value. If correct, it transmits an message EAP_TLS indicating success.

11. The AUSF derives a new key K_{SEAF} from the established R_{PREKEY} and securely pushes it, along with the Success message and SUPI, to the SEAF.

12. The SEAF forwards the Success message to the UE. The UE derives the matching K_{SEAF} in the same way as AUSF, completing the secure bootstrap. The key K_{SEAF} will then be used to secure the subsequent communications between the user equipment and the networks.

III. Fomal verification using Proverif

This section outlines the overview of Proverif, the threat model and the security requirements, followed by a description of the formal model of the modified 5G EAP-TLS protocol constructed using Proverif.

The input language supports the specification of cryptographic protocols, their associated security objectives, and the formal verification of claimed security properties [10, 12].

Proverif enables protocol analysis with an unbounded number of sessions by employing special approximation techniques to achieve this capability. To verify unbounded sessions for certain classes of protocols, it utilises over-approximation techniques, including the abstraction of fresh nonce generation. When analysing a protocol, Proverif may determine

that a property is false and produce an attack trace, prove that the property holds, report that the property cannot be established, or fail to terminate. It is also capable of automatically exploring the full state space of a protocol, or at least a substantial portion of it. In addition, it generates detailed outputs containing English-like derivations and traces that help users understand how an attacker might compromise the security properties under consideration. Proverif also provides an interactive mode that allows the simulation of process execution runs [11 - 13].

As illustrated in the structure of Proverif in Figure 3, users define the protocol using a dialect of applied Pi calculus enriched with cryptographic primitives. The tool then automatically translates this input into Horn clauses and performs resolution algorithms to determine if the specified security goals are met.

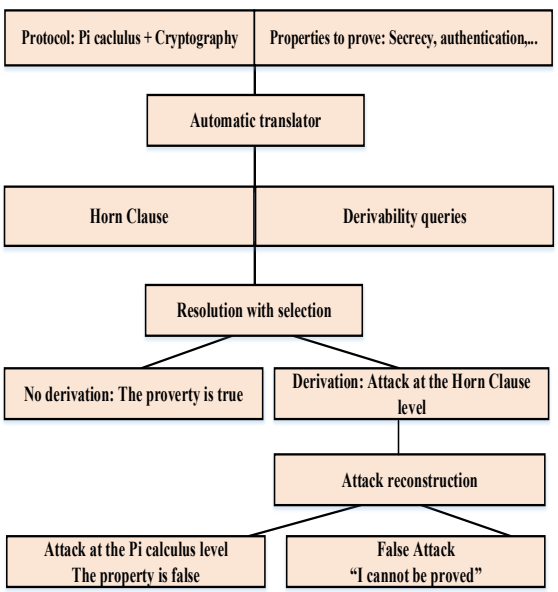

Figure 3. Structure of Proverif

B. Threat model and security requirements

1) Threat model: The model employed in this study is the standard Dolev-Yao adversary model [14, 15], which provides a rigorous framework for evaluating protocol security in an open network environment. Within this framework, we assume that the adversary possesses complete control over the public communication channels, particularly the air interface between the UE and the SN. Consequently, the attacker is capable of eavesdropping, intercepting, modifying, and injecting messages at will, as well as replaying captured traffic to impersonate legitimate network functions. Despite these extensive capabilities, the analysis is constrained by the "perfect cryptography" assumption. This implies that the underlying cryptographic primitives, such as symmetric/asymmetric encryption and digital signatures, are mathematically unbreakable; therefore, the adversary cannot decrypt messages or forge cryptographic proofs without obtaining the necessary secret keys through protocol interactions.

2) Security Requirements for The Modified 5G EAP-TLS Protocol: The protocol must ensure robust mutual authentication between the UE and the HN via the SN. This process involves subscription authentication, serving network authentication, and authorisation checks.

Based on these requirements, we formalize two critical authentication properties:

- The HN and the UE must agree on each other's identities upon the successful completion of the protocol.

- The HN and the UE must agree on the security parameters R_{PREKEY} after successful termination.

In addition to authentication, the protocol must guarantee the confidentiality of user data and identity over the radio interface. This implies that sensitive parameters exchanged during the key agreement phase must be strictly protected against eavesdropping. We define three essential secrecy properties:

- The attacker cannot obtain the identity SUPI from an honest subscriber.

