An Improved and Provably Secure Symmetric-Key Based 5G-AKA Protocol

Article *in* Computer Networks · September 2022

DOI: 10.1016/j.comnet.2022.109400

Awaneesh kumar Yadav
Indian Institute of Technology Roorkee
11 PUBLICATIONS **9** CITATIONS

An Braeken
Vrije Universiteit Brussel
219 PUBLICATIONS **3,389** CITATIONS

Madhusanka Liyanage
University College Dublin
260 PUBLICATIONS **6,194** CITATIONS

An Improved and Provably Secure Symmetric-Key Based 5G-AKA Protocol

Awaneesh Kumar Yadav[a], Manoj Misra[a], Pradumn Kumar Pandey[a], An Braeken[b] and Madhusanka Liyange[c]

[a] *Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India,*
[b] *Department of engineering, technology (INDI), Vrije Universiteit Brussel, Belgium,*
[c] *School of Computer Science, University Collage Dublin, Ireland and Centre for Wireless Communications, University of Oulu, Finland,*

ARTICLE INFO

Keywords:
5G-AKA
Authentication
GNY logic
Network Security
Protocol
ROR logic
Scyther tool

ABSTRACT

One of the primary authentication mechanisms defined for the 5G system is the 5G-Authentication and Key Agreement (5G-AKA) protocol. It is set to be used in the next generation of mobile communications but has several serious flaws such as privacy issues, vulnerability to traceability attacks, and has de-synchronization problem. To deal with these issues, An Braeken presented a lightweight authentication mechanism that provides security features not present in 5G-AKA, but the scheme fails to provide perfect forward secrecy. Later Munilla et al. introduced an improved version of the Braeken authentication scheme that claims to provide perfect forward secrecy but is computationally expensive and prone to DoS attacks if the size of the server database is large. Taking this in view, we propose a cost-effective scheme that provides all the security features, including perfect forward secrecy. We do the informal (non-mathematical) and formal analysis (using the ROR, GNY, and Scyther tool) of the security properties of the proposed protocol and show that the proposed protocol provides all the security features. Furthermore, we measure the performance of the proposed protocol in terms of energy consumption and computational, communication and storage costs. The evaluation results show that the proposed protocol takes significantly less cost than most of its competitors. In addition to this, we also compute the performance of the proposed protocol under unknown attacks in terms of computational, communication, and energy consumption costs. The outcome of analysis shows that the proposed protocol takes very less overhead under unknown attacks compared to its competitors.

1. Introduction

Due to recent advancements in wireless and mobile technology, mobile services have exploded significantly. More than 5.2 billion individuals, or 67 % of the global population, had subscribed to mobile services by the end of 2019. Although 4th Generation (4G) mobile technology is now prevalent, 5th Generation (5G) technology is rapidly evolving and is predicted to account for more than 20% of worldwide connections by 2025 [1]. The new mMTC (massive Machine Type Communications) service, which improves on the existing NB-IoT (Narrow Band-Internet of Things) and LTE-M (Long Term Progression Cat-M1) services released in 2015, will make IoT networks an integral component of the 5G evolution. With the number of worldwide IoT connections estimated to nearly double between 2019 and 2025, mMTC is designed to handle connection densities of up to one million devices/ and ultra-low-cost devices, with ultra-low-cost operation and maintenance (battery life of 10–15 years)[2]. However, Mobile technology security has been revealed as a critical issue that may derail or at least postpone large-scale implementation due to privacy issues discovered in previous mobile network generations. The 3GPP consortium (3rd Generation Partnership Project), which created the 3G and 4G standards and is now working on 5G, has already established a security architecture for 5G systems [3] that is 5G-AKA(Authentication and Key Agreement). However, 5G-AKA fails to provide the session-unlinkability, perfect forward secrecy, protection from malicious SN, and de-synchronization attack [4, 5, 6]. Several symmetric or asymmetric encryption-based solutions have been proposed to address these issues. However, they all contain security flaws that make them unsuitable for practical deployment. As a result, there is an urgent need to provide an authentication mechanism that meets all security requirements, as current protocols fail to meet.

1.1. Motivation & Contributions

Authentication and key agreement (AKA) protocols such as 5G-AKA, 5G-EAP-TLS, and EAP-AKA' which mutually authenticate subscribers, and operator networks are principally used to offer security and privacy for 5G communication. These protocols have been modified and standardized to safeguard subscribers' identities using randomized public-key encryption. Unfortunately, despite these improvements, these protocols are vulnerable to a variety of privacy attacks [6, 7], including a replay attack that violates the unlinkability [8] initially devised for previous mobile telephony networks. As a result, several modifications of the 5G-AKA protocol have been suggested in the literature. Some of these variants [6, 7, 8, 9, 10, 11] use public-key encryption and are computationally expensive for ultra-low-cost IoT devices.

✉ akumaryadav@cs.iitr.ac.in (A.K. Yadav); manoj.misra@cs.iitr.ac.in (M. Misra); pradumn.pandey@cs.iitr.ac.in (P.K. Pandey); an.braeken@vub.be (A. Braeken); madhusanka@ucd.ie (M. Liyange)

ORCID(s):

Others [4, 5, 12, 13, 14] use symmetric encryption for authentication to reduce the computational cost. Though these protocols are lightweight and ideal for ultra-low-cost IoT devices but have some serious vulnerability issues such as violation of perfect forward secrecy and session-unlinkability. To address these issues, An Braeken [15] introduced a lightweight authentication protocol that includes security characteristics not available in 5G-AKA. Later, Munilla et al. [16] offered an upgraded version of [15], since [15] did not offer perfect forward secrecy. While his proposed authentication technique assures perfect forward secrecy, it does have some drawbacks, such as being prone to DoS attacks if the database is large and high computational cost. Our study reveals that all existing protocols are inappropriate for realistic deployment in 5G communication, driving us to develop a secure and cost-effective protocol against the above-stated attacks.

Our contributions are as follows:

1. We propose an improved version of the protocol given in [15], which provides all the security features, especially perfect forward secrecy, session-unlinkability, session temporary key material leakage protection, non-repudiation, and is very cost-effective.
2. We do the informal (non-mathematical) and formal analysis (mathematical) of the security properties of the proposed protocol using the ROR logic, GNY logic, and Scyther tool. The comparison of the security properties of the proposed protocol with the existing protocols show that the proposed protocol meets all those security requirements which are missing in the other schemes.
3. The test-bed experiments on various cryptographic primitives have been performed under two scenarios that are server and Raspberry PI settings using the broadly accepted "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) [17]."
4. We compute the costs of the proposed protocol in terms of energy consumption, computational, communication, and storage costs. The cost comparison of the proposed protocol with the existing schemes shows that the proposed protocol is the least costly.
5. We evaluate the performance of the proposed protocol and its counterparts under unknown attacks, revealing that the proposed protocol takes significantly less overhead when unknown attacks happen as compared to most of its counterparts.

1.2. Outlines

In Section 2, we summarise the existing literature of 5G authentication, including the research gaps. Preliminaries and background used in the paper are provided in Section 3. Section 4 presents the security analysis of Braeaken's protocol [15] and Section 5 describes the proposed protocol. Further, informal and formal security analysis of the proposed protocols are discussed in Section 6 and in Section 7, respectively. The performance of the proposed protocol along with existing protocols are demonstrated in Section 8 followed by the conclusion in Section 9.

2. Related Work

This section examines the current state-of-the-art solutions for 5G authentication, which are classified into two types.

2.1. Symmetric encryption based authentication protocols

In the symmetric encryption-based authentication protocols, the secret key is used to encrypt and decrypt the exchanged message. The authentication protocols [4, 5, 12, 13] are lightweight, but they fail to offer perfect forward secrecy, insider attack protection, and session-unlinkability. Cao et al. [14] presented a Chebyshev chaotic maps-based authentication protocol. Although it ensures forward secrecy, it is computationally expensive since it requires a trusted Key Generation Centre (KGC) to generate the secret keys for the UE's and SN's. To address these issUE's, a lightweight authentication scheme [15] was proposed that provides session-unlinkability. An upgraded version of the [15] was introduced by Munilla et al. [16] that looked at the security aspects of [15] and came to the conclusion that it does not give perfect forward secrecy. The authors proposed a protocol that provides perfect forward secrecy but involves much computation on the server end.

2.2. Asymmetric encryption based authentication protocols

In asymmetric encryption-based authentication, different keys are used to encrypt and decrypt. 3GPP has suggested 5G-AKA [1] as an authentication protocol for 5G communication. The security features of the 5G-AKA were researched by [9, 18], and it was discovered that the 5G-AKA had significant vulnerabilities such as location privacy, desynchronization, and unlinkability attack. Aside from that, it fails to keep perfect forward secrecy. Koutsos [9] looked into the security features of the 5G-AKA and discovered that it has several weaknesses. A modified version of the 5G-AKA has been presented in [6], which addresses all of the 5G-shortcomings and offers additional features such as perfect forward secrecy and post-compromise security by slightly changing the 5G-AKA. Madea [11] presented a blockchain-based authentication solution that employs both symmetric and asymmetric encryption techniques. Although it offers perfect forward secrecy, it falls short of traceability. Li et al. [10] proposed an authentication scheme that combines ECC, symmetric, and asymmetric encryption and provides session key confirmation and forward secrecy. Yuchen [7] proposed an authentication scheme based on symmetric and asymmetric encryption, which allows for perfect forward secrecy but not untraceability.

2.3. Shortcomings in the existing authentication protocols

We identified following issues in the existing protocols after doing the literature review.

- Perfect forward secrecy: The majority of symmetric encryption-based authentication protocols [3, 4, 5, 12, 13, 14, 15] do not guarantee perfect forward secrecy.
- Session-unlinkability: The bulk of authentication protocols [3, 4, 5, 12] do not provide the session-unlinkability feature.
- Privacy Attack: The authentication protocols [3, 5] are subject to privacy attack.
- De-synchronization attack: The authentication protocols [3, 4, 13] face the de-synchronization problem.
- Stolen device attack: The authentication protocols [4, 12] are prone to stolen device attack.

3. Preliminaries and Background

This section introduces the preliminaries and provides background information.

3.1. Network model

The three entities that make up the 5G network model are as follows:

- User Equipment (*UE*): It is the user's physical equipment, which is usually a smartphone or an IoT device. *UE* has a cryptographic chip called Universal subscriber Identity module (USIM) that saves subscriber data and performs the security tasks required for the 5G AKA protocol to work.
- Serving Network (SN): It is the antenna or base station to which *UE* is communicating, such as when roaming. In 5G, the Security Anchor Function (SEAF) of the SN serves as an intermediary between the *UE* and its *HN* in the authentication process.
- Home Network (*HN*): A subscriber's Home Network (*HN*) is responsible for user authentication and belongs to the subscriber's service provider. It has a database with information about each of its subscribers' authentication.

The subscriber communicates with the server's base station (SN) via an insecure wireless channel through his phone (*UE*), whereas the SN communicates with the *HN* on the right via a secure (wired) channel [15, 16]. We make the following assumptions about the network same as [15, 16, 19].

- We assume *HN* and SN as one single entity for clarity and without security implications, similar to [9, 15, 16] by integrating the SN with the *HN*. This is because the SN just relays data from the *UE* and *HN*. The *HN* is in possession of the *UE*'s secret key information and is capable of computing all of the required data for the authentication procedure, which the SN subsequently sends back to the *UE*. A secure and authorized channel is considered between SN and *HN*, as indicated in the standard [3] -[TS 33.501, Section 5.9.3]. As a result, focusing on communication between *UE* and *HN* is all that is required.
- We assume that the attacker does not have access to the secrets kept in the tamper-proof hardware at *UE* and *HN′s* database [15, 16, 19]. This is due to the fact that it can be believed that physical access to the servers that contain such secrets is challenging, and tamper-proof security is very high [15].

