5G-RNAKA : A Random Number-based Authentication and Key Agreement Protocol for 5G Systems

Hui Li
Beijing University of Posts and Telecommunications
Beijing, China
lihuill@bupt.edu.cn

Haotian Li
Beijing University of Posts and Telecommunications
Beijing, China
lihaotian@bupt.edu.cn

Chi Ma
Beijing University of Posts and Telecommunications
Beijing, China
machi@bupt.edu.cn

Jingjing Guan
Beijing University of Posts and Telecommunications
Beijing, China
guaner@bupt.edu.cn

Junchi Zeng
Beijing University of Posts and Telecommunications
Beijing, China
zengjunchi@bupt.edu.cn

Haonan Feng
Beijing University of Posts and Telecommunications
Beijing, China
fenghaonan222@gmail.com

Ziming Zhao
Northeastern University
Boston, Massachusetts, USA
z.zhao@northeastern.edu

Abstract

The 5G-AKA protocol, defined by 3GPP for authentication and key agreement in 5G networks, remains vulnerable to linkability, synchronization failure, and Sequence Number (*SQN*) exposure attacks. These issues threaten user privacy and service availability. Existing improvements often retain these flaws or cause high overhead due to continued use of the legacy *SQN* mechanism from 3G.

In this paper, we propose 5G-RNAKA, a secure and efficient AKA protocol for 5G systems. Unlike 5G-AKA, 5G-RNAKA eliminates *SQN* counters and instead utilizes random numbers generated by the Universal Subscriber Identity Module (USIM) in 5G User Equipment (UE) for session identification. This random number is embedded in the reply message from the service network (SN) to prevent replay attacks against the UE. Additionally, by removing the *SQN* mechanism, 5G-RNAKA enhances user privacy by preventing attackers from linking challenge-response sessions. It also enables the UE to authenticate the SN, effectively mitigating the risk of SN impersonation.

We formally verify that 5G-RNAKA achieves its security goals of privacy, authentication, and secrecy using the state-of-the-art formal verification tool, Tamarin Prover. Our implementation and evaluation further demonstrate that 5G-RNAKA improves communication efficiency and reduces storage overhead. While primarily designed for 5G, 5G-RNAKA's features align with emerging trends in 6G authentication, suggesting its potential for adaptation to future 6G architectures.

CCS '25, Taipei

ACM ISBN 979-8-4007-1525-9/2025/10
https://doi.org/10.1145/3719027.3744844

CCS Concepts

• **Security and privacy → Security protocols**; **Mobile and wireless security**.

Keywords

5G, Protocol, Authentication and key agreement, 5G-AKA, Formal analysis

ACM Reference Format:
Hui Li, Haotian Li, Chi Ma, Jingjing Guan, Junchi Zeng, Haonan Feng, and Ziming Zhao. 2025. 5G-RNAKA : A Random Number-based Authentication and Key Agreement Protocol for 5G Systems. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), October 13–17, 2025, Taipei.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3719027.3744844

1 Introduction

Authentication and Key Agreement (AKA) protocols are crucial mechanisms for achieving mutual authentication between subscribers and their carriers, establishing secure channels to protect subsequent communications. Standardized by the 3GPP [3, 4, 9], AKA protocols have been the foundation of mobile communication security since their inception in the 3G mobile communication system, integrated into both the Universal Subscriber Identity Module (USIM) card and the mobile core network globally.

In June 2018, the 3rd Generation Partnership Project (3GPP) released the final version (15.1.0) of Technical Specification (TS) 33.501 [2], defining a new version of AKA protocol, known as 5G-AKA, which enhances subscriber privacy protection and can resist the International Mobile Subscriber Identity (*IMSI*) catching attack [33, 60] by using randomized asymmetric encryption to protect subscribers' permanent identifiers. However, some previous

attacks remain effective against 5G-AKA [11, 19, 41] due to its inheritance of features from the AKA protocols in 3G and 4G, including a challenge-response procedure based on symmetric cryptography and the use of sequence number (*SQN*) counters.

The identified security and privacy issues in 5G-AKA are mainly related to the *SQN* counters that were introduced in 3G to prevent replay attacks and can be classified into three categories: (i) The AKA linkability attack [11, 14, 37, 44] exploits distinguishable failure messages caused by either *SQN* or *MAC* checking failures in the User Equipment (UE), enabling an attacker to determine whether a target UE is present in a specific area. This vulnerability also enables SUCI-catching attacks [26] that exploit similar failure patterns to identify specific subscribers, posing a serious privacy threat. (ii) Synchronization failure attack [41], in which the attacker uses the incremental nature of *SQN* to make the *SQN* values between the UE and the Home Network (HN) inconsistent, causing the target UE fails to verify the *SQN* of the HN and resulting in service disruptions; and (iii) *SQN* exposure attack [19], in which attackers can exploit *SQN* hidden vulnerabilities to obtain partial least significant bits of *SQN*, allowing them to understand the location or typical service usage of subscribers.

To date, numerous enhanced protocols have been proposed to mitigate the aforementioned attacks. These efforts include encrypting messages over the wireless channel with asymmetric algorithms [46], enhancing *SQN* concealment mechanisms [19], and encrypting challenges generated by HN [61], among others. However, most efforts have focused on enhancing *S QN* protection or reducing the possibility of UE being distinguished by *SQN*, without considering abandoning the *SQN* counters entirely. This often leads to more complex mechanisms that may be impractical or leave the methods still vulnerable and lacking in security.

Beyond SQN-related vulnerabilities, the *SQN* counters also require implementing complex logic on both the UE and the HN to ensure the correct generation, verification, and synchronization of *SQN* [44]. Within the AKA framework, the UE and the HN maintain an *SQN* that increments with every authentication attempt. Ideally, the *SQN* values in these two entities are consistent. However, various issues such as network disruptions, message loss, or replay attacks can lead to inconsistency. When this occurs, the UE fails the freshness verification of *SQN*, demanding a re-synchronization procedure to harmonize the *SQN* values and restart the authentication procedure. The security flaws and implementation complexity caused by *SQN* highlight the necessity of exploring alternative methods for enhancing security in AKA protocols without reliance on *SQN* counters.

There are two possible methods, timestamps, and random numbers, that serve the same function as *SQN* counters. However, timestamps rely on highly synchronized clocks between the sender and receiver, which is difficult to implement in mobile communication systems for practical reasons. As for random numbers, during the 3G and 4G eras, USIMs were incapable of generating secure random numbers [14, 42]. But USIMs in the 5G era are capable of generating random numbers [5] and perform randomized asymmetric encryption, as required to compute SUbscription Concealed Identifier (*SUCI*) from SUbscription Permanent Identifier (*SUPI*) in 5G-AKA [6]. Notably, the USIM's pseudo-random number generator (PRNG) must comply with ANSI X9.82 or NIST SP 800-90A [9, 22]. This provides a foundation for using the freshness of random numbers to ensure session uniqueness and prevent replay attacks, fulