A Provably Secure and Efficient 5G-AKA Authentication Protocol using Blockchain

Awaneesh Kumar Yadav*, An Braeken†, Manoj Misra‡, Madhusanka Liyange§
*‡Dept. of Computer Science and Engineering, Indian Institute of Technology Roorkee, Uttarakhand, India.
†Department of engineering, technology (INDI), Vrije Universiteit Brussel, Belgium.
‡School of Computer Science, University College Dublin, Ireland and CWC, University of Oulu, Finland
Email:*akumaryadav@cs.iitr.ac.in, †an.braeken@vub.be, ‡manoj.misra@cs.iitr.ac.in, §madhusanka@ucd.ie,

***Abstract*—The next generation of mobile communication systems must be secured because of the ongoing entrance of numerous security attacks. Thus, to secure the underlying network, the 3GPP has designed an authentication and key agreement protocol, 5G-AKA, to safely and stably access the mobile services. However, some recent observations indicate that 5G-AKA has numerous shortcomings such as perfect forward secrecy violation, malicious Serving Network (SN), de-synchronization attack, privacy theft, stolen device, and denial of Service (DoS) attacks when the user uses the roaming mobile services. Considering the shortcomings of existing protocols and the requirement to offer increased security, we propose a provable secure, efficient 5G-AKA authentication protocol using the blockchain. The security features of the proposed protocol are examined using the Real-Or-Random (ROR) logic and Scyther tool. Furthermore, the performance of the proposed protocol is evaluated, which shows that it is the least costly compared to its counterparts in terms of computational and communication costs. In addition, the comparison of the Ethereum blockchain depicts that the proposed protocol takes less transaction and execution costs compared to its counterparts.**

***Keywords*—5G-AKA, Authentication, ECC, Cloud Computing, Internet of Things (IoT), ROR logic, Network Security**

I. Introduction

The development of fifth-generation mobile networks and communication services has led to the development of an authentication and key agreement scheme for 5G communication. This scheme provides high data rates, multiple device connections, and low service latency needed in new applications like the Industrial Internet of Things (IIoT), autonomous vehicles, smart homes, logistics, and hospitals [1]. As stated in literature [2], 4G is vulnerable to privacy theft and does not meet the high-speed criteria. To deliver fast and secure communication, 5G relies on three main components: massive machine-type communications (mMTC), improved mobile broadband (eMBB), and ultra-reliable and low latency communications. The eMBB is utilized to deliver high speed, mMTC allows numerous devices to communicate with one another, and uRLLC ensures low latency [3]. In spite of the fact that it delivers high speed and privacy protection, it has been reported that it has a number of other weaknesses, including traceability attacks, violations of perfect forward secrecy, malicious Serving Networks (SN), DoS attacks, and privacy theft which could impede the implementation of 5G in critical applications [4]. As a result, many symmetric and asymmetric-based authentication protocols have been proposed by security experts to address these types of security issues. The symmetric encryption-based authentication protocols [3]–[6] do not meet the prominent security features even though they offer lightweight protocols. To address this, a number of asymmetric-based authentication protocols have been proposed, including [7]–[10]. They offer superior security compared to symmetric encryption, but they are costly and vulnerable to several types of attacks. Most importantly, neither symmetric nor asymmetric authentication methods function well in roaming scenarios (i.e., when UE roams to and connects to a different SN that is out of the HN coverage area, the SN operator must ask the UE's home operator for their subscription information). These protocols are challenging to implement in roaming situations without a secure link between the SN and HN. In order to address this problem, Hojjati et al. [11] designed an authentication protocol that does not require a secure channel between the SN and HN. In addition, the method offers security from DoS attacks using blockchain technology. However, this protocol is prone to impersonation attacks and fails to ensure perfect forward secrecy [12]. It is therefore imperative to create an authentication system that can solve the aforementioned issues while still being suitable for roaming scenarios.

A. Motivation & Contributions

The majority of symmetric and asymmetric-based authentication protocols require a secure channel between the SN and HN, making them unsuitable for real-world implementations when an insecure channel is necessary, or UE enters a cell that is not within the HN coverage area. There is currently one protocol proposed in the literature [11] specifically for situations where the SN and HN use the insecure channel (i.e., roaming scenarios). Security checks in [12] show that this technique does not maintain perfect forward secrecy and is open to impersonation attacks.

Therefore, this research aims to propose an authentication mechanism that addresses the aforementioned security concerns while requiring the insecure route between the SN and HN.

Our contributions are as follows:

- We propose an improved Blockchain-based authentication and key agreement protocol for 5G communication using ECC that offers perfect forward secrecy and impersonation attack protection.
- The security of the proposed protocol is investigated using the ROR logic and Scyther tool to show that there is no attack on the protocol and the session key is generated

securely.