- The attacker cannot obtain the pre-master key R_{PREKEY} of an honest subscriber.

- The attacker cannot obtain the session key $K_{SESSION}$ of an honest subscriber.

C. Modeling the Modified 5G EAP-TLS Protocol

1) Cryptographic primitives: We model cryptographic primitives under the perfect

cryptography assumption using constructors and destructors:

- Symmetric encryption: Modeled by the constructor $senc$ and destructor $sdec$. The reduction rule $sdec(senc(m,k),k) = m$ ensures that decryption only succeeds with the identical key used for encryption.

- Asymmetric encryption: Defined by key pairs $skey$ and $pkey$. The destructor $adec$ recovers the message m from $aenc(m, pk(sk))$ only if the corresponding private key sk is provided.

- Hashing and type conversion: The function h represents a one-way collision-resistant hash function with no inverse. The function b_to_k acts as a type converter to transform hash outputs into valid symmetric keys.

- Digital signatures: The constructor $sign$ creates a signature using a secret key $sskey$. The destructor $checksign$ verifies integrity, returning the message m only if the signature matches the public key spk.

```
type key.
fun senc(bitstring, key): bitstring.
reduc forall m: bitstring, k: key; sdec(senc(m,k),k) = m.
type skey.
type pkey.
fun pk(skey): pkey.
fun aenc(bitstring, pkey): bitstring.
reduc forall m: bitstring, sk: skey; adec(aenc(m,pk(sk)),sk) = m.
fun h(bitstring,bitstring,bitstring): bitstring.
fun b_to_k(bitstring):key [data,typeConverter].
type sskey.
type spkey.
fun spk(sskey): spkey.
fun sign(bitstring, sskey): bitstring.
reduc forall m: bitstring, k: sskey; getmess(sign(m,k)) = m.
reduc forall m: bitstring, k: sskey;
checksign(sign(m,k),spk(k)) = m.
```

2) Channels model and Constants: We define the channels model and the parameters used within the protocol.

- The channel $c1$ represents the public air interface from the UE to the SN. It is declared as a free name without the private attribute, granting the Dolev-Yao adversary full control to intercept, inject, and modify messages. The channels $c2$ and $c3$ represent the secure backhaul links from SEAF to AUSF and from AUSF to UDM. The $[private]$ attribute ensures these channels are invisible to the adversary, simulating trusted internal connections.

- $CertAUSF$ and $CertUE$ are declared as free names, representing the digital certificates of the HN and the UE. These are public values known to all parties, including the attacker.

- $SUPI$, $RPREKEY$, and $KSESSION$ are declared with the $[private]$ attribute. This denotes that these values are initially known only to the honest participants.

```
free c1:channel.
free c2:channel [private].
free c3:channel [private].
free CertAUSF:bitstring.
free CertUE:bitstring.
free SUPI:bitstring [private].
free KSESSION:KEY [private].
free RPREKEY:bitstring [private].
```

3) Security properties verification: The security analysis in Proverif is conducted by defining a comprehensive set of goals that encompass both confidentiality and integrity. Confidentiality is modeled as a reachability problem, where we verify whether the adversary can derive sensitive terms such as the permanent identity ($SUPI$) or the pre-master secret ($PREKEY$) from the public channel. Simultaneously, authentication and key agreement are captured through injective correspondence assertions ($inj - event$), which enforce a strict one-to-one relationship between protocol events (such as term and accepts) to effectively detect impersonation and replay attacks. By combining these approaches, we define the following queries to verify that the adversary cannot learn secret keys and that the UE and HN correctly authenticate each other and agree on the session parameters.

```
query attacker(RPREKEY).
query attacker(KSESSION).
query attacker(SUPI).
```

```
event SendRPREKEY(bitstring).
event AcceptRPREKEY(bitstring).
query x:bitstring; inj-event(AcceptRPREKEY(x))
==> inj-event(SendRPREKEY(x)).
event AcceptsUE(bitstring).
event TermAUSF(bitstring).
query x:bitstring; inj-event(TermAUSF(x)) ==> inj-
event(AcceptsUE(x)).
event AcceptsAUSF(bitstring).
event TermUE(bitstring).
query x:bitstring; inj-event(TermUE(x)) ==> inj-
event(AcceptsAUSF(x)).
```