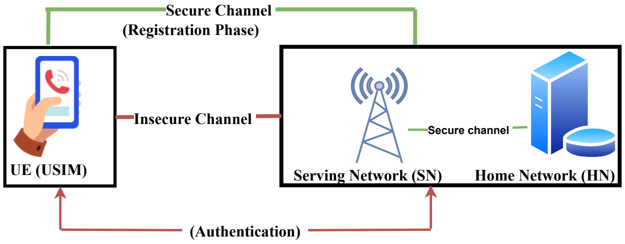


Figure 1: Proposed system model for 5G

3.2. Threat model

The proposed protocol consider the well-known Dolev-Yao (DY) [20] and CK-adversary [21] threat models to examine the robustness of the proposed protocol. According to these threat models, the adversary's (A) capabilities are as follows.

- Messages exchanged across open wireless channels are completely under A's control, and they can be read, deleted, or changed.
- Because trying to guess two values at once, such identity and password, is a "computationally infeasible task." In a polynomial amount of time, A can only predict one value.
- A can capture the exchanged messages from many sessions and perform tracability attack.
- A is capable of launching a man-in-the-middle attack by acting as a middleman. When two parties think they are speaking directly to one another, A can repeat modified versions of their communications.
- A can obtain the private key of both the communicating entities (i.e., *UE* and *HN*) at the same time.

3.3. Design Goals of AKA protocol for 5G

The expected function and security aims of AKA protocols [15, 16, 18] are outlined below.

- Session-unlinkability or Unlinkability: If an attacker intercepts the previously exchanged messages of two successful authentication sessions, he or she will not be able to relate them with each other or to the user's location.

- Privacy attack protection: The user's identity cannot be deduced from the transmitted messages.

- Resistance against replay attack: Use a nonce or timestamp in the transmitted message to provide replay attack prevention.

- Perfect forward secrecy: It ensures that even if the attacker get the long-term credentials, he or she is unable to retrieve the previous session keys.

- Resistance against session temporary key material leakage: Even if the attacker obtains all computed and intermediate data, he will be unable to infer the session key.

- Impersonation attack protection: Eavesdropping or capturing the exchanged message will not allow an attacker to impersonate *UE* and *HN*.

- Resistance against stolen device attack: Even if the attacker obtains secret data stored on the device through physical access, he will be unable to derive the previous session key.

- Resistance to de-synchronisation attack: If an authentication process is interrupted before it is completed, any secrets or sequence number that were modified during the aborted process will not affect the next round of authentication.

- Non-repudiation: It assures that one party is able to prove origin of the message to third party, without leaking crucial key data of the second party.

4. Analysis of Braeken scheme [15]

An Braeken designed a lightweight authentication mechanism to overcome the security flaws of 5G-AKA. The authentication mechanism uses the hash function and x-or operation to secure the authentication. There are two phases in the authentication scheme namely: 1) registration phase and 2) authentication phase.

4.1. Registration phase

In the registration phase, the *HN* chooses a random number K_n. Using K_n, master key K_m and common shared secret key K for each *UE* with identity id, it computes $a_n = id \oplus h(K_m, K_n)$, $b_n = a_n \oplus K_m \oplus K_n$ and $c = h(K_m, id)$. After computing (a_n, b_n, c), the *HN* securely shares the parameters $< id, a_n, b_n, c, n, K >$ with *UE* and stores $< K_m, (K, n, id) >$ into its database.

4.2. Authentication phase

- If *HN* and *UE* are synchronized then *UE* computes $h_n = h(K, id, c, a_n, b_n, n)$, increments its sequence number and sends $< a_n, b_n, h_n >$, to the *HN*. If *HN* and *UE* are de-synchronized then *UE* selects a random number r_n in order to compute $y_n = a_n \oplus id \oplus r_n$, $Z_n = n \oplus h(K, r_n, y_n)$, $h_n = h(K, id, c, a_n, b_n, n, Z_n)$, increments n by one and sends $< a_n, b_n, h_n, y_n, Z_n >$ to the *HN*.

- When *HN* receives the message $< a_n, b_n, h_n \backslash a_n, b_n, y_n, Z_n, h_n >$, if *UE* and *HN* are synchronized then *HN* derives the random number $K_n = a_n \oplus \; b_n \oplus K_m$ in order to compute $id = a_n \oplus h(K_n, K_m)$, $C = h(K_m, id)$. After getting the id, *HN* extracts the stored credentials from the database and computes the response $h_n^* = h(K, id, c, a_n, b_n, n)$ for $n\epsilon\{n......n + \triangle\}$ with $\triangle$ a predefined fixed sequence numbers and verifies $(h_n == h_n^*)$. If it matches, then it believes that *UE* is authentic, otherwise it discards the process. if *UE* and *HN* are de-synchronized then *HN* derives the random number $K_n = a_n \oplus \; b_n \oplus K_m$ in order to compute $id = a_n \oplus h(K_n, K_m)$, $C = h(K_m, id)$, $r_n = (a_n \oplus id \oplus y_n)$, $n = Z_n \oplus h(K, r_n, y_n)$ and then $h_n^* = h(K, id, c, a_n, b_n, n, Z_n)$. Afterwords, it compares $(h_n == h_n^*)$, If $(h_n == h_n^*)$ and n is larger then n^* sequence number stored in *HN*'s database. *HN* believes that *UE* is authentic and selects two new random numbers f_{n+1}, K_{n+1} in order to compute $a_{n+1} = (id \oplus h(k_m, K_{n+1}))$, $b_{n+1} = (a_{n+1} \oplus K_m \oplus K_{n+1})$, $\eta = (h(f_{n+1}, c) \oplus a_{n+1})$, $\mu = (h(f_{n+1}, c) \oplus b_{n+1})$, $\alpha = (c \oplus f_{n+1})$ and common key $K_{SEAF} = (h(K, f_{n+1}, \eta, \mu, n + 1))$, $\beta = h(K_{SEAF}, a_{n+1}, b_{n+1}, id, c)$. After computing the authentication response $(\beta, \eta, \mu, \alpha)$, *HN* forwards this to *UE*.

- When *UE* receives $(\beta, \eta, \mu, \alpha)$, it extracts $f_{n+1} = (c \oplus \alpha)$ in order to compute $a_{n+1} = (\eta \oplus h(f_{n+1}, c))$, $b_{n+1} = (\mu \oplus h(f_{n+1}, c))$, $K_{SEAF} = (h(K, f_{n+1}, \eta, \mu, n + 1))$. After computing theses credentials, *UE* verifies the response $\beta^* = h(K_{SEAF}, a_{n+1}, b_{n+1}, id, c)$ with the received β. If it matches, then *UE* believes the *HN* is authentic and starts communication.

4.3. Cryptananlysis on Braeken protocol

Monilla et al. [16] presented a security analysis of [15] which shows that it does not satisfy the perfect forward secrecy. In fact, the scheme of [15] never claimed to satisfy the perfect forward secrecy feature. Monilla et al. [16] proposed the use of hash chains to include these features.

4.3.1. *Perfect Forward Secrecy*

Braeken's scheme violates the perfect forward secrecy also shown by Munila et al. [16] because it uses the static key K, similar to the 5G-AKA standard in order to limit the amount of changes. Therefore, if an attacker obtains the long term credentials such as (K, c, id, n) then he can derive all the previous session keys. The steps below show how the Braeken protocol breaks perfect forward secrecy.

- Attacker captures the exchanged messages $< a_n, b_n, h_n \backslash a_n, b_n, y_n, Z_n, h_n \ >$, of previous authentication session.

- Obtains access to confidential data (id, K, c, n) stored on *UE*'s USIM card.

- Attacker has (a_n, b_n, c, K, n). So, it gets the sequence number of session i as follows: for synchronization case: compute $h = (K, id, n, c, a_n, b_n)$ for $j = (n-1 : -1 : 1)$ until $(h^* == h_i)$: then set $n_i \leftarrow j$ and for the de-synchronization $n_i = Z_n \oplus h(K, r_n, y_n)$.

- Now attacker has n_i for sesion i, then he or she can obtain the session key $K_{SEAF_i} = h(K, f_{n+1}, \mu, \eta, a_{n+1})$ for i^{th} session.

5. Proposed Protocol

In this section, we present a improved version of Braeken [15] authentication scheme. We update the long term key K and certain hash functions in order to be able to offer resistance against session temporary key material leakage and non-repudiation in addition to perfect forward secrecy. At the same time, the proposed scheme is also more cost-effective as compared to protocols proposed in [15] and [16]. There are two phases in the proposed protocol outlined below.

- Registration phase: In this registration phase, *UE* obtains the USIM that stores secrets via secure channel.

- Authentication phase: In this phase, *UE* and *HN* authenticate each other and securely generate the session key for data confidentiality and integrity.

5.1. Registration phase

In the registration phase, *HN* computes the following parameters for *UE* with the identity UE_{id}. It first selects a random number R_1, key K and flag $(f) = 0$ in order to compute $A = (UE_{id} \oplus H(K_m, R_1))$, $B = (A \oplus R_1 \oplus K_m)$, $K_1 = H(K[f], R_1)$. Afterwards, *HN* securely shares these credentials $\langle A, B, K_1, f, n, UE_{id} \rangle$ with the *UE* and saves the $\langle K_m, (UE_{id}, n, K[0] = K_1, K[1] = K) \rangle$ into his database.

UE_{id} represents *SUPI* of the *UE*, n is the sequence number (initially $n = 0$), and K_m is the secret key of *HN*, which is the same for all users, but it is not shared with them. K_1 and K are the short-term keys shared with *UE* and *HN* respectively. K_m and *UE*'s data in the server database are stored in different places as in [6, 15, 16].

5.2. Authentication phase

In the authentication phase, *UE* and *HN* prove their authenticity and securely generate a session key for data transfer using the pre-shared secrets.

- *UE* selects the random number R_2 in order to compute $I = (H(K_1) \oplus R_2)$, $J = (n \oplus H(K_1, R_2))$, $F_1 = H(UE_{id} \parallel H(K_1) \parallel f \parallel n \parallel R_2)$. Afterwords it forwards $\langle A, B, F_1, f, I, J \rangle$ to *HN* and increments n by one.

- Upon receiving the message $\langle A, B, F_1, f, I, J \rangle$, *HN* first extract $UE_{id} = A \oplus H(K_m, A \oplus B \oplus K_m)$ in order to extract the secrets $(n, K[f])$ stored in database. Thereafter, it computes $R_2 = I \oplus H(H(K[f], A \oplus B \oplus K_m))$, and from R_2, it extracts $n^* = J \oplus (H(H(K[f], A \oplus B \oplus K_m)), R_2)$ then checks that $n \succ n^*$ and $n \varepsilon \{n.\ldots.n + \Delta\}$ with Δ a predefined fixed threshold value. Denote by n the value which satisfies the equality. If it does not meet value of n inside range Δ, *HN* aborts the process. Now it computes the ${F_1}^* = H(UE_{id} \parallel H(H(K[f], A \oplus B \oplus K_m)) \parallel f \parallel n \parallel R_2)$ and compares $\{F_1 == {F_1}^*\}$. If it matches then *HN* believes that *UE* is authentic, increments n by one and selects the random number R_3 and new key K_{new} in order to compute $A_{new} = (UE_{id} \oplus H(K_m, R_3))$, $B_{new} = (A_{new} \oplus R_3 \oplus K_m)$, $K_{SEAF} = H(R_2 \parallel H(K[f], A \oplus B \oplus K_m) \parallel n + 1)$, ${K_1}^{new} = H(K_{new}, R_3)$, $D_1 = ({K_1}^{new} \oplus H(K[f], A \oplus B \oplus K_m) \oplus R_2)$, $D_2 = A_{new} \oplus H({K_1}^{new}, R_2)$, $D_3 = B_{new} \oplus H(R_2, {K_1}^{new})$ and $F_2 = H(K_{SEAF} \parallel A_{new} \parallel B_{new})$. It sets $K[(f + 1) mod 2] = K_{new}$ and sends the $\langle D_1, D_2, D_3, F_2 \rangle$ to the *UE*.