- The comparative analysis and comparison depicts that the proposed protocol is the least expensive in terms of computational and communication costs when compared to its competitors.
- The Ethereum blockchain comparison shows that the proposed protocol has lower transaction and execution costs than its competitors.

II. Preliminaries and Backgrounds

This part explains the background knowledge of the approaches and concepts employed in the paper.

A. *5G Network Model for proposed protocol*

There are four entities involved in the 5G network model of the proposed protocol. The description of the involved entities is as follows.

- User Equipment (UE): Smartphones or Internet of Things (IoT) devices that are carried by users. Each UE has a Universal-Integrated circuit card (UICC) that contains the Universal-Subscriber Identity Module (USIM). The USIM stores the authentication key and secrets that have been pre-saved.
- Serving Network (SN): The SN is made up of two parts: the gNB, which provides the UE with a radio access network, and the Security Anchor Function (SEAF), which provides an interface between the UE and the HN to exchange messages.
- Blockchain: It primarily receives messages from the SEAF, checks their validity, and forwards them to the HN using the Authentication Server Function (AUSF), and vice versa.
- Home Network (HN): HN is made up of four entities, namely: AUSF, which is in charge of authentication and decision-making, Unified Data Management (UDM) saves authentication data, and assists other HN entities, Authentication Credentials and Repository and Processing Function (ARPF), which is in charge of selecting the appropriate authentication method depending on the user's identification and the policy that has been defined and Subscription Identity De-concealing Function (SIDF) decrypts the SUCI to obtain the SUPI.

All the entities of the network model communicate with each other through the insecure channel.

B. *Threat Model*

We suppose that an adversary can perform both active and passive attacks, as described by Dolev-Yao (DY) [13] and CK-adversary [14] threat models. As a result, the attacker can intercept the exchange messages and carry out the following actions: actively edit, delete, and insert (some sections), as well as determine the communicating entities' long-term private key to carry out the assaults.

C. *Blockchain Technology*

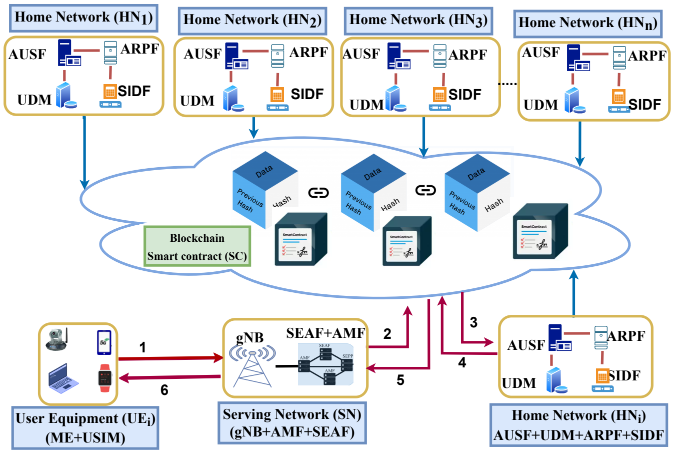


Fig. 1: Architecture of 5G-Network model for the proposed protocol.

Blockchain technology is a foundation for an unchangeable distributed ledger. The second generation of this technology enables the execution of smart contracts, which are pre-programmed transactions [15]. Blockchain for 5G authentication may serve as a barrier between the SN and the HN. Blockchain thus provides a secure channel for communication exchange. This prevents DoS attacks on the HN and improves user anonymity by blocking access from malicious SNs that can operate as active attackers. It also maintains a tamper-proof and auditable record of the authentication procedures. We make use of the blockchain-based smart contract function, which is software that runs automatically when specific conditions are satisfied. Simple "if/when...then" clauses recorded in code on a blockchain make up smart contracts. A network of computers performs the tasks after the predetermined conditions are met and validated [11].

D. *Elliptic Curve Cryptography*

An elliptic curve $E_m(x, y)$ is expressed as $a^2 = b^3 + xa + y(4x^3 + 27y^2)$ over the finite field F_m: where m and n are known as two large primes, and G is known as the subgroup of the additive group of points $E_m(x, y)$ with order n [16]. The prominent three mathematical computational problems of ECC are as follows.

1) Elliptic-curve discrete logarithm (ECDL) problem: Having xM will provide no inside to extract x, wherein $x \epsilon F_m^*$ and $F_m^* = \{1, 2, ...m - 1\}$.
2) Elliptic curve-computational Diffie-Hellman (ECCDH) problem: Having xM and yM will not let the attacker to compute xyM, where $(x, y, \epsilon F_m^*)$.
3) Elliptic curve-descisional Diffie-Hellman (ECDDH) problem: Having xM, yM, and wM will not provide any inside to the attacker to compute $wM == xyM$.