4) User Equipment Process: In the proposed architecture, the UE process is modified to incorporate an identity-key binding mechanism. This involves two critical enhancements compared to the standard 5G EAP-TLS protocol:

- Session identity initialization: At the inception of the protocol, the UE generates a fresh nonce R_{UE} and transmits it alongside the $SUPI$ within the encrypted $SUCI$ $(aenc((SUPI, R_{UE}), pk_{UDM}))$. This explicitly establishes a unique "session fingerprint" from the very first message.

- Session key binding: During the key transport phase, instead of transmitting the $RPREKEY$ in isolation, the UE concatenates it with the initial nonce R_{UE} before encryption $(aenc((RPREKEY, R_{UE}), pk_{AUSF}))$. By cryptographically binding the $RPREKEY$ to the R_{UE}, the protocol ensures that only the legitimate entity can construct a valid authentication packet, thereby strengthening resistance against identity-session misbinding and MITM-style attacks.

```
let UE(pkAUSF:pkey, pkUDM:pkey,
spkAUSF:spkey, spkUE:spkey, sskUE:sskey,
skUE:skey) =
   new RUE:bitstring;
   out(c1, aenc((SUPI, RUE), pkUDM));
   in(c1, Y:bitstring);
   let (START_X:bitstring, RUE_CHECK:bitstring) =
adec(Y, skUE) in
   let (=RUE) = RUE_CHECK in
   new RUE1:bitstring;
   out(c1, aenc((RUE1, START_X), pkAUSF));
   in(c1, (X:bitstring, certAUSF_X:bitstring));
   let (RAUSF_X:bitstring, RUE1_CHECK:bitstring)
= adec(X, SKUE) in
   let (=RUE1) = RUE1_CHECK in
   let (=certAUSF) = certAUSF_X in
   new RPREKEY:bitstring;
   let A = h(RUE1, RAUSF_X, RPREKEY) in
   let KSESSION = b_to_k(A) in
   let HSUE = (START_X, RUE1, RAUSF_X,
certAUSF_X) in
   event AcceptsUE(RUE1);
   event SendRPREKEY(RPREKEY);
   out(c1, (aenc((RPREKEY, RUE), pkAUSF), certUE,
sign(HSUE, sskUE), senc(HSUE, KSESSION)));
   in(c1, HSAUSF_X:bitstring);
   let (HSUE_CHECK:bitstring,
SUPI_CHECK:bitstring) = sdec(HSAUSF_X,
KSESSION) in
   let (=HSUE) = HSUE_CHECK in
   let (=SUPI) = SUPI_CHECK in
   new EAPM:bitstring;
   out(c1, EAPM);
   in(c1, SUCM_X:bitstring);
   event TermUE(RAUSF_X).
```

5) Serving Network Process: The SEAF is modeled as a transparent intermediary that bridges the communication between the User Equipment on the public channel $c1$ and the Core Network on the secure channel $c2$. In this model, the SEAF performs no cryptographic operations on the EAP-TLS payload. Its primary function is to relay messages back and forth. However, in the initial step, it appends its identity $SEAFN$ to the user's request before forwarding it to the AUSF. This architecture is crucial for the formal analysis as it exposes the protocol to potential MITM adversaries operating on the public air interface $c1$, while maintaining the assumption that the internal backhaul link $c2$ remains secure.

```
let SEAF(pkAUSF:pkey, pkUDM:pkey,
spkAUSF:spkey, spkUE:spkey, SEAFN:bitstring) =
   in(c1, X1:bitstring);
   out(c2, (X1, SEAFN));
   in(c2, X2:bitstring);
   out(c1, X2);
   in(c1, X3:bitstring);
   out(c2, X3);
   in(c2, (X4:bitstring, X5:bitstring));
   out(c1, (X4, X5));
   in(c1, (X6:bitstring, X7:bitstring, X8:bitstring,
X9:bitstring));
   out(c2, (X6, X7, X8, X9));
```

```
  in(c2, X10:bitstring);
  out(c1, X10);
  in(c1, EAPM_X:bitstring);
  out(c2, EAPM_X);
  in(c2, SUCM_X:bitstring);
  out(c1, SUCM_X).
```