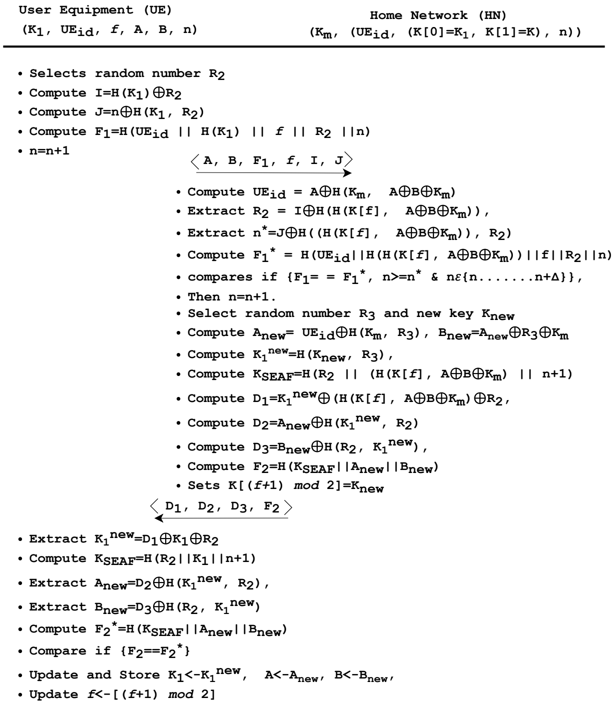
- When *UE* receives the $\langle D_1, D_2, D_3, F_2 \rangle$ then it extracts the ${K_1}^{new} = D_1 \oplus K_1 \oplus R_2$ in order to compute $A_{new} = D_2 \oplus H({K_1}^{new}, R_2)$, $B_{new} = D_3 \oplus H(R_2, {K_1}^{new})$, $K_{SEAF} = H(R_2 \parallel K_1 \parallel n+1)$, ${F_2}^* = H(K_{SEAF} \parallel A_{new} \parallel B_{new})$, compare $\{F_2 == {F_2}^*\}$. If it matches then *UE* believes that *HN* is authentic and saves the K_{SEAF} and replaces the old secrets with the new, $\langle K_1 \leftarrow {K_1}^{new}, A \leftarrow A_{new}, B \leftarrow B_{new}, f \leftarrow (f + 1) mod 2 \rangle$.

6. Informal Analysis of the Proposed Protocol

In this section, we do an informal assessment (non-mathematical) of the proposed protocol to confirm that it satisfies the security requirements stated in Section 3.3.

Proposition 1. The proposed protocol provides mutual authentication.

Proof. When *HN* receives $\langle A, B, F_1, f, I, J \rangle$, it extracts R_2 and n to compute the F_1^*. After that, *HN* compares $\{F_1^* == F_1\}$, if they match, *HN* believes that *UE* is authentic because only *UE* knows secret K_1 included in F_1^*. On the other side, when *UE* receives the authentication response $\langle D_1, D_2, D_3, F_2 \rangle$ from *HN*, it extract ${K_1}^{new}$, A_{new} and B_{new} in order to compute K_{SEAF} and F_2^*. After that, *UE* compares $\{F_2^* == F_2\}$, if they match, *UE* believes that *HN* is legitimate because K_{SEAF} included in F_2^* cannot be computed without knowing the secrets K_1 and K_m; otherwise, *UE* aborts the authentication process. Hence, the proposed protocol provides mutual authentication.



User Equipment (UE)
(K_1, UE_{id}, f, A, B, n)

Home Network (HN)
(K_m, (UE_{id}, ($K[0]=K_1$, $K[1]=K$), n))

- Selects random number R_2
- Compute $I=H(K_1)\oplus R_2$
- Compute $J=n\oplus H(K_1, R_2)$
- Compute $F_1=H(UE_{id} \,||\, H(K_1) \,||\, f \,||\, R_2 \,||n)$
- n=n+1

$\langle A, B, F_1, f, I, J\rangle$

- Compute $UE_{id} = A\oplus H(K_m, A\oplus B\oplus K_m)$
- Extract $R_2 = I\oplus H(H(K[f], A\oplus B\oplus K_m))$,
- Extract $n^*=J\oplus H((H(K[f], A\oplus B\oplus K_m)), R_2)$
- Compute $F_1{}^* = H(UE_{id}||H(H(K[f], A\oplus B\oplus K_m))||f||R_2||n)$,
- compares if $\{F_1= = F_1{}^*, n>=n^*$ & $n\varepsilon\{n.......n+\Delta\}\}$,
- Then n=n+1.
- Select random number R_3 and new key K_{new}
- Compute $A_{new}= UE_{id}\oplus H(K_m, R_3)$, $B_{new}=A_{new}\oplus R_3\oplus K_m$
- Compute $K_1{}^{new}=H(K_{new}, R_3)$,
- Compute $K_{SEAF}=H(R_2 \,||\, (H(K[f], A\oplus B\oplus K_m) \,||\, n+1)$
- Compute $D_1=K_1{}^{new}\oplus(H(K[f], A\oplus B\oplus K_m)\oplus R_2$,
- Compute $D_2=A_{new}\oplus H(K_1{}^{new}, R_2)$
- Compute $D_3=B_{new}\oplus H(R_2, K_1{}^{new})$,
- Compute $F_2=H(K_{SEAF}||A_{new}||B_{new})$
- Sets $K[(f+1)\ mod\ 2]=K_{new}$

$\langle D_1, D_2, D_3, F_2\rangle$

- Extract $K_1{}^{new}=D_1\oplus K_1\oplus R_2$
- Compute $K_{SEAF}=H(R_2||K_1||n+1)$
- Extract $A_{new}=D_2\oplus H(K_1{}^{new}, R_2)$,
- Extract $B_{new}=D_3\oplus H(R_2, K_1{}^{new})$
- Compute $F_2{}^*=H(K_{SEAF}||A_{new}||B_{new})$
- Compare if $\{F_2==F_2{}^*\}$
- Update and Store $K_1<-K_1{}^{new}$, $A<-A_{new}$, $B<-B_{new}$,
- Update $f<-[(f+1)\ mod\ 2]$

Figure 2: proposed protocol

Proposition 2. The proposed protocol preserves Perfect Forward Secrecy.

Proof. Even if long term secrets UE_{id}, K_m of the proposed protocol are compromised, the attacker cannot derive the session keys $K_{SEAF} = H(R_2 \parallel K_1 \parallel n+1)$ of previous sessions because of K_1 and R_2. *UE* and *HN* update their secret key K_1 after every successful authentication session. As a result, knowing UE_{id}, K_m will provide no insight to the attacker into the secret short-term Key (K_1) and the random number (R_2).

Proposition 3. The proposed protocol provides session-unlinkability.

Proof. This attack is impossible due to the use of temporary identities A and B, which are updated after each successful authentication request. The temporary identities used in two different successful authentication sessions are entirely independent of each other. Suppose the attacker captures the previously exchanged message of the different successful authentication sessions. In that case, he cannot link the messages of one successful authentication session to another successful authentication session because identities (A, B) and authentication responses are computed using new random numbers in each session. As a result, an attacker can not relate messages of one successful authentication session to another.

Proposition 4. The proposed protocol is resilient against replay attack.

Proof. In the proposed protocol, sequence numbers are employed in every message to ensure that message can not be replayed. When *HN* receives $\langle A, B, F_1, f, I, J\rangle$ from *UE*, it extract n from J and checks that $n > n^*$ (n^* is the sequence number stored in *HN*'s database) and $n\varepsilon\{n.....n+\Delta\}$ with Δ a predefined fixed threshold value. If n extracted from J does not meet these two conditions, *HN* aborts the process. Therefore, the proposed protocol is resilient against replay attack.

Proposition 5. The proposed protocol is resilient against privacy attack.

Proof. The identity of *UE* is always transmitted in the masked form in the proposed protocol. As a result, collecting the exchanged messages $\langle D_1, D_2, D_3, F_2\rangle, \langle A, B, F_1, f, I, J\rangle$ will not provide any information regarding the identity of the *UE*. Hence, the proposed protocol is resilient against privacy attack.

Proposition 6. The proposed protocol is resilient against session temporary key material leakage.

Proof. Even if the attacker obtains the secrets computed and stored during authentication process such as $\langle R_2, R_3, UE_{id}, n, A_{new}, B_{new}, A, B\rangle$, he will be unable to deduce the session key $K_{SEAF} = H(R_2 \parallel K_1 \parallel n+1)$ because R_1 and K_1 are computed but not stored during the authentication session. Therefore, an attacker can not deduce the session key even if he can access the computed and stored secrets during the authentication session.

Proposition 7. The proposed protocol is resilient against impersonation attack.

Proof. In order to impersonate as *UE* or *HN*, the attacker must compute the legitimate message. The proposed protocol uses the random numbers R_1, R_2, R_3 and the dynamic key K_1 and K, making it difficult for an attacker to compute the forged message. Thus, the proposed protocol is resilient against impersonation attack.

Proposition 8. The proposed protocol is resilient against de-synchronization attack.

Proof.

- De-synchronization due to sequence number: The sequence number is used in all the messages in the proposed protocol. *UE* increments the sequence number before sending the first message to *HN*. When *HN* receives the message, it verifies the received sequence number, increments its sequence number, and increases the threshold value of the sequence number domain. If the attacker replays captured messages to *HN*, *HN* will find out that the messages are replayed because the received message's sequence number will be less than the *HN*'s sequence number.

- De-synchronization due to key updation: *HN* maintains a table with two entries and stores two keys in it (i.e., current key K_1 and future key K_{new}). *UE* keeps a flag (f) which decides the key used in the new session. When *UE* sends the message to *HN*, it includes the flag (f) in the message. After receiving the message from *UE*, *HN* uses the key at the table entry $[f]$, selects a new key, and stores it in the table entry $[\bar{f}]$. On the other side, when *UE* receives the message from *HN*, it toggles the flag value. We now show that whenever *HN* receives a message from *UE* with a flag (f), the correct key K is always at the entry table $[f]$. Let us assume that at time t, *UE*'s flag is f, and the correct key is stored in the *HN* at location table $[f]$. When *UE* starts the authentication protocol following possibilities may arise: (i) Message from *UE* to *HN* is lost. When *UE* runs the protocol again, *UE*'s flag is f, and the correct key is stored in the *HN* at the location table$[f]$. (ii) Message from *HN* to *UE* is lost but *HN* computes and stores the new key at table $[\bar{f}]$. When *UE* starts the protocol again, *UE*'s flag is f, and the correct key is stored in the *HN* at location table $[f]$. (iii) Both messages are delivered. When *UE* starts the process again, *UE*'s flag is $[\bar{f}]$, and the correct key is stored in the *HN* at the location tablet $[\bar{f}]$.

This shows that our proposed protocol is resilient against the de-synchronization attack.

Proposition 9. The proposed protocol is resilient against stolen device attack.

Proof. We assume that the secrets are stored in tamper-resistant hardware in the *UE* and *HN* same as [15, 16, 19]. Still, suppose the attacker physically accesses the device by stealing the device and gets the secrets (K, UE_{id}, n, f, A, B). Even in that case, he will be unable to deduce the previous session keys due to the dynamic updation of key and sequence numbers. Therefore, an attacker cannot obtain the key, random number, and session sequence number because they are updated after every successful authentication. Hence, our proposed protocol is resilient against the physical access of the device.