III. Proposed Protocol

This section presents the improved blockchain-based authentication and key agreement protocol to address the flaws such as violation of perfect forward secrecy and being prone to impersonation attacks that are in [11]. We employ ECC and some minor modifications in the improved protocol to offer the perfect forward secrecy and protection from impersonation attacks. The proposed protocol has three phases, namely: i) Initialization, ii) Registration phase and iii) Authentication and key agreement.

A. Initialization

In our proposed approach, each HN creates its own smart contract on a public blockchain. The smart contract address is made public (for instance, on the operator's official website). Each SN that intends to provide roaming service to an incoming HN's subscriber uses this smart contract. The HN may function on a pay-per-service basis thanks to the usage of a public blockchain, which eliminates the need to create any infrastructure. It also serves as an integration platform for mobile network operators (MNOs) who want to provide roaming services to one another [11].

B. Registration phase

The registration phase is performed to exchange the secrets between the communication entities using the secure channel same as [7]–[11]. Initially, UE sends the connection request to the HN, the HN receives the requests and selects a $SUPI$, K, counter value Ctr, a point $G \in E$ of order n and stores this information into the USIM and shares it with the UE. On the other hand, a secret key is shared between the SN and HN for the same communication as [10]. Table II and Fig 2 represents the notation used in the paper and the description of the protocol respectively.

TABLE I: Notations and meanings

Notations	Meanings
$SUPI$	Identity of UE
$E\&G$	Elliptic curve over F_m & A point of E of order n
$r_1, r_2, r_3\&$ K_{SEAF}	Random numbers & Session key for mutual- authentication

C. Authentication & key agreement phase

This section shows the authentication and key agreement process between the communication entities UE, SN and HN through the insecure channel. The following steps of authentication and key agreements are as follows:

- At the starting, SN chooses the random number S_1 and forwards $\langle M_1 = \{S_1, SN_{ID}\}\rangle$ to the UE.
- When UE obtains this, then it selects a random number r_1, computes $P_1 \leftarrow r_1 G$, $SUCI = \{E_{pk_{HN}}(SUPI, P_1, S_1, Ctr, SN_{ID}), HN_{ID}\}$, increments the Ctr by one and then forwards $\langle M_2 = \{SUCI\}\rangle$ to the SN.
- After receiving the $SUPI$, SN selects the random number r_2 in order to compute $P_2 \leftarrow r_2 G$, $ID_{req} = H(K_{SN-HN}, SN_{ID}, HN_{ID}, SUCI, P_2, S_1)$ and transmit $\langle M_2 = \{ID_{req}, P_2, SUCI, SN_{ID}\}\rangle$ to the HN.
- When UE obtains the message $\langle ID_{req}, P_2, SUCI, SN_{ID}\rangle$, then it checks the freshness of the ID_{req} in order to prevent a duplicate request. If the request is fresh then it forwards this to the HN, otherwise it rejects the request and the transaction will be reverted.
- When HN receives the message from UE, then it decrypts the $SUCI$ to obtain the credentials. After obtaining the credentials it computes the ID^*_{req} and compares it with the received ID_{req}. If it matches, then it compares $(Ctr == Ctr\&\&Ctr \in (Ctr_1, Ctr_2 ... Ctr + \Delta))$ in order to check the freshness. Next, it increments Ctr by one. Afterwards, HN selects the random number r_3 in order to compute $P_3 \leftarrow r_3 G, P_4 \leftarrow r_3 P_1, P_5 \leftarrow r_3 P_2$, $xMac = f_1(K_{UE-HN}, P_4, Ctr, SN_{ID})$, $Res = challenge(P_4, SN_{ID})$, $K_{SEAF} = KeySeed(K_{UE-HN}, P_4, Ctr)$, $hxRes = H(K_{SN-HN}, Res)$, $EK = E_{H(P_5, K_{SN-HN})}(SUPI, K_{SEAF})$, $ID_{res} \leftarrow H(hxRes, xMac, EK, K_{SN-HN})$ and forwards $\langle M_4 = \{EK, xMac, hxRes, ID_{req}, ID_{res}, P_3\}\rangle$ to UE.
- When UE receives the message, then it verifies that the messages sender is owner of the smart contract. We can put a condition that only the HN is allowed to register a response transaction to the smart contract.
- When SN obtains the message, then it keeps $(EK, hxRes, ID_{req}, ID_{res}, P_3)$ and forwards $(xMac, P_3)$ to the UE.
- After receiving the message from SN, UE computes the $P_6 \leftarrow r_1 P_3$ in order to compute $xMac = f_1(K_{UE-HN}, P_6, Ctr, SN_{ID})$, $Res = challenge(P_6, SN_{ID})$, $K_{SEAF} = KeySeed(K_{UE-HN}, P_6, Ctr)$. After computing this, it compares the $xMac^* == xMac$. If it matches then it believes that HN is authentic and sends the Res to SN for the key confirmation.
- After receiving the message from the UE, HN computes the $P_7 \leftarrow r_2 P_3$ in order to compute and compare the $H(K_{SN-HN}, Res) == hxRes$. If it matches, then it decrypts the EK to obtain $K_{SEAF}, SUPI$.