6) Home Network Process: In the proposed architecture, the network side authentication logic is distributed between the UDM and the AUSF to ensure session integrity. The UDM process acts as the initial root of trust; it decrypts the $SUCI$ using its private key to extract the subscriber identity and, most importantly, the nonce R_{UE} which serves as the unique session identifier before forwarding this context to the AUSF. The AUSF process orchestrates the EAP-TLS handshake. The core enhancement lies in the key verification phase: upon receiving the key transport message from the UE, the AUSF decrypts the payload and structurally validates the presence of the Pre-master secret paired with the nonce R_{UE}. This binding check ($let \ldots = adec(\ldots)$) ensures that the established session key is cryptographically bound to the initial user identity, thereby mitigating MITM attacks attempting to substitute the key.

```
let AUSF(skAUSF:skey, pkUE:pkey, spkUE:spkey,

  in(c2, (SUPI_X:bitstring, SEAFN_X:bitstring));
  if SEAFN_X = SEAFN then
    out(c3, (SUPI_X, SEAFN_X));
    in(c3, (START_X:bitstring,
RUE_BINDING:bitstring));
    out(c2, aenc((START_X, SUPI_X), pkUE));
    in(c2, V:bitstring);
    let (RUE1_X:bitstring,
START_CHECK:bitstring) = adec(V, skAUSF) in
    let (=START_X) = START_CHECK in
    new RAUSF:bitstring;
    out(c2, (aenc((RAUSF, RUE1_X), pkUE),
certAUSF));
    in(c2, (Y:bitstring, CertUE_X:bitstring,
T:bitstring, Z:bitstring));
    let (=CertUE) = CertUE_X in
    let (RPREKEY_X:bitstring,
RUE_CHECK:bitstring) = adec(Y, skAUSF) in
    if RUE_CHECK = RUE_BINDING then
      let B = h(RUE1_X, RAUSF, RPREKEY_X) in
      let KSESSION_X = b_to_k(B) in
      new HSAUSF:bitstring;
      let HSAUSF = (START_X, RUE1_X, RAUSF,
certAUSF) in
      let HSAUSF_CHECK:bitstring = sdec(Z,
KSESSION_X) in
      let (=HSAUSF) = HSAUSF_CHECK in
      let SIGNATURE_CHECK:bitstring =
checksign(T, SPKUE) in
      let (=HSAUSF) = SIGNATURE_CHECK in
      event AcceptRPREKEY(RPREKEY_X);
      event AcceptsAUSF(RUE1_X);
      out(c2, senc((HSAUSF, SUPI_X),
KSESSION_X));
      in(c2, EAPM_X:bitstring);
      new SUCM:bitstring;
      out(c2, SUCM);
      event TermAUSF(RUE1_X).

let UDM(skUDM:skey) =
  in(c3, (X:bitstring, SEAFN_X:bitstring));
  let (SUPI_X:bitstring, RUE_X:bitstring) = adec(X,
skUDM) in
  if SUPI_X = SUPI then
    new START:bitstring;
    out(c3, (START, RUE_X)).
```

7) Main Process: The main process serves as the initialization environment for the formal simulation. It begins by generating fresh cryptographic material, including private encryption keys and digital signature keys for all participating entities (UE, AUSF, and UDM). To accurately model a Dolev-Yao adversary with full knowledge of the Public Key Infrastructure, the system immediately broadcasts all derived public keys over the public channel $c1$. Finally, the protocol execution is defined as the parallel composition of the UE, AUSF, and UDM processes. The use of the replication operator (!) triggers an unbounded number of instances for each process, allowing Proverif to exhaustively verify security properties across multiple concurrent sessions and detect potential replay attacks.

```
process
new skUE:skey;
new skAUSF:skey;
new skUDM:skey;
new sskAUSF:sskey;
new sskUE:sskey;
```

```
new SEAFN:bitstring;
let pkUE = pk(skUE) in
out(c1, pkUE); out(c2, pkUE); out(c3, pkUE);
let pkAUSF = pk(skAUSF) in
out(c1, pkAUSF); out(c2, pkAUSF); out(c3,
pkAUSF);
let pkUDM = pk(skUDM) in
out(c1, pkUDM); out(c2, pkUDM); out(c3,
pkUDM);
let spkAUSF = spk(sskAUSF) in
out(c1, spkAUSF); out(c2, spkAUSF); out(c3,
spkAUSF);
let spkUE = spk(sskUE) in
out(c1, spkUE); out(c2, spkUE); out(c3, spkUE);
( (!UE(pkAUSF, pkUDM, spkAUSF, spkUE, sskUE,
skUE))
| (!SEAF(pkAUSF, pkUDM, spkAUSF, spkUE,
SEAFN))
| (!AUSF(skAUSF, pkUE, spkUE, sskAUSF,
spkAUSF, SEAFN))
| (!UDM(skUDM)) )
```