Proposition 10. The proposed protocol provides non-repudiation.

Proof. In the proposed protocol, *HN* can prove that the received message is sent by *UE* without leaking the secrets of *UE* to the third party because F_1 contains the hash of K_1. *HN* can keep a copy of $F_1, UE_{id}, H(K_1), n$ and R_2. If *UE* denies sending the message, then *HN* can provide these values to the third party. F_1 can be computed again from $UE_{id}, H(K_1), n$ and R_2. If both values of F_1 match, it confirms that *UE* has sent the message because only *UE* knows these values and can compute F_1. Since the third party can only see the hashed form of the key during the verification, not the original credentials of the *UE*, this analysis shows that *HN* can prove the origin of that message to a third party without leaking the crucial key data of the *UE*. Therefore, our proposed protocol provides non-repudiation.

Proposition 11. The proposed protocol is resistant against the DoS attack.

Proof. When *HN* receives a request, it extracts the sequence number and compares it with the range of sequence numbers stored in $HN's$ database. It proceeds in case of a match; otherwise, it aborts the process. To verify the freshness of the sequence number, the proposed protocol only computes five hash functions, while the number of hash functions computed in [16] are equal to the size of the database. Therefore, if the attacker resends the old request, the proposed protocol can verify the freshness by computing only five hash functions while [16] requires to compute the number of hash function equal to the size of *HN*'s database. This shows that the proposed protocol is resistant to the DoS attack.

7. Formal Security Analysis

This section demonstrate the formal verification of the proposed protocol using Real-Or-Random (ROR) logic, GNY logic, and Scyther tool to depict that proposed protocol offer all the security feature as mentioned in Section 3.3.

7.1. Formal security analysis using ROR Logic

This section uses the ROR model proposed by Abdalla et al.[22] to examine the hardness of obtaining the securely generated session key during authentication by the adversary ($\wp$). The authentication protocol involves two entities: a) User Equipment (UE), b) Home Network (*HN*). Let UE^i and HN^j represent the instances i and j of UE and *HN* respectively. The ROR model assumes that the adversary ($\wp$) can delete, edit, insert and learn conveyed messages during communication. In this model, $\wp$ can use queries listed below to simulate a real attack. E^k in the following discussion represents the instance k of entity E.

- *Execute* (UE^i, HN^j)*:* $\wp$ captures or eavesdrops on the exchanged message over the public channel between instances UE^i and HN^j.
- *Reveal* (E^k)*:* $\wp$ can use this query to get access to the current session's session key between UE and *HN*.
- *Send* (E^k, m)*:* $\wp$ can capture a message and then can either simply forward it to the other participant or can forward it after modifying it. $\wp$ can also generate a message and can forward it to the intended participant E^k. Responses received by $\wp$ will be response generated by E^k on the receipt of this message.
- *Test* (E^K)*:* This ROR-based query verifies the session key security between UE and *HN*. An unbiased coin is flipped before the game begins, and the outcome (0 or 1) is stored in a bit ($\aleph$). When $\wp$ runs this query, the decision is made depending on the coin toss outcome. Assume $\wp$ is running *Test*, and session key (SK) is new. If ($\aleph$ = 1), the participant returns a random number; if ($\aleph = 0$), the participant returns the session key. Otherwise, a null value is returned

Furthermore, $\wp$ and any other participant have access to a random oracle, Hash, which is modeled as a collision-resistant hash function.

Theorem 1: Consider an adversary ($\wp$) attempting to break the session key (Sk) in polynomial time during the authentication phase. Then $Adv_{\wp} \leq \frac{q_h^2}{2^M} + \frac{(q_s+q_e)^2}{N}$

where q_h, q_s, q_e, M and N denote the number of $Hash$ queries, $Send$ queries, $Execute$ query, length of the hash function output value and range space of random number respectively.

Proof: We present a proof that is similar to [23], [24]. We demonstrate session key security of the proposed protocol using a series of three games termed as G_i, where $i \in$ {0, 1, 2} and an event $Success_{\wp G_i}$ defined as " $\wp$ can accurately predict the random bit $\aleph$ in game G_i, and its probability to win the game G_i is specified by $Pr[Success_{\wp G_i}]$." The following three games are listed below.

Game(G_0) : In this game, $\wp$ perform the real attack on proposed protocol. Since, bit $\aleph$ is randomly selected at the staring, so, from the semantic security, we obtain

$$Adv_{\wp} = |2Pr[Success_{\wp G_0}] - 1] \tag{1}$$

Game (G_1) : In this game, $\wp$ captures the exchanged message $< A, B, F_1, f, I, J >, < D_1, D_2, D_3, F_2 >$ to perform the eavesdropping attack on proposed protocol by executing the *Execute* query. Afterword, $\wp$ executes the *Test* and *Reveal* query to find out that the return value is real or random. Since, session key $K_{SEAF} = H(R_2 \parallel K_1 \parallel n+1)$ is generated using the random numbers (R_2, R_3), keys (K_1), and sequence number $(n + 1)$. However, random numbers used in session key generation are unknown to $\wp$. So, $\wp$ will be unable to derive the SK. Therefore, we can conclude that $\wp$ will be unable to win the game even if he or she capture or eavesdrops the exchanged message. Thus, the winning probability of G_1 and G_0 will be equal.

$$Pr[Success_{\wp G_1}] = Pr[Success_{\wp G_0}] \tag{2}$$

Game(G_2) : In this game, $\wp$ executes the *Send* query in order to model it as an active game. The term used in exchanged messages $< F_1, F_2 >$ and $< A, B, I, J, D_1, D_2, D_3 >$ are protected by the hash function, random numbers, and nonce. Therefore, $\wp$ will never obtain the random numbers and nonce from the exchanged message because of the collision resistance nature of h(.). As a result, there is no collision when the *Hash* query is run. The G_2 will be the same as G_1 except for hash collision and random number collision. So, we can obtain the following result by adopting the birthday paradox

$$Pr[Success_{\wp G_1}] - Pr[Success_{\wp G_2}] \leq \frac{q_h^2}{2^{M+1}} + \frac{(q_s + q_e)^2}{2N} \tag{3}$$

As all the games have been executed, $\wp$ must conjecture the exact bit c. Hence, it follows

$$Pr[Success_{\wp G_2}] = \frac{1}{2} \tag{4}$$

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

$$\begin{aligned} Adv_{\wp} &= |2Pr[Success_{\wp G_0}] - 1| \\ \frac{1}{2}Adv_{\wp} &= |Pr[Success_{\wp G_0}] - \frac{1}{2}| \\ &= Pr[Success_{\wp G_1}] - Pr[Success_{\wp G_2}] \end{aligned} \tag{5}$$

Table 1
GNY Notations

Symbol	Description
$UE \ni M$	UE possess M.
$UE \lhd * M$	UE receives M and UE did not convey it previously in the current session.
$UE \lhd M$	UE is told formula M, UE receives M.
$UE \mid\sim M$	UE once conveyed formula M.
$UE \mid\equiv \phi M$	UE believes that M is recognizable or computable.
$UE \mid\equiv \#(M)$	UE believes that M is not used earlier.
$\{M\}_K$	M is encrypted using shared secret key K.
$UE \mid\equiv UE \overset{K}{\leftrightarrow} HN$	UE believes that the secret key (K) is shared between UE and HN.

We get the following result from the Eq (3) and (5).

$$\begin{aligned} \frac{1}{2}Adv_{\wp} &\leq \frac{q_h^2}{2^{M+1}} + \frac{(q_s + q_e)^2}{2N} \\ Adv_{\wp} &\leq \frac{q_h^2}{2^M} + \frac{(q_s + q_e)^2}{N} \end{aligned} \tag{6}$$

Hence, we can infer from the output that the adversary cannot get the session key in polynomial time.

7.2. Formal security analysis using GNY Logic

We use GNY [25] logic (i.e., extended version of widely used BAN Logic) to do the mathematical analysis of the proposed protocol, which reveals that UE and HN mutually authenticate and securely share the session key for data confidentiality and integrity [26, 27].

Let UE and HN represent two principles, and M represent a statement. The notations used in GNY logic are shown in Table 1.

7.2.1. *Logical postulates*

1. Being Told Rule (BTR_1): If UE receives M and has not yet conveyed it in this session, UE receives M.

$$\frac{UE \lhd * M}{UE \lhd M}$$

2. Being Told Rule (BTR_2): If UE receives hashed form and he has one of the two arguments (M, N), then other argument is assumed to have been told as well. Where F denotes a one-to-one function that is also computationally feasible in its inverse.

$$\frac{UE \lhd F(M, N), UE \ni M}{UE \lhd N}$$

3. Possession rule (PR_1): If UE receives M then UE can be assumed to possess M

$$\frac{UE \lhd M}{UE \ni M}$$

4. Possession rule (PR_2): If UE possess M and N then UE can be assumed to possess (M, N)

$$\frac{UE \ni M, UE \ni N,}{UE \ni (M, N)}$$

5. Possession rule (PR_3): If UE possess M then UE can be assumed to possess $H(M)$

$$\frac{UE \ni M}{UE \ni H(M)}$$

6. Freshness rule (FR): If UE believes M is new, then UE has the right to believe that any message containing M is fresh, as well as a computationally viable one-to-one function of the message contents.

$$\frac{UE \mid\equiv \#(M)}{UE \mid\equiv \#(M, N), UE \mid\equiv \#(F(M, N)),}$$

7. Message Interpretation rule (MIR_1) : if UE possess M, N, UE believes M is shared between $UE\&HN$, and UE believes $M\&N$ are fresh then UE is entitled to believe that HN has sent the M, N, and UE believes that HN has sent the $H(M, N)$.

$$\frac{UE \lhd * H(M, N), UE \ni (M, N), \quad UE \mid\equiv UE \overset{M}{\longleftrightarrow} HN, UE \mid\equiv \#(M, N)}{UE \mid\equiv HN \mid\sim (M, N), UE \mid\equiv HN \mid\sim H(M, N),}$$

8. Message Interpretation rule (MIR_2): If UE believes that HN has sent the M, N and that the M is new and fresh, then UE has the right to assume that HN holds the M, N.

$$\frac{UE \mid\equiv HN \mid\sim (M, N), UE \mid\equiv \#(M)}{UE \mid\equiv HN \ni (M, N)}$$

7.2.2. Initial assumptions for the protocol

The protocol's assumptions are as follows:

H_1 : $UE \ni K_1$
H_2 : $UE \ni UE_{id}$
H_3 : $UE \ni n$
H_4 : $UE \mid\equiv UE \overset{K_1}{\longleftrightarrow} HN$
H_5 : $UE \mid\equiv UE \overset{UE_{id}}{\longleftrightarrow} HN$
H_6 : $UE \mid\equiv \#(n)$
H_7 : $HN \ni K_1$
H_8 : $HN \ni n$
H_9 : $HN \ni UE_{id}$
H_{10} : $HN \mid\equiv UE \overset{K_1}{\longleftrightarrow} HN$
H_{11} : $HN \mid\equiv UE \overset{UE_{id}}{\longleftrightarrow} HN$
H_{12} : $HN \mid\equiv \#(n)$
H_{13} : $UE \mid\equiv \phi(R_2)$
H_{14} : $UE \ni R_2$

7.2.3. Security goals of the proposed protocol:

The protocol's security goals are as follows:

$HN \mid\equiv UE \ni (H(UE_{id}, H(K_1), f, R_2))$,
$HN \mid\equiv UE \mid\equiv UE \overset{K_{SEAF}}{\longleftrightarrow} HN$
$UE \mid\equiv HN \ni (K_{SEAF}, A_{new}, B_{new})$
$UE \mid\equiv HN \mid\equiv UE \overset{K_{SEAF}}{\longleftrightarrow} HN$