IV. Formal verification of the proposed protocol

This section describes how the proposed protocol is formally verified using ROR logic and the Scyther tool to demonstrate that it satisfies all security requirements.

A. Formal Security Analysis using ROR logic

In order to verify the session key security, this section depicts the formal verification of the proposed protocol using the well-known mathematical logic ROR model suggested by Abdalla *et al.* [17]. The proposed protocol involves the three participants (i.e., UE, SN and HN). Let us consider UE^r and HN^s, who denote the instances of r and s respectively. There are certain queries that the adversary (φ) uses to launch the real attack. The entity Z is denoted by the variable M^t in the following explanation. The description of ROR queries are outlined below

TABLE II: Notations and Meanings

Notations	Meanings
Execute (UE^r, HN^s)	φ issues this query to track the messages exchanged between instances UE^r and HN^s.
Reveal (M^t)	φ issues this query to acquire the session key between the instances UE^r and HN^s.
Test (M^t)	φ issues this query to test the security of the derived session key between the instances UE^r and HN^s. For that, a coin C is tossed by the φ in order to guess the outcome of the *Test query*.

Theorem 1: Let us assume that φ is trying to obtain the session key (SK) in polynomial time. Then $Ad_{\varphi} \leq \frac{q_H^2}{|F|} + 2Ad_{\varphi}^{ECDDH}$.