IV. VERIFICATION RESULTS AND DISCUSSION

We utilize Proverif version 2.05 as the verification engine, conducting experiments on a PC with Windows 11 Pro OS, an Intel(R) Core i5-6200U CPU operating at 2.4 GHz, and 16.0 GB of RAM.

The output of the EAP-TLS modified protocol is shown in Figure 4. This result demonstrates the modified protocol's robust resistance to MiTM attacks, replay attacks, and impersonation attacks. First, the secrecy queries for RPREKEY, KSESSION, and SUPI are all validated as true, indicating that an adversary cannot obtain these critical values. This secrecy, combined with the injective correspondence properties shown in the queries, ensures that each acceptance event by a protocol entity uniquely corresponds to a legitimate message actually generated by its peer. Consequently, a MiTM attacker cannot alter, inject, or relay messages to mislead either party into accepting forged keys or identities. Furthermore, the injective authentication properties such as inj-event(TermAUSF(x)) ==> inj-event(AcceptsUE(x)) confirm that every session is bound to a fresh and unique execution trace. This strictly prevents replay attacks, since an adversary cannot reuse previously recorded protocol messages to trigger new authentication events. Finally, the confidentiality of SUPI and KSESSION session keys, together with successful authentication correspondences, ensures that the attacker lacks both the credential information and the protocol interaction capability required to impersonate a legitimate user. Overall, the formal verification results show that the protocol achieves robust security guarantees against key interception, replay manipulation, and user spoofing attacks.

From the analysis of the above results, it can be observed that. The proposed User Nonce Binding mechanism offers stronger security guarantees than previous solutions (see Table II for details), as it establishes a strict cryptographic relationship between the subscriber identity, the session key, and the authentication process itself. In earlier 5G EAP-TLS studies, the exchanged key materials and authentication messages were not sufficiently bound to the user's original identity request, which allowed attackers to exploit weaknesses in session association and message forwarding. As a result, attacks such as Man-in-the-Middle (MITM) attacks, replay attacks, and user impersonation can still be performed under certain conditions. In the proposed scheme, the nonce generated by the User Equipment (UE), denoted as R_{UE}, is securely embedded within the encrypted SUCI message during the initial identity transmission phase. Later, this same nonce is incorporated into the encrypted key transport payload together with the pre-master key.

Figure 4. Proverif result for the modified 5G EAP-TLS protocol

Consequently, the AUSF can verify that the entity generating the session key is exactly the same entity that initiated the authentication

request. This establishes a cryptographically protected identity-key binding mechanism. Compared with previous approaches that mainly focused on detecting vulnerabilities through formal analysis, the proposed mechanism actively prevents attacks by enforcing strong mutual authentication and session consistency. Furthermore, the integration of nonce verification with certificate validation and handshake integrity checking significantly improves resistance against replay attacks, session hijacking, impersonation attacks, and MITM attacks, thereby enhancing the overall robustness of the 5G EAP-TLS protocol.

Finally, although the proposed protocol demonstrates strong security properties under the symbolic verification model, future technological advances may introduce new threats beyond the assumptions considered in this study. In particular, the emergence of large-scale quantum computers could significantly weaken current public-key cryptographic algorithms, including RSA and Elliptic Curve Cryptography (ECC), which are widely employed in TLS-based authentication mechanisms. As a result, attackers may eventually gain the capability to compromise key exchange procedures, decrypt previously recorded communications, or forge digital signatures. Therefore, future research should investigate the integration of post-quantum framework, as well as evaluate the compatibility and performance impact of such mechanisms in practical 5G environments.

TABLE II. COMPARISON OF THE FORMAL MODELS OF 5G EAP-TLS AUTHENTICATION PROTOCOLS

Work	**Modeling Tool**	**Main Protocol Characteristics**	**Security Properties Verified**	**Identified Weaknesses**	**Resistance Against Attacks**	**Limitation of Prior Solution**
[7]	Proverif	Simplified EAP-TLS model using public-key certificates	Secrecy and authentication queries	Authentication flaws and impersonation possibilities	Weak against replay/session-mixup attacks	No explicit binding between UE identity, nonce, and session transcript
[8]	Proverif	Includes TLS 1.3 handshake and session resumption	Authentication and privacy properties	Potential privacy leakage and replay-related weaknesses	Limited resistance to replay and resumption abuse	Session resumption still lacks strong UE-origin binding
[9]	Tamarin	Most detailed model with certificate distribution and four entities	Confidentiality and authentication	Concurrent-session race attacks and missing key confirmation	Stronger than prior works but still vulnerable to concurrent session misuse	Key confirmation alone cannot fully prevent identity/session misbinding
This Work	Proverif	Adds User nonce binding into authentication exchange and session derivation	Mutual authentication, secrecy, injective agreement, replay resistance, session integrity	Prevents nonce reuse and session confusion	Strong resistance against replay, MITM, impersonation, session hijacking, and concurrent-session attacks	-----

IV. Conclusion

This paper presented a detailed design of the modified 5G EAP-TLS protocol and conducted a formal security analysis using the ProVerif verification tool. The obtained verification results demonstrate that the proposed protocol is resistant to several critical attacks, including user impersonation, replay, and interleaving attacks. Furthermore, the formal analysis confirms that the proposed modifications enhance the robustness of authentication and session integrity within the 5G EAP-TLS framework. The findings of this study provide researchers with a deeper understanding of security challenges in 5G authentication protocols while also illustrating the applicability of formal verification techniques for evaluating cryptographic protocol security using ProVerif. In addition, the proposed approach establishes a foundation for strengthening secure network infrastructure development and contributes to the advancement of secure digital transformation and next-generation communication systems.

References

[1] 3GPP TS 33.501 version 18.10.0 Release 18, "TS 133 501 - V18.10.0 - 5G; Security architecture and procedures for 5G System".

[2] J. P. Mattsson and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3", *Internet Engineering Task Force, Request for Comments RFC 9190*, 2022. DOI: 10.17487/RFC9190.

[3] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", I*nternet Engineering Task Force, Request for Comments RFC 8446*, 2018. DOI: 10.17487/RFC8446.

[4] G. A. Tuan, N. H. Ha, D. N. Quang, L. H. Ton, and T. T. Hieu, "Weak Links in Smart Surveillance: An Empirical Security Evaluation of Evil Twin Attacks on IoT Cameras", *Journal of Science and Technology on Information security*, no. 3, vol. 26, pp. 62–69, 2025, DOI: 10.54654/isj.v3i26.1163.

[5] D. Simon, R. Hurst, and B. D. Aboba, "The EAP-TLS Authentication Protocol", *Internet Engineering Task Force, Request for Comments RFC 5216*, 2008. DOI: 10.17487/RFC5216.

[6] D. Basin, J. Dreier, L. Hirschi, S. Radomirović, R. Sasse, and V. Stettler, "A Formal Analysis of 5G Authentication", *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1383–1396, 2018. DOI: 10.1145/3243734.3243846.

[7] J. Zhang, L. Yang, W. Cao, and Q. Wang, "Formal Analysis of 5G EAP-TLS Authentication Protocol Using Proverif", *IEEE Access*, vol. 8, pp. 23674–23688, 2020, DOI: 10.1109/ACCESS.2020.2969474.

[8] N. Zhu, J. Xu, and B. Cui, "Formal Analysis of 5G EAP-TLS 1.3", *Advances in Internet, Data & Web Technologies*, vol. 193, L. Barolli, Ed., in Lecture Notes on Data Engineering and Communications Technologies, vol. 193. , Cham: Springer Nature Switzerland, pp. 140–151, 2024. DOI: 10.1007/978-3-031-53555-0_14.