7.2.4. Idealized form of the proposed protocol:

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

M_{10}:$UE \rightarrow HN$: $HN\lhd$: $* (* (H(K_1))\oplus * R_2)$,
M_{11}:$UE \rightarrow HN$: $HN\lhd$: $* (* n\oplus * H(K_1, R_2))$,
M_{12}:$UE \rightarrow HN$, $HN\lhd$:$* (H(* UE_{id} \| H(* K_1) \| f \| * n \| * R_2))$,
M_{21}: $HN \rightarrow UE$: $UE\lhd$: $* (* (K_1{}^{new}\oplus * H(K_1) * \oplus R_2)$,
M_{22}: $HN \rightarrow UE$: $UE\lhd$:$* (A_{new}\oplus * H(K^{new}, R_2))$,
M_{23}: $HN \rightarrow UE$: $UE\lhd$: $(* B_{new}\oplus * H(R_2, K^{new}))$,
M_{24}: $HN \rightarrow UE$: $UE\lhd$: $* H(* K_{SEAF} \| * A_{new} \| * B_{new}$),

7.2.5. Proof and derivation of security goals:

1. When the BTR_1, BTR_2 and PR_1 rule is applied to M_{10} based on H_7, the result is
   S_1 : $HN \ni R_2$
2. When the BTR_1, BTR_2 and PR_1 rule is applied to M_{11} based on H_7 and S_1, the result is
   S_2 : $HN \ni n$
3. Applying the PR_2 rule based on S_1, S_2, H_7 and H_9, we get
   S_3 : $HN \ni (UE_{id} \| H(K_1) \| f \| n \| R_2)$
4. Applying the PR_3 rule we get
   S_4 : $HN \ni (H(UE_{id} \| H(K_1) \| f \| n \| R_2))$
5. Applying the FR rule on S_4 based on S_2 and H_8 we get
   S_5 : $HN \mid\equiv \#(H(UE_{id} \| H(K_1) \| f \| R_2))$
6. Applying the MIR_1 rule based on S_4, S_5, and H_{10}, H_{11} we get
   S_6 : $HN \mid\equiv UE \mid\sim (H(UE_{id} \| H(K_1) \| f \| R_2))$
7. Based on S_6 and S_5, we apply MIR_2
   S_7 : $HN \mid\equiv UE \ni (H(UE_{id} \| H(K_1) \| f \| R_2))$
8. When the BTR_1, BTR_2 and PR_1 rule is applied to M_{21} based on H_1, H_{14}, the result is
   S_8 : $UE \ni K_1{}^{new}$
9. When the BTR_1, BTR_2 and PR_1 rule is applied to M_{22} based on H_{14} and S_8, the result is
   S_9 : $UE \ni A_{new}$
10. When the BTR_1, BTR_2 and PR_1 rule is applied to M_{22} based on H_{14} and S_8, the result is
    S_{10} : $UE \ni B_{new}$
11. Based on S_9, S_{10} and H_1 and H_{14}, we apply the PR_2 rule.
    S_{11} : $UE \ni: (K_{SEAF} \| A_{new} \| B_{new})$
12. Applying the PR_3 rule on S_{11}.
    S_{12} : $UE \ni: (H(K_{SEAF} \| A_{new} \| B_{new}))$
13. We apply FR to S_{12} based on H_{12}.
    S_{13} : $UE \ni: (K_{SEAF} \| A_{new} \| B_{new})$
14. Based on H_4, H_5, S_{13}, and S_{12}, we apply the MIR_1
    S_{14} : $UE \mid\equiv HN \mid\sim (K_{SEAF} \| A_{new} \| B_{new})$,
15. MIR_2 is applied based on S_{14} and S_{13}.
    S_{15} : $UE \mid\equiv HN \ni (K_{SEAF} \| A_{new} \| B_{new})$
16. From S_7 (i.e., Goal-1) and S_{15} (i.e., Goal-3), we can conclude that UE believes that HN possesses $(K_{SEAF} \| A_{new} \| B_{new})$ and HN believes that UE

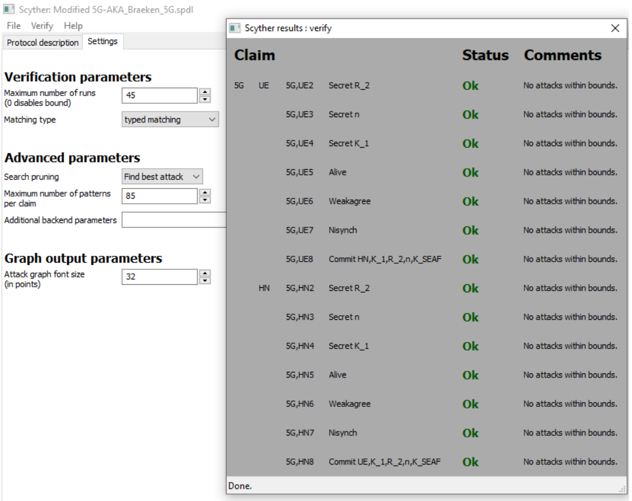
possesses $(UE_{id} \parallel R_2 \parallel H(K_1) \parallel n)$ then UE and HN both will believe that K_{SEAF} is shared between them. So, based on that we can infer

S_{14} : $HN \mid\equiv UE \mid\equiv UE \xleftrightarrow{K_{SEAF}} HN$ Goal-2.

S_{15} : $UE \mid\equiv HN \mid\equiv UE \xleftrightarrow{K_{SEAF}} HN$ Goal-4.

7.3. Formal security verification using Scyther tool

The proposed protocol's security properties are verified using the scyther tool [28] that has been shown beneficial for checking and analysing security protocols, it supports a variety of adversary models, including the traditional Dolev–Yao model, the CK model, and the eCK model [14]. As shown in Fig.3, Fig.4, and Fig.5, the validation outcome depicts that our proposed protocol ensures all security claims such as Alive (i.e., ensures that all the events are performed by the communicating parties), Weakagree (i.e., ensures that the protocol offer the protection from impersonation attacks), Nisynch (i.e., ensures that the one party sends all messages and that the other party receives them), and Secret (i.e., unknown to attacker) specified by the scyther tool [29, 30, 31, 32]. We test our proposed protocol in different parameter settings to show the proposed protocol's resiliency. There are three parameters: verification parameter, advanced parameter, and graph output parameter. The detail description of these parameters are give in [33]. In the first setup, we execute the protocol by setting the component of verification parameters such as the maximum number of runs 45 and matching type as *typed matching*, the component of the advanced parameter such as search pruning as *find best attack*, and the maximum number of patterns per claim 85 and at last graph output parameter to 32. The results are shown in Fig.3. In the second setup, we execute the protocol by setting the component of verification parameters such as the maximum number of runs 100 (i.e., maximum) and matching type as *find all flaws*, component of the advanced parameter such as search pruning as *find best attack* and the maximum number of pattern per claim 100 and at last graph output parameter to 32. The results are shown in Fig.4. In the third setup, we execute the protocol by setting the component of verification parameters such as the maximum number of runs 100 (i.e., maximum) and matching type as *find all type flaws*, component of the advanced parameter such as search pruning as *find all attacks* and the maximum number of pattern per claim 85 and at last graph output parameter to 32. The results are shown on Fig.5. The execution results clearly show that all the claims are verified and no attack is found. As a result, we may conclude that the Scyther tool found no attacks on the proposed protocol.



Verification parameters

Maximum number of runs (0 disables bound): 45

Matching type: typed matching

Advanced parameters

Search pruning: Find best attack

Maximum number of patterns per claim: 85

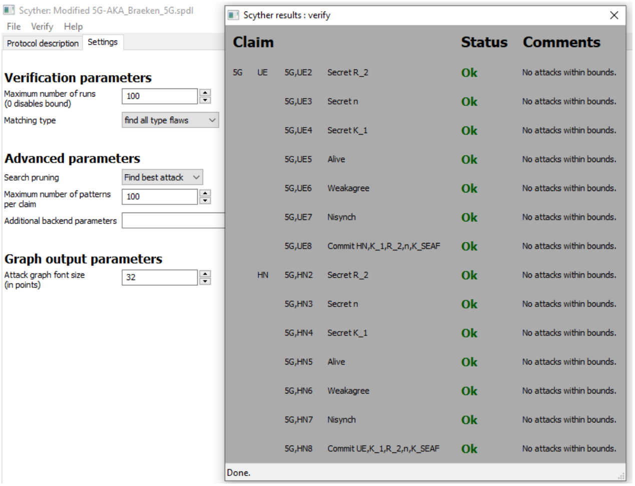
Additional backend parameters:

Graph output parameters

Attack graph font size (in points): 32

Claim				Status	Comments
5G	UE	5G,UE2	Secret R_2	Ok	No attacks within bounds.
		5G,UE3	Secret n	Ok	No attacks within bounds.
		5G,UE4	Secret K_1	Ok	No attacks within bounds.
		5G,UE5	Alive	Ok	No attacks within bounds.
		5G,UE6	Weakagree	Ok	No attacks within bounds.
		5G,UE7	Nisynch	Ok	No attacks within bounds.
		5G,UE8	Commit HN,K_1,R_2,n,K_SEAF	Ok	No attacks within bounds.
	HN	5G,HN2	Secret R_2	Ok	No attacks within bounds.
		5G,HN3	Secret n	Ok	No attacks within bounds.
		5G,HN4	Secret K_1	Ok	No attacks within bounds.
		5G,HN5	Alive	Ok	No attacks within bounds.
		5G,HN6	Weakagree	Ok	No attacks within bounds.
		5G,HN7	Nisynch	Ok	No attacks within bounds.
		5G,HN8	Commit UE,K_1,R_2,n,K_SEAF	Ok	No attacks within bounds.

Figure 3: Scyther tool result for proposed protocol



Verification parameters

Maximum number of runs (0 disables bound): 100

Matching type: find all type flaws

Advanced parameters

Search pruning: Find best attack

Maximum number of patterns per claim: 100

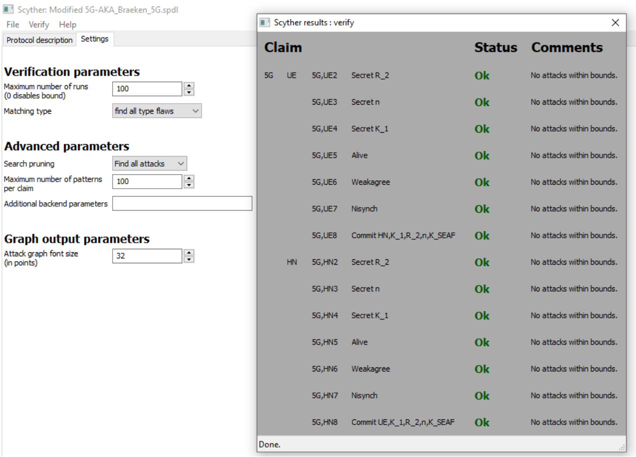
Additional backend parameters:

Graph output parameters

Attack graph font size (in points): 32

Claim				Status	Comments
5G	UE	5G,UE2	Secret R_2	Ok	No attacks within bounds.
		5G,UE3	Secret n	Ok	No attacks within bounds.
		5G,UE4	Secret K_1	Ok	No attacks within bounds.
		5G,UE5	Alive	Ok	No attacks within bounds.
		5G,UE6	Weakagree	Ok	No attacks within bounds.
		5G,UE7	Nisynch	Ok	No attacks within bounds.
		5G,UE8	Commit HN,K_1,R_2,n,K_SEAF	Ok	No attacks within bounds.
	HN	5G,HN2	Secret R_2	Ok	No attacks within bounds.
		5G,HN3	Secret n	Ok	No attacks within bounds.
		5G,HN4	Secret K_1	Ok	No attacks within bounds.
		5G,HN5	Alive	Ok	No attacks within bounds.
		5G,HN6	Weakagree	Ok	No attacks within bounds.
		5G,HN7	Nisynch	Ok	No attacks within bounds.
		5G,HN8	Commit UE,K_1,R_2,n,K_SEAF	Ok	No attacks within bounds.

Figure 4: Scyther tool result for proposed protocol



Verification parameters

Maximum number of runs (0 disables bound): 100

Matching type: find all type flaws

Advanced parameters

Search pruning: Find all attacks

Maximum number of patterns per claim: 100

Additional backend parameters:

Graph output parameters

Attack graph font size (in points): 32

Claim				Status	Comments
5G	UE	5G,UE2	Secret R_2	Ok	No attacks within bounds.
		5G,UE3	Secret n	Ok	No attacks within bounds.
		5G,UE4	Secret K_1	Ok	No attacks within bounds.
		5G,UE5	Alive	Ok	No attacks within bounds.
		5G,UE6	Weakagree	Ok	No attacks within bounds.
		5G,UE7	Nisynch	Ok	No attacks within bounds.
		5G,UE8	Commit HN,K_1,R_2,n,K_SEAF	Ok	No attacks within bounds.
	HN	5G,HN2	Secret R_2	Ok	No attacks within bounds.
		5G,HN3	Secret n	Ok	No attacks within bounds.
		5G,HN4	Secret K_1	Ok	No attacks within bounds.
		5G,HN5	Alive	Ok	No attacks within bounds.
		5G,HN6	Weakagree	Ok	No attacks within bounds.
		5G,HN7	Nisynch	Ok	No attacks within bounds.
		5G,HN8	Commit UE,K_1,R_2,n,K_SEAF	Ok	No attacks within bounds.

Figure 5: Scyther tool result for proposed protocol

8. Performance Measurements

In this section, we conduct a series of tests such as security characteristic examination and overhead analysis to assess the effectiveness of the proposed protocol.

8.1. MIRACL-based testbed experiments

This section discusses the results of our experiments on two different testbeds. For each testbed, we measured the

Table 2
Computational time for a server of cryptographic-primitives using MIRACL

Primitives	longest. Time (ms)	shortest. Time (ms)	Average time(ms)
T_H	0.00364	0.00301	0.00321
T_{AES}	0.00399	0.00326	0.00356
T_{RSA}	4.81	4.61	4.69
T_{INV}	0.0192	0.0162	0.0165

Table 3
Computational time under Raspberry PI 4 setting for cryptographic-primitives using MIRACL

Primitives	longest Time (ms)	shortest Time (ms)	Average time(ms)
T_H	0.038	0.029	0.0315
T_{AES}	0.045	0.038	0.041
T_{RSA}	8.25	8.04	8.14
T_{INV}	0.23	0.2	0.21

computational time required for the various cryptographic primitives using the widely used "MIRACL library" [17]. MIRACL is a "C/C++ based programming software library that has been already acknowledged as the standard library by the cryptographers [34, 35] for cryptographic primitives." The symbols T_H, T_{AES}, T_{RSA} and T_{INV} are used to represent computation time required to run "one-way hash function (SHA-256), (AES-128) encryption/decryption, (RSA-2048) encryption/decryption and modular inversion operation," respectively. We performed the experiments in two different scenarios: Desktop environment and Raspberry environment, respectively.

- **Scenario-1: A desktop as the Home Network (*HN*):** The first approach is implemented using the following configuration: Intel(R) Core(TM) i7-3770 with 3.40 GHz clock, 8 GB RAM running Linux Ubuntu 18.04.6 LTS. We execute the each cryptographic operations 100 times to compute the average run-time (ms) based on the longest and shortest run time (ms). Table 2 shows the experimental outcomes.

- **Scenario-2: A Raspberry Pi as the IoT Device (*UE*):** The second approach is implemented using the following configuration: a Raspberry Pi (Model: 4B, CPU: ARM® Cortex®-A7, Cores: 4, and RAM: 8GB) was deployed as (*UE*). We execute the each cryptographic operations 100 times to compute the average run-time (ms) based on the longest and shortest run time (ms). Table 3 shows the experimental outcomes.

8.2. Security properties comparison

This section contains an informal assessment of the proposed protocol in terms of security and functionality (mutual authentication, session-unlinkability or Unlinkability, privacy attack protection, resistance against replay attack, perfect forward secrecy, resistance against session temporary key material leakage, impersonation attack protection, resistance against stolen device attack, resistance against de-synchronization attack, non-repudiation, resistance against DoS attack).

Table 4
Comparison of security characteristics/ NOTE: V_1: mutual authentication; V_2: Session-unlinkability or Unlinkability; V_3: Privacy attack protection; V_4: Resistance against replay attack; V_5: Prefect forward secrecy; V_6: Resistance against session temporary key material leakage; V_7: Impersonation attack protection; V_8: Resistance against stolen device attack; V_9: De-synchronization attack protection; V_{10}: Non-repudiation; V_{11}: Resistance against DoS attack; V_{12}: Testbed experiments; V_{13}: Performance under unknown attack; V_{14}: Formal Analysis / Protocols-P/ √-provides the security, ×-fail to provide the security.

P	[3]	[13]	[14]	[15]	[16]	ours
V_1	√	√	√	√	√	√
V_2	×	×	√	√	√	√
V_3	√	√	√	√	√	√
V_4	×	√	√	√	√	√
V_5	×	×	×	×	√	√
V_6	×	×	×	√	×	√
V_7	√	√	√	√	√	√
V_8	×	×	√	√	√	√
V_9	×	×	√	√	√	√
V_{10}	×	×	√	×	×	√
V_{11}	×	√	√	√	×	√
V_{12}	×	√	×	×	×	√
V_{13}	×	×	×	×	×	√
V_{14}	×	AVISPA tool	Proverif, Scyther tool	Rubin logic	×	RoR logic, GNY logic, Scyther tool

The comparison result of Table.4 clearly indicates that the proposed protocol is robust against all the security requirements as well as offers the most desirable security features such as perfect forward secrecy, resistance against session temporary key material leakage, resistance against stolen device attack, non-repudiation, de-synchronization attack prevention, session-unlinkability or Unlinkability as mentioned in Section 3.3. The proposed protocol offers additional benefits because the secret keys and other parameters such as random number, identities and identifiers utilized in the message exchange have been updated after each successful authentication. Hence, it is quite obvious from the analysis that the proposed protocol provides better security as compared to [3, 13, 14, 16]. However, as compared to [15], our proposed protocol provides additional security feature such as perfect forward secrecy.

8.3. Computation cost

In this section, we compute the number of cryptographic operations used in the proposed protocol and compares it to other existing protocols. For comparison, the time of cryptographic operations of MIRACL presented in 8.1 has been used. For a server, we utilise the average computational

Table 5
Comparison of computation cost for mutual authentication protocols

Protocols	*UE* side	*HN* side	Total time (ms)
[3]	$T_{RSA} + 8T_H$	$9T_H$	8.43
[13]	$21T_H$	$12T_H$	0.8
[14]	$4T_H + 2T_{AES} + 17T_{INV}$	$4T_H + 2T_{AES} + 15T_{INV}$	4.2
[15]	$6T_H$	$9T_H$	0.22
[16]	$11T_H$	$16T_H$	0.4
Ours	$8T_H$	$16T_H$	0.3

Table 6
Comparison of communication cost for proposed protocols/ No. of bits for identity=64 bit , timestamp and random number= 160 bits. No. of bits required for AES symmetric enc/dec= 128 bits. No. of bits for Hashed output= 256 bits. No. of bits for public key enc/dec RSA = 2048 bits.

Protocols	[3]	[13]	[14]	[15]	[16]	Ours
Total (bits)	2720	2560	2048	2304	2880	2304

time for various cryptographic operations given in Table 2 whereas, for *UE*, we utilise the average computational time for various cryptographic operations given in Table 3. We ignore the bitwise XOR operation as the time needed for an XOR operation is negligible in comparison to other operations. The proposed protocol takes $8T_H$ at *UE* side and $16T_H$ at *HN* side which is ≈ 0.3 (ms), while [15] requires $6T_H$ at *UE* side and $9T_H$ at *HN* side which is ≈ 0.22 (ms) and [16] requires $(11T_H)$ at *UE* side and $16T_H$ at *HN* side which is $\approx$ 0.4 (ms) for authentication. The ration behind this is that the proposed protocol uses the hash function, which requires less cost than symmetric and asymmetric encryption. Therefore, we can infer that the proposed protocol is lightweight not only the protocols [3, 13, 14] that use a combination of symmetric and hash or combination of symmetric and asymmetric but also the protocol [16] that uses only hash function as shown in Table.5 and Figure 6a. However, compared to [15], the proposed protocol has a somewhat greater cost but offers additional security characteristics such as perfect forward secrecy.

8.4. Communication cost

In this section, the number of bits transferred in the channel during authentication for the proposed protocol is computed and compared to other protocols. We use the cost suggested by NIST [36, 37] to evaluate the communication cost. The proposed protocol requires $(A, B, F_1, f, I, J), (D_1, D_2, D_3, F_2) \approx 2304$ bits, while [15] requires $((a_n, b_n, y_n, Z_n, F_n), (\eta, \alpha, \beta, \mu)) \approx 2304$ bits and [16] requires $((a_n, b_n, y_n, Z_n, A, R, B, F_n), (\eta, \alpha, \beta, \mu)) \approx 2880$ bits which clearly indicate that proposed protocol requires less communication cost as compare to [3, 13, 16]. However, slightly higher than [14] and has the same as compared to [15] as indicated in Table.6 and Figure 6b.

Table 7
Comparison of storage cost for protocols

Protocols	[3]	[13]	[14]	[15]	[16]	Ours
Storage (bits)	2400	1056	1056	1216	1504	1000

Table 8
Comparison of energy consumption for protocols

Protocols	[3]	[13]	[14]	[15]	[16]	Ours
Energy Consumption (mj)	17.74	1.69	30.48	1.52	1.90	1.52

8.5. Storage cost

In this section, we compute the memory required to store the secrets in the device. We take the cost of operation and size of credentials same as suggested by NIST[37]. The proposed protocol requires $((A, B, K_1, f, n, UE_{id})) \approx$ 1000 bits while [15] requires $(a_n, b_n, id, c, n, K) \approx 1216$ bits and [16] requires $(a_n, b_n, id, c, n, cnt, K) \approx 1504$ bits in the device. The proposed protocol takes less storage because it does not need c as required in [15, 16]. The outcome of Table 7 and Figure 6c shows that proposed protocol requires very less storage as compared to [3, 13, 14, 15, 16]. Hence, we can infer that the proposed protocol is lightweight in terms of storage cost.

8.6. Energy efficient

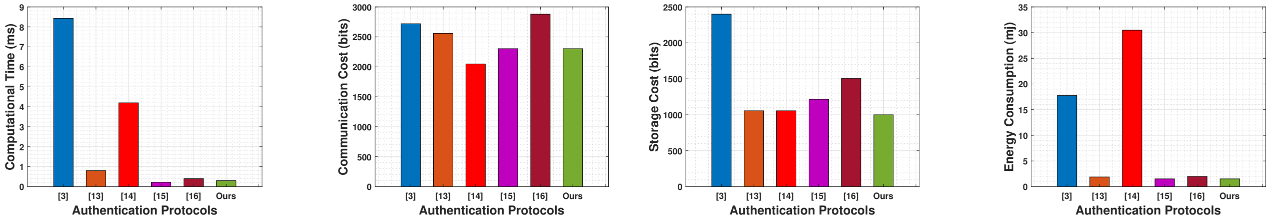
This section estimates the energy needed for the proposed protocol and compares it to existing protocols. We estimate the energy consumption similar to [23]. The energy usage of a "Strong ARM" CPU running at 133 MHz doing various task is summarised as energy required for transmitting a bit, AES symmetric enc/dec, Hashed output and public key enc/dec RSA is 0.00066 mj, 0.00217 mj, 0.000108 mj, 15.3 mj, respectively. The proposed protocol consumes $(2304 * .00066 + 24 * .000108) \approx 1.52$ mj, whereas [15] consumes $(2304 * .00066 + 15 * .000108)$ ≈ 1.52 (mj) and [16] consumes $(2880 * .00066 + 27 * .000108) \approx 1.90$ (mj). The outcome of Table 8 and Figure 6d shows that proposed protocol consumes very less energy as compared to [3, 13, 14, 16] and has same as compared to [15].