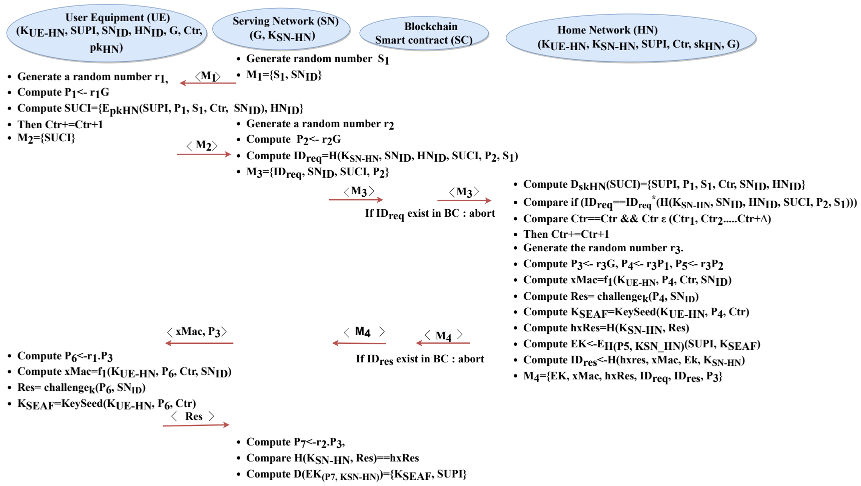

User Equipment (UE) ($K_{UE\text{-}HN}$, SUPI, SN_{ID}, HN_{ID}, G, Ctr, pk_{HN})
Serving Network (SN) (G, $K_{SN\text{-}HN}$)
Blockchain Smart contract (SC)
Home Network (HN) ($K_{UE\text{-}HN}$, $K_{SN\text{-}HN}$, SUPI, Ctr, sk_{HN}, G)
Generate random number S_1
M_1={S_1, SN_{ID}}
⟨M_1⟩
Generate a random number r_1,
Compute P_1<- r_1G
Compute SUCI={E_{pkHN}(SUPI, P_1, S_1, Ctr, SN_{ID}), HN_{ID}}
Then Ctr+=Ctr+1
M_2={SUCI}
⟨M_2⟩
Generate a random number r_2
Compute P_2<- r_2G
Compute ID_{req}=H($K_{SN\text{-}HN}$, SN_{ID}, HN_{ID}, SUCI, P_2, S_1)
M_3={ID_{req}, SN_{ID}, SUCI, P_2}
⟨M_3⟩
⟨M_3⟩
If ID_{req} exist in BC : abort
Compute D_{skHN}(SUCI)={SUPI, P_1, S_1, Ctr, SN_{ID}, HN_{ID}}
Compare if (ID_{req}==ID_{req}^{*}(H($K_{SN\text{-}HN}$, SN_{ID}, HN_{ID}, SUCI, P_2, S_1)))
Compare Ctr==Ctr && Ctr ε (Ctr_1, Ctr_2.....Ctr+Δ)
Then Ctr+=Ctr+1
Generate the random number r_3.
Compute P_3<- r_3G, P_4<- r_3P_1, P_5<- r_3P_2
Compute xMac=f_1($K_{UE\text{-}HN}$, P_4, Ctr, SN_{ID})
Compute Res= $challenge_k$(P_4, SN_{ID})
Compute K_{SEAF}=KeySeed($K_{UE\text{-}HN}$, P_4, Ctr)
Compute hxRes=H($K_{SN\text{-}HN}$, Res)
Compute EK<-$E_{H(P5,\ KSN\_HN)}$(SUPI, K_{SEAF})
Compute ID_{res}<-H(hxres, xMac, Ek, $K_{SN\text{-}HN}$)
M_4={EK, xMac, hxRes, ID_{req}, ID_{res}, P_3}
⟨ M_4 ⟩
⟨ M_4 ⟩
If ID_{res} exist in BC : abort
⟨ xMac, P_3⟩
Compute P_6<-$r_1.P_3$
Compute xMac=f_1($K_{UE\text{-}HN}$, P_6, Ctr, SN_{ID})
Res= $challenge_k$(P_6, SN_{ID})
K_{SEAF}=KeySeed($K_{UE\text{-}HN}$, P_6, Ctr)
⟨ Res ⟩
Compute P_7<-$r_2.P_3$,
Compare H($K_{SN\text{-}HN}$, Res)==hxRes
Compute D($EK_{(P7,\ KSN\text{-}HN)}$)={$K_{SEAF}$, SUPI}

Fig. 2: Proposed Protocol for mutual authentication

The terms q_H, $|F|$ and Ad^{ECDDH} represent the hash query, range space for hash function H(.) and $\varphi's$ advantage to get the $ECDDH$ problem (see Section II-D), respectively.

Proof: In order to demonstrate the security of the session key, we do the proof of the proposed protocol, same as [18]. For that, we use the three games names as G_i, where $i \in$ [0, 2] and an event $S_{\varphi G_i}$ which is defined as "in the game G_i, φ is able to properly predict the random bit c, and $Pr[S_{\varphi G_i}]$ describes this ability as a competitive advantage."

Game(G_0)**:** This game is executed to launch the real time attack on the basis of the randomly chosen C at the beginning. Therefore, It can be inferred from the semantic analysis

$$Ad_{\varphi} = [2Pr[S_{\varphi G_0}] - 1] \tag{1}$$

Game (G_1)**:** This game is formulated by the φ to intercept the exchanged message by running the *Execute* query. Afterwards, *Reveal* and *Test* queries are executed by the attacker in order to to ensure that SK generated between the UE and HN is real or random. Although, the SK is derived using the combination of random number and long secrets that are not known to the attacker. Therefore, it can be observed that intercepting the exchanged massage, will not let the attacker derive the session key, and the winging probability of G_0 and G_1 will be equivalent

$$Pr[S_{\varphi G_1}] = Pr[S_{\varphi G_0}] \tag{2}$$

Game(G_2)**:** This game is formulated by the attacker to execute the *HASH* query in order to model this game as active attack. Since all the messages are transmitted either in encrypted form or hashed from, capturing the intercepted message will not disclose any secret through which he can derive the SK. Apart from that, if an attacker tries to obtain the random number used in SK from P_1, P_2, P_3, he cannot determine it because of the $ECDDH$ problem (see Section II-D). Therefore, to generate the SK, the attacker needs P_6 or P_4 which is impossible for the attacker to get in polynomial time and there is no collision when the $HASH$ query is executed. So, it can be observed that the wining probability of G_2 will be equivalent to G_1 except for the hash collision. The following conclusion may be reached by combining the birthday paradox with the intractability of $ECDDH$.

$$Pr[S_{\varphi G_1}] - Pr[S_{\varphi G_2}] \leq Ad_{\varphi}^{ECDDH} + \frac{(q_H^2)}{2F} \tag{3}$$

Since every game has been played, φ must guess the precise value of bit c. In light of this,

$$Pr[S_{\varphi G_2}] = \frac{1}{2} \tag{4}$$

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

$$\begin{aligned} Ad_{\varphi} &= |2Pr[S_{\varphi G_0}] - 1| \\ &= Pr[S_{\varphi G_1}] - Pr[S_{\varphi G_2}] \end{aligned} \tag{5}$$

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

$$Ad_{\varphi} \leq 2Ad_{\varphi}^{ECDDH} + \frac{q_H^2}{F}$$

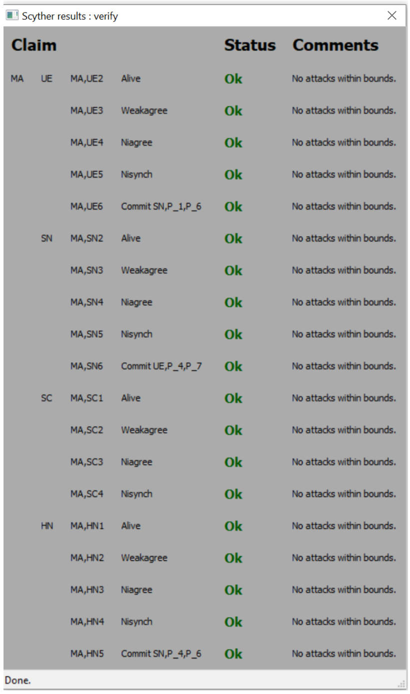
Thus, this demonstration shows that an attacker will not be able to ascertain the session key in polynomial time.

B. Security verification using Scyther tool

The security of the proposed protocol is investigated using the formal validation model Scyther tool [19]. It uses the Security Protocol Description Language (.spdl) to model the protocol. There are four types of properties defined that must be followed by the protocol (i.e., Alive, Weakagree, Niagree and Nisynch). The Scyther tool includes the various threat models such as Delvo Yao, ck adversary and eck threat model to verify the security. The outcome is indicated in Fig. 3 clearly shows that the proposed protocol passes all the security claims, indicating that there are no attacks and the proposed protocol is safe.

V. Performance measurement

To illustrate the effectiveness of the proposed protocol, this section compares the security features and does experimental analysis in order to calculate the costs in terms of computational



```
Scyther results : verify

Claim                                  Status   Comments

MA   UE   MA,UE2   Alive                 Ok     No attacks within bounds.
          MA,UE3   Weakagree             Ok     No attacks within bounds.
          MA,UE4   Niagree               Ok     No attacks within bounds.
          MA,UE5   Nisynch               Ok     No attacks within bounds.
          MA,UE6   Commit SN,P_1,P_6     Ok     No attacks within bounds.
     SN   MA,SN2   Alive                 Ok     No attacks within bounds.
          MA,SN3   Weakagree             Ok     No attacks within bounds.
          MA,SN4   Niagree               Ok     No attacks within bounds.
          MA,SN5   Nisynch               Ok     No attacks within bounds.
          MA,SN6   Commit UE,P_4,P_7     Ok     No attacks within bounds.
     SC   MA,SC1   Alive                 Ok     No attacks within bounds.
          MA,SC2   Weakagree             Ok     No attacks within bounds.
          MA,SC3   Niagree               Ok     No attacks within bounds.
          MA,SC4   Nisynch               Ok     No attacks within bounds.
     HN   MA,HN1   Alive                 Ok     No attacks within bounds.
          MA,HN2   Weakagree             Ok     No attacks within bounds.
          MA,HN3   Niagree               Ok     No attacks within bounds.
          MA,HN4   Nisynch               Ok     No attacks within bounds.
          MA,HN5   Commit SN,P_4,P_6     Ok     No attacks within bounds.
Done.
```

Fig. 3: Scyther tool result for proposed protocol

and communication cost. In addition to this, we also compute the cost for smart contract functions in terms of transaction and execution cost.

A. Security analysis

This section does the comparison of security features of the proposed protocol with [1], and [11]. The findings of Table III indicate that the proposed protocol offers all security features while the existing authentication protocols fail. It is worth noting that the proposed protocol uses the blockchain technology and compared to the protocol [11] that also relies on blockchain, the proposed protocol offers additional security features such as perfect forward secrecy and protection against impersonation attack.

TABLE III: Comparison of Security features of authentication protocols/ mutual authentication (MA), resistant to privacy attack (RPA), perfect forward secrecy (PFS), resistant to device stolen attack (RDSA), resistant to traceable attack (RTA), resistant to de-synchronization attack (RDA), resistant to malicious SN problem (RMSNP), resistant to replay attack (RRA), DoS attack prevention (DoSP), resistant to impersonation attack (RIA), Yes-$\surd$, No-$\times$.

Protocol	MA	RPA	PFS	RDSA	RTA	RDA	RMSN	RRA	DoSP	RPI
[1]	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\times$	$\times$
[11]	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
Ours	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

B. Test-bed implementation using MIRACL

This section describes the experimental analysis performed using the MIRACLE library [20]. MIRACL is known as c++ based programming software library to compute the cost of the cryptographic operations used in the protocols [16]. The experiments are performed on two different platforms such as Raspberry PI 4 as UE and desktop as HN.

TABLE IV: Simulations results in each platforms

Cryptographic operations	T_H	T_{PM}	T_{RSA}	T_{AES}
Desktop (ms)	0.0032	0.295	4.69	.0036
Raspberry PI 4 (ms)	0.0315	1.23	8.14	0.041

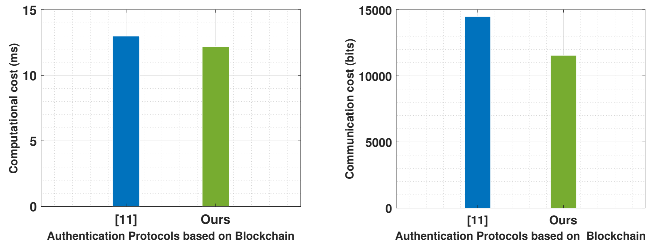
For the HN, the specification of the desktop is, Intel (R) Core(TM) i7-3770 with 3.40 GHz clock, 8 GB RAM running Linux Ubuntu 18.04.6 LTS, and the Raspberry as UE with the configurations: Raspberry Pi (Model: 4B, CPU: ARM® Cortex®-A7, Cores: 4, and RAM: 8GB). We run the cryptographic primitive 100 times and compute the average run time based on highest and lowest run time. The terms T_H, T_{PM}, T_{AES}, T_{RSA} describe the execution time required for "one-way hash function (SHA-256), elliptic curve multiplications, (AES-128) encryption/decryption, and (RSA-2048) encryption/decryption ," respectively.

C. Computation & Communication cost analysis

This section illustrates the cost of the cryptographic operations utilised in the proposed protocol as well as the amount of bits transmitted during the authentication session between the communicating entities. The computational cost of the proposed prototype is estimated using the cryptographic costs obtained through the experimental analysis displayed in Table IV. The proposed protocol takes $(1T_{RSA} + 3T_H + 2T_{PM})$ cryptographic operations at UE side and $(1T_{AES} + 9T_H + 5T_{PM})$ operations at HN side; which is $\approx$ 12.20 ms. Whereas [11] requires $1T_{RSA} + 4T_H$ operations at UE side and $1T_{RSA} + 9T_H + 2T_{AES}$ requires at HN side which is $\approx$ 12.99 ms. We use the size of the cryptographic operation recommended by NIST [21] to calculate the number of bits sent during the message exchange. The proposed protocol contains the eight message exchanges: $(\langle S_1, SN_{ID}\rangle$, $\langle SUCI\rangle$, $\langle ID_{req}, SN_{ID}, SUCI, P_2\rangle$, $\langle ID_{req}, SN_{ID}, SUCI, P_2\rangle$, $\langle EK, xMac, hxRes, ID_{req}, ID_{res}, P_3\rangle$, $\langle EK, xMac, hxRes, ID_{req}, ID_{res}, P_3\rangle$, $\langle xMac, , P_3\rangle, \langle Res\rangle)$, which takes 11552 bits for communication. In contrast, [11] contains eight messages: $(\langle R_1, ID_{SN}\rangle$, $\langle SUCI\rangle$, $\langle req_{id}, ID_{SN}, SUCI\rangle$, $\langle req_{id}, ID_{SN}, SUCI\rangle$, $\langle EK, xMac, hxRes, req_{id}, reqsid, HN_R\rangle$, $\langle EK, xMac, hxRes, req_{id}, reqsid, HN_R\rangle$, $\langle xMac, , HN_R\rangle, \langle Res\rangle))$ which takes 14496 bits for communication. The comparison of computational and communication costs of the proposed protocol and its counterparts is displayed in Fig. 4a and Fig. 4b, which indicates that the proposed protocol takes less computational as well as communication cost compared to its counterparts. The key reason for this is that the proposed protocol utilises ECC to send messages from the HN to the SN, whereas [11] uses RSA. It is also noted that ECC is less expensive than RSA because the ECC_{256} bit offers the same amount of security as RSA_{2048} bit [16] [18].