[9] M. Shi, J. Chen, Z. Ma, K. He, M. Jia, and R. Du, "A Formal Analysis of 5G EAP-TLS Protocol", *IEEE Trans. Netw.*, vol. 33, no. 5, pp. 2179–2191, 2025, DOI: 10.1109/TON.2025.3556374.

[10] B. Blanchet, "Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif", *Foundations and Trends® in Privacy and Security*, vol. 1, no. 1–2, pp. 1–135, Oct. 2016, DOI: 10.1561/3300000004.

[11] B. Blanchet, "The Security Protocol Verifier ProVerif and its Horn Clause Resolution Algorithm", *Electron. Proc. Theor. Comput. Sci.*, vol. 373, pp. 14–22, 2022, DOI: 10.4204/EPTCS.373.2.

[12] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif 2.05: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial", 2023.

[13] C. J. F. Cremers, "Unbounded verification, falsification, and characterization of security protocols by pattern refinement", *Proceedings of the 15th ACM conference on Computer and communications security*, pp. 119–128, 2008. DOI: 10.1145/1455770.1455787.

[14] L. V. Thinh, "Paradigms in Security Protocol Verification: A Multi-Tool Analysis', *Journal of Science and Technology on Information security*, no. 3, vol 23, pp. 62–81, 2024, DOI: 10.54654/isj.v3i23.1059.

[15] D. Dolev and A. Yao, 'On the security of public key protocols", *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983, DOI: 10.1109/TIT.1983.1056650.

ABOUT THE AUTHOR

Tran Thi Nga

Workplace: Academy of Cryptography Techniques, Vietnam.

Email: tranthinga@actvn.edu.vn

Education: Tran Thi Nga received her Engineer's degree (2011) in Cryptographic Engineering and her Master's degree (2017) in Cryptographic Engineering from the Academy of Cryptography Techniques.

Recent research direction: Her research interests include cryptographic engineering, protocols and new network technology for information security.

Tên tác giả: **Trần Thị Ngà**

Cơ quan công tác: Học viện Kỹ thuật mật mã

Email: tranthinga@actvn.edu.vn

Quá trình đào tạo: Nhận bằng kỹ sư chuyên ngành kỹ thuật mật mã năm 2011 và bằng thạc sĩ chuyên ngành kỹ thuật mật mã năm 2017 tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu hiện nay: Kỹ thuật mật mã, giao thức và công nghệ mạng mới cho an toàn thông tin.

Nguyen Quoc Hung

Workplace: Viettel Cyber Security, Vietnam

Email: quochung19092003@gmail.com

Education: Engineer degree (2026) in Information Security from the Academy of Cryptography Techniques, Hanoi, Viet Nam.

Recent research direction: His research interests include cryptography in information security, network security monitoring.

Tên tác giả: **Nguyễn Quốc Hưng**

Cơ quan công tác: Công ty TNHH MTV An ninh mạng Viettel, Việt Nam

Email: quochung19092003@gmail.com

Quá trình đào tạo: Nhận bằng Kỹ sư chính quy (2026) hệ 4.5 năm chuyên ngành An toàn thông tin tại Học viện Kỹ thuật mật mã, Hà Nội, Việt Nam.

Hướng nghiên cứu hiện nay: Mật mã trong an toàn thông tin, giám sát an ninh mạng.

Bui Thu Giang

Workplace: Academy of Cryptography Techniques, Vietnam.

Email: thugiang010991@gmail.com

Education: Bui Thu Giang received her Engineer's degree (2014) in Cryptographic Engineering and her Master's degree (2020) in Cryptographic Engineering from the Academy of Cryptography Techniques

Recent research direction: Cryptography and information security

Tên tác giả: **Bùi Thu Giang**

Cơ quan công tác: Học viện Kỹ thuật mật mã, Việt Nam

Email: thugiang010991@gmail.com

Quá trình đào tạo: Nhận bằng kỹ sư chuyên ngành kỹ thuật mật mã năm 2014 và bằng thạc sĩ chuyên ngành kỹ thuật mật mã năm 2020 tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu: Mật mã và an toàn thông tin.