8.7. Message exchange

This section shows comparison of the number of messages exchanged of the proposed protocol with the existing protocols. The comparison outcome shown in Table 9 clearly shows that the proposed protocol requires only two message exchanges that is less as compared to [3, 13, 14] which is approximately half as compared to 5G-AKA and equal to [15, 16].

8.8. Performance under unknown attacks

This section presents the energy consumption, computational, and communication overhead analysis under the unknown attacks same as [31, 38, 39]. Although we have



(a) (b) (c) (d)

Figure 6: Comparison of (a) Computational cost (b) Communication cost (c) Storage cost, and (d) Energy consumption of proposed protocol with its competitors.

Table 9
Comparison of message exchange for protocols

Protocols	[3]	[13]	[14]	[15]	[16]	Ours
Message exchange	5	4	3	2	2	2

demonstrated that our proposed protocol is resistant to all identified known attacks as mentioned in Section 3.3, there will undoubtedly be some unknown attacks that we will be unable to predict when they happen. In order to access the performance under unknown attacks, we assume that when an unknown attack occurs, the authentication process will be terminated.

$$Cost_{average} = \frac{Cost_{Success} \times (1-P) + Cost_{fail} \times P}{(1-P)} \quad (7)$$

We use the Eq. (7) to analyse the performance under unknown attacks. Where $Cost_{average}$ represents the average computational/communication/ energy consumption overhead under unknown attacks. $Cost_{Success}$ represents total computational/communication/ energy consumption overhead for successful authentication, This has a probability of P, the step in which the unknown attack happens is entirely random, i.e., the chance of an unknown attack occurring in step j is $1/n$, where n is the total number of signalling messages in a single execution of the protocol and $Cost_{fail}$ (i.e. shown in Eq 8), represents the computational/communication/ energy consumption overhead when unknown attack happen before step j.

$$Cost_{fail} = \sum_{j=1}^{n} Cost_j * \frac{1}{n} \quad (8)$$

Table. 10 represents the average computational, communication, and energy consumption overhead under unknown attack and Table. 11 and Figure 7 represents the impact on performance when unknown attack occur. It is quite clear that the proposed protocol takes very less overhead as compared to most of its competitors. The reason behind this is that our proposed protocol requires less computational/communication/ energy consumption cost as compared to its competitors. Therefore, the proposed protocol also performs better under unknown attack as compared to [3, 13, 14, 16]. However, as compared to [15], the proposed protocol has a slightly greater average computational overhead, but it takes the same amount of energy and has the same communication overhead under an unknown attack.

8.9. Discussion on comparison results

In this section, to demonstrate the effectiveness of the proposed proposed protocol, we summarize the results obtained in previous section.

- The comparison of security features of [3, 13, 14, 15, 16] with the proposed protocol, shown in Table.4, clearly indicates that the proposed protocol provides better security and offers extra security features such as perfect forward secrecy, resistance against session temporary key material leakage, resistance against stolen device attack, non-repudiation, desynchronization attack prevention, session un-linkability or untraceability.
- We also compare the performance of the proposed protocol with [3, 13, 14, 15, 16] which shows that proposed protocol takes very less costs. Although, our proposed protocol takes slightly higher computational cost than [15] but provides the most desirable security features such as perfect forward secrecy which [15] does not provide.
- The comparative analysis shows that proposed protocol reduces the computational cost by ≈ 97%, 63%, 93%, 25% as compared to [3, 13, 14, 16], communication cost is reduced by 16%, 10%, 0%, 20% with respect to [3, 13, 15, 16], storage cost by 59%, 20%, 6%, 23%, 35% with respect to [3, 13, 14, 15, 16] and energy consumption by 92%, 11%, 96%, 21% with respect to [3, 13, 14, 16].
- We also analyse the performance of the proposed protocol and its counterparts under unknown attacks, revealing that the proposed protocol outperforms most of its competitors when unknown attacks occur.
- Thus, the comparison analysis shows that our proposed protocol provides better security, takes less computation, communication, storage and energy consumption costs and requires less overhead under unknown attack as compared to most of its counterparts.

Table 10
Average computational/ communication/ energy consumption overhead under unknown attacks for protocols

Protocols	Computational Overhead (ms)	Communication Overhead (bits)	Energy Overhead (mj)
[3]	(P×(9.8147)/(1-P))+4.804	((P×5561)/(1-P))+2720	((P×38.0476))/(1-P))+17.74
[13]	((P×0.961)/(1-P))+1.802	((P×5055)/(1-P))+2560	((P×3.03358)/(1-P))+1.69
[14]	((P×7.6146)/(1-P))+7.728	((P×1776)/(1-P))+2048	((P×364.30)/(1-P))+30.48
[15]	((P×.3975)/(1-P))+0.795	((P×2432)/(1-P))+2304	((P×1.6059)/(1-P))+1.52
[16]	((P×0.689)/(1-P))+1.378	((P×2912)/(1-P))+2880	((P×1.7337)/(1-P))+1.9036
Ours	((P×0.6625)/(1-P))+1.272	((P×2432)/(1-P))+2304	((P×1.6103)/(1-P))+1.52

Table 11
Comparison of computational overhead/ Communication overhead / Energy overhead under unknown attacks for protocols/ P-Probability, O- Proposed protocol.

P	Computation overhead (ms)						Communication overhead (bits)						Energy overhead (mj)					
	[3]	[13]	[14]	[15]	[16]	O	[3]	[13]	[14]	[15]	[16]	O	[3]	[13]	[14]	[15]	[16]	O
0.1	5.8	1.9	8.5	.83	1.4	1.3	3337	3121	2245	2574	3203	2574	21.9	2.0	70.9	1.6	2.0	1.6
0.2	7.2	2.0	9.6	.89	1.5	1.4	4110	3823	2492	2912	3608	2912	27.25	2.4	121.4	1.9	2.3	1.9
0.3	9.0	2.2	10.9	.96	1.6	1.5	5103	4726	2809	3346	4128	3346	34.0	2.9	186.4	2.2	2.6	2.2
0.4	11.3	2.4	12.8	1.0	1.8	1.7	6427	5930	3232	3925	4821	3925	43.1	3.7	273.1	2.5	3.0	2.5
0.5	14.6	2.7	15.3	1.1	2.0	1.9	8281	7615	3824	4736	5792	4736	55.7	4.7	394.5	3.1	3.6	3.1
0.6	19.5	3.2	19.1	1.3	2.4	2.2	11061	10142	4712	5952	7248	5952	74.8	6.2	576.5	3.9	4.5	3.9
0.7	27.6	4.0	25.4	1.7	2.9	2.8	15695	14355	6192	7978	9674	7978	106.5	8.8	879.9	5.2	5.9	5.2
0.8	44.0	5.6	38.1	2.3	4.6	3.9	24964	22780	9152	12032	14528	12032	169.9	13.8	1486.6	7.9	8.8	7.9
0.9	93.3	10.4	76.2	4.3	7.5	7.2	52769	48055	18032	24192	29088	24192	360.1	29.1	3306.8	15.9	17.5	16

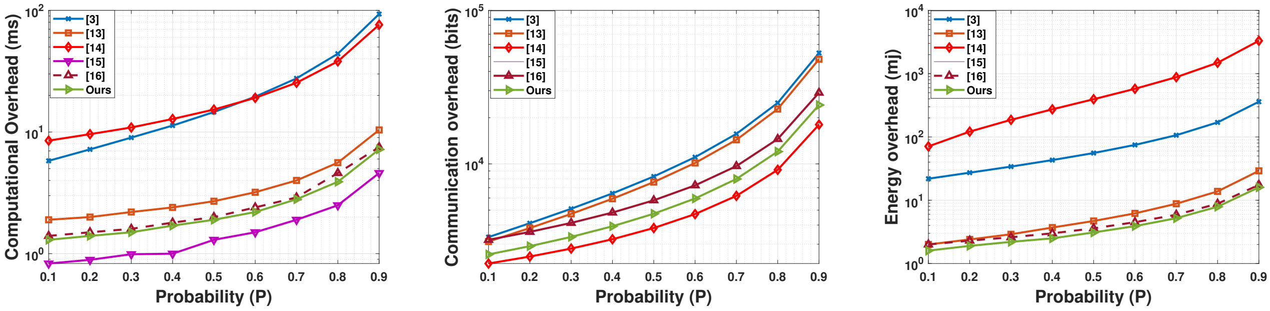
Therefore, we can conclude that our proposed protocol outperform [3, 13, 14, 16]. However as compared to [15], our proposed protocol takes slightly higher computational cost but adds security feature like perfect forward secrecy, requires less storage cost, and has the same communication cost and energy consumption.

8.10. Limitations of the proposed protocol

This section outlines the limitation of the proposed protocol. Although we have shown that the proposed protocol is secure and requires less cost compared to its competitors, there are some limitations that we will explore here and would like to address in the future. The following are the limitations of the proposed protocol.

- For the proposed protocol, we consider that the attacker has no access to the secrets stored in the tamper-resistant hardware, being $\{K_1, UE_{id}, f, A, B, n\}$ in the UE and $\{K_m, (UE_{id}, K[0] = K_1, K[1] = K), n)\}$ at the HN, same as [15, 16, 19]. This is due to the fact that it can be believed that physical access to the servers that contain such secrets is challenging, and tamper-proof security is very high [15].

- The proposed protocol provides the session-unlinkability or unlinkability, not the full unlinkability. Since [16,



(a) (b) (c)

Figure 7: Comparison of (a) Computational (b) Communication (c) Energy consumption overhead of proposed protocol with its competitors.

40], states that with symmetric-key based architectures, there is a trade-off between privacy and availability, and any effort to enhance privacy has unavoidable impacts on the security against DoS (Denial of Service) attacks.

- Instead of the 5G-AKA protocol, we do not communicate to the UE the difference in type of error between synchronization or MAC failure. Similar as in the 5G-AKA, we also assume the existence of a threshold on acceptable number of false requests. In our case, this number includes both types of errors and not only the synchronization errors like in 5G-AKA. In fact, the definition of this threshold is very important. If it is put too high, the scheme is vulnerable for DoS attacks, while if it is too low, the level of userfriendliness is not acceptable. Depending on the current threat level, this number should be defined, which is outside the scope of this paper.

9. Conclusion

In this work, we investigate the security of two recently proposed papers [15, 16] and find that [15] is vulnerable to perfect forward secrecy, while [16] is prone to DoS attack if the server database size is large and has a high cost. In light of this, we designed an improved authentication protocol that is superior in terms of security features and cost-effective to most of its competitors. We informally (non-mathematical) verify the proposed protocol's security properties, demonstrating that it provides robust security against all identified attacks as well as additional security features. Furthermore, the formal (mathematical) verification of the proposed protocol is done using the ROR logic, GNY logic, and the validation tool Scyther, demonstrating that it is resistant to all identified attacks and securely generates the session key. The comparison of the proposed protocol's performance with its competitors in terms of energy consumption and computational, communication, and storage costs demonstrates that the proposed protocol is lightweight. It also has lesser overhead as compared to most of its competitors under unknown attacks. Hence, we conclude that our proposed protocol provides better trade-off between security and performance compared to its competitors.