D. Performance measurements for smart contract functions

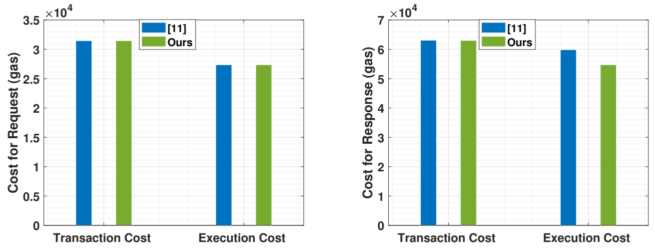
This section computes the execution time needed for the smart contract function in terms of transaction and



(a) (b)

Fig. 4: Comparison of (a) computational and (b) communication cost of authentication protocols.

execution time. The operations of functions of the 4^{th} step (i.e., $ID_{req}, SN_{ID}, SUCI, P_2$) and 5^{th} step (i.e., $EK, xMac, hxRes, ID_{req}, ID_{res}, P_3$) are executed by the smart contract. We implement the proposed protocol and its competitor using the Solidity code, which is an object orient high level language for implementing the smart contract. These transactions are measured in gas. All the implementations are performed on desktop having the configuration, Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz 3.60 GHz clock, 16 GB RAM running on Window 10 pro. In order to compute the cost, we consider the size of cryptographic operations as used in [11]. The comparison outcome of Request message (4^{th}) shown in Fig 5a indicates that proposed protocol and [11] have the same transaction and execution cost. It is due to the fact that the size of first message of proposed and [11] are same. The comparison outcome of Response message (i.e., 5^{th}) shown in Fig 5b indicates that proposed protocol has less transaction and execution cost as compared to [11]. It is due to the fact that the size of the message sent from HN to SN of the proposed protocol is less then the size of [11].



(a) (b)

Fig. 5: Comparison of (a) request and (b) response of transaction and execution cost of authentication protocols.

VI. Conclusion

This paper proposes an authentication mechanism for roaming scenarios. The scheme employed the blockchain to provide a secure channel for exchanging messages between the HN and SN in order to protect the HN from the malicious SN. The security features of the proposed protocol are verified using the ROR logic and Scyther tool. Furthermore, we compute the computational and communication cost to show that the proposed protocol is less costly compared to its counterparts. In addition, we compute the cost for response and request for the smart contract in terms of transaction and execution time, which shows that the proposed protocol is less than its counterparts. Therefore, it can be observed that the proposed protocol outperforms its counterparts in terms of security and performance.

References

[1] " 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/.

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

[3] A. Braeken, "Symmetric key based 5g aka authentication protocol satisfying anonymity and unlinkability," *Computer Networks*, vol. 181, p. 107424, 2020.

[4] J. Munilla, M. Burmester, and R. Barco, "An enhanced symmetric-key based 5g-aka protocol," *Computer Networks*, p. 108373, 2021.

[5] H. Choudhury, "Hashxor: A lightweight scheme for identity privacy of iot devices in 5g mobile network," *Computer Networks*, vol. 186, p. 107753, 2021.

[6] Y. Liu, L. Huo, and G. Zhou, "Tr-aka: A two-phased, registered authentication and key agreement protocol for 5g mobile networks," *IET Information Security*, 2021.

[7] A. Braeken, M. Liyanage, P. Kumar, and J. Murphy, "Novel 5g authentication protocol to improve the resistance against active attacks and malicious serving networks," *IEEE Access*, vol. 7, pp. 64 040–64 052, 2019.

[8] Y. Wang, Z. Zhang, and Y. Xie, "Privacy-preserving and standard-compatible {AKA} protocol for 5g," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

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

[10] Y. Xiao and Y. Wu, "5g-ipaka: An improved primary authentication and key agreement protocol for 5g networks," *Information*, vol. 13, no. 3, p. 125, 2022.

[11] M. Hojjati, A. Shafieinejad, and H. Yanikomeroglu, "A blockchain-based authentication and key agreement (aka) protocol for 5g networks," *IEEE Access*, vol. 8, pp. 216 461–216 476, 2020.

[12] Z. Gao, D. Zhang, J. Zhang, Z. Liu, H. Liu, and M. Zhao, "Bc-aka: Blockchain based asymmetric authentication and key agreement protocol for distributed 5g core network," *China Communications*, vol. 19, no. 6, pp. 66–76, 2022.

[13] 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.

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

[15] P. Hegedűs, "Towards analyzing the complexity landscape of solidity based ethereum smart contracts," *Technologies*, vol. 7, no. 1, p. 6, 2019.

[16] S. Son, J. Lee, Y. Park, Y. Park, and A. K. Das, "Design of blockchain-based lightweight v2i handover authentication protocol for vanet," *IEEE Transactions on Network Science and Engineering*, 2022.

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

[18] A. K. Das, M. Wazid, A. R. Yannam, J. J. Rodrigues, and Y. Park, "Provably secure ecc-based device access control and key agreement protocol for iot environment," *IEEE Access*, vol. 7, pp. 55 382–55 397, 2019.

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

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

[21] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A new mutual authentication and key agreement protocol for mobile client—server environment," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1275–1286, 2021.