In the future, we would like to enhance our authentication protocol to support full-unlinkability and group authentication for 5G-based IoT applications.

References

[1] GSMA the mobile economy, 2020. https://www.gsma.com/mobileeconomy/wpcontent/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf.

[2] 3GPP, Study on scenarios and requirements for next generation access technologies, (3GPP), TR 38.913, 2020, Available Online:. https://www.3gpp.org/ftp//Specs/archive/38_series/38.913/.

[3] 3GPP, Security architecture and procedures for 5G system, (3GPP), TS 33.501, 2020, Available Online:. https://www.3gpp.org/ftp/Specs/archive/33_series/33.501/.

[4] Khan, H.; Dowling, B.; Martin, K. M. Identity confidentiality in 5G mobile telephony systems. International Conference on Research in Security Standardisation. 2018; pp 120–142.

[5] Gharsallah, I.; Smaoui, S.; Zarai, F. A secure efficient and lightweight authentication protocol for 5g cellular networks: Sel-aka. 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). 2019; pp 1311–1316.

[6] Braeken, A.; Liyanage, M.; Kumar, P.; Murphy, J. Novel 5G authentication protocol to improve the resistance against active attacks and malicious serving networks. *IEEE Access* **2019**, *7*, 64040–64052.

[7] Wang, Y.; Zhang, Z.; Xie, Y. Privacy-Preserving and Standard-Compatible {AKA} Protocol for 5G. 30th {USENIX} Security Symposium ({USENIX} Security 21). 2021.

[8] Ramadan, M.; Liao, Y.; Li, F.; Zhou, S. Identity-based signature with server-aided verification scheme for 5G mobile systems. *IEEE Access* **2020**, *8*, 51810–51820.

[9] Koutsos, A. The 5G-AKA authentication protocol privacy. 2019 IEEE European Symposium on Security and Privacy (EuroS&P). 2019; pp 464–479.

[10] Liu, T.; Wu, F.; Li, X.; Chen, C. A new authentication and key agreement protocol for 5G wireless networks. *Telecommunication Systems* **2021**, 1–13.

[11] Hojjati, M.; Shafieinejad, A.; Yanikomeroglu, H. A Blockchain-Based Authentication and Key Agreement (AKA) Protocol for 5G Networks. *IEEE Access* **2020**, *8*, 216461–216476.

[12] Parne, B. L.; Gupta, S.; Gandhi, K.; Meena, S. PPSE: Privacy Preservation and Security Efficient AKA Protocol for 5G Communication Networks. 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). 2020; pp 1–6.

[13] Choudhury, H. HashXor: A lightweight scheme for identity privacy of IoT devices in 5G mobile network. *Computer Networks* **2021**, *186*, 107753.

[14] Cao, J.; Yan, Z.; Ma, R.; Zhang, Y.; Fu, Y.; Li, H. LSAA: a lightweight and secure access authentication scheme for both UE and mMTC devices in 5G networks. *IEEE Internet of Things Journal* **2020**, *7*, 5329–5344.

[15] Braeken, A. Symmetric key based 5G AKA authentication protocol satisfying anonymity and unlinkability. *Computer Networks* **2020**, *181*, 107424.

[16] Munilla, J.; Burmester, M.; Barco, R. An enhanced symmetric-key based 5G-AKA protocol. *Computer Networks* **2021**, 108373.

[17] MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library. (2022). Accessed: Junary 2022. [Online]. Available:. https://github.com/miracl/MIRACL.

[18] Basin, D.; Dreier, J.; Hirschi, L.; Radomirovic, S.; Sasse, R.; Stettler, V. A formal analysis of 5G authentication. Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018; pp 1383–1396.

[19] Liu, Y.; Huo, L.; Zhou, G. TR-AKA: A two-phased, registered authentication and key agreement protocol for 5G mobile networks. *IET Information Security* **2022**, *16*, 193–207.

[20] Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Transactions on information theory* **1983**, *29*, 198–208.

[21] Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. International Conference on the Theory and Applications of Cryptographic Techniques. 2002; pp 337–351.

[22] Abdalla, M.; Fouque, P.-A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. International Workshop on Public Key Cryptography. 2005; pp 65–84.

[23] Xu, Z.; Li, X.; Xu, J.; Liang, W.; Choo, K.-K. R. A secure and computationally efficient authentication and key agreement scheme for Internet of Vehicles. *Computers & Electrical Engineering* **2021**, *95*, 107409.

[24] Lee, J.; Kim, G.; Das, A. K.; Park, Y. Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks. *IEEE Transactions on Network Science and Engineering* **2021**, *8*, 2412–2425.

[25] Gong, L.; Needham, R. M.; Yahalom, R. Reasoning about Belief in Cryptographic Protocols. IEEE Symposium on Security and Privacy. 1990; pp 234–248.

[26] Liu, H.; Yao, X.; Yang, T.; Ning, H. Cooperative privacy preservation for wearable devices in hybrid computing-based smart health. *IEEE Internet of Things Journal* **2018**, *6*, 1352–1362.

[27] Sureshkumar, V.; Anandhi, S.; Amin, R.; Selvarajan, N.; Madhumathi, R. Design of robust mutual authentication and key establishment security protocol for cloud-enabled smart grid communication. *IEEE Systems Journal* **2020**, *15*, 3565–3572.

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

[29] Cao, J.; Ma, M.; Fu, Y.; Li, H.; Zhang, Y. CPPHA: Capability-based privacy-protection handover authentication mechanism for SDN-based 5G HetNets. *IEEE transactions on dependable and secure computing* **2019**, *18*, 1182–1195.

[30] Ma, R.; Cao, J.; Feng, D.; Li, H.; He, S. FTGPHA: Fixed-trajectory group pre-handover authentication mechanism for mobile relays in 5G high-speed rail networks. *IEEE transactions on vehicular technology* **2019**, *69*, 2126–2140.

[31] Cao, J.; Yu, P.; Xiang, X.; Ma, M.; Li, H. Anti-quantum fast authentication and data transmission scheme for massive devices in 5G NB-IoT system. *IEEE Internet of Things Journal* **2019**, *6*, 9794–9805.

[32] Yadav, A. K.; Misra, M.; Pandey, P. K.; Kaur, K.; Garg, S.; Liyanage, M. LEMAP: A Lightweight EAP based Mutual Authentication Protocol for IEEE 802.11 WLAN. ICC 2022 - IEEE International Conference on Communications. 2022; pp 692–697.

[33] Shunmuganathan, S. A Reliable Lightweight Two Factor Mutual Authenticated Session Key Agreement Protocol for Multi-Server Environment. *Wireless Personal Communications* **2021**, *121*, 2789–2822.

[34] Das, A. K.; Kumar, N.; Alazab, M., et al. Designing authenticated key management scheme in 6G-enabled network in a box deployed for industrial applications. *IEEE Transactions on Industrial Informatics* **2020**,

[35] Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A. K. Design of Blockchain-Based Lightweight V2I Handover Authentication Protocol for VANET. *IEEE Transactions on Network Science and Engineering* **2022**,

[36] Tsobdjou, L. D.; Pierre, S.; Quintero, A. A New Mutual Authentication and Key Agreement Protocol for Mobile Client—Server Environment. *IEEE Transactions on Network and Service Management* **2021**, *18*, 1275–1286.

[37] Barker, E.; Dang, Q. Nist special publication 800-57 part 1, revision 4. *NIST, Tech. Rep* **2016**, *16*.

[38] Ma, R.; Cao, J.; Feng, D.; Li, H.; Li, X.; Xu, Y. A robust authentication scheme for remote diagnosis and maintenance in 5G V2N. *Journal of Network and Computer Applications* **2021**, 103281.

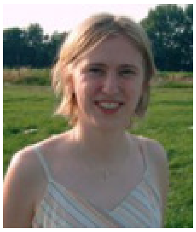
[39] Yadav, A. K.; Misra, M.; Pandey, P. K.; Liyanage, M. An EAP-Based Mutual Authentication Protocol for WLAN connected IoT devices. *IEEE Transactions on Industrial Informatics* **2022**, 1–12.

[40] Burmester, M.; Munilla, J. Pre vs post state update: Trading privacy for availability in RFID. *IEEE Wireless Communications Letters* **2014**, *3*, 317–320.

Awaneesh Kumar Yadav is doing Ph.D in the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, India. His research area includes Network security, 5G security, Network Slice, formal verification, IoT & Cloud Security. He did his M.Tech in the Department of Computer Science and Engineering, National Institute of Technology Rourkela, India, in 2019.

Manoj Misra is a Professor in Department of Computer Science and Engineering at IIT Roorkee. Dr. Misra got his PhD from University of New Castle upon Tyne and has past experience of working as an Engineer at CMC Limited Noida, Assistant Engineer at Hindustan Aeronautic Limited at Kanpur India, Assistant Professor at HBTI Kanpur. His research interests include Distributed Computing, Performance Evaluation, Computer Networks, Network Security and Cyber frauds.

Pradumn Kumar Pandey received the Bachelor of Technology and Ph.D. degrees in computer science and engineering from IIT Jodhpur, Jheepasani, India, in 2012 and 2018, respectively. He was an Institute Post-Doctoral Fellow with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur, India, from May to September 2018. He worked as a DST INSPIRE Faculty Member with the Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India, from October 2018 to October 2019, where he has been working as an Assistant Professor since November 2019. His research areas include modeling of complex networks, information diffusion on real networks, social security on online social networks, and network representation learning



An Braeken is full time professor at VUB-INDI. Her interests include lightweight security and privacy protocols for IoT, cloud and fog, blockchain and 5G security. She has developed several lightweight security solutions in the healthcare domain in collaboration with University of Oulu, University College Dublin and University of Oxford. She is (co-)author of over 200 publications. She has been member of the program committee for numerous conferences and workshops and member of the editorial board for Security and Communications magazine. In addition, she is since 2015 expert reviewer for several EU calls. She has cooperated and coordinated more than 15 national and international projects.

Madhusanka Liyanage is an Assistant Professor/Ad Astra Fellow and Director of Graduate Research at the School of Computer Science, University College Dublin, Ireland. He is also acting as a Docent/Adjunct Professor at the Center for Wireless Communications, University of Oulu, Finland, and Honorary Adjunct Professor at the Department of Electrical and Information Engineering, University of Ruhuna, Sri Lanka. He received his Doctor of Technology degree in communication engineering from the University of Oulu, Oulu, Finland, in 2016. From 2011 to 2012, he worked as a Research Scientist at the I3S Laboratory and Inria, Sophia Antipolis, France. He was also a recipient of the prestigious Marie Skłodowska-Curie Actions Individual Fellowship and Government of Ireland Postdoctoral Fellowship during 2018-2020. During 2015-2018, he has been a Visiting Research Fellow at the CSIRO, Australia, the Infolabs21, Lancaster University, U.K., Computer Science and Engineering, The University of New South Wales, Australia, School of IT, University of Sydney, Australia, LIP6, Sorbonne University, France and Computer Science and Engineering, The University of Oxford, U.K. He is also a senior member of IEEE. In 2020, he received the "2020 IEEE ComSoc Outstanding Young Researcher" award by IEEE ComSoc EMEA. In 2021, he was ranked among the World's Top 2% Scientists (2020) in the List prepared by Elsevier BV, Stanford University, USA. Also, he was awarded an Irish Research Council (IRC) Research Ally Prize as part of the IRC Researcher of the Year 2021 awards for the positive impact he has made as a supervisor. Dr. Liyanage's research interests are 5G/6G, SDN, IoT, Blockchain, MEC, mobile, and virtual network security. More info: www.madhusanka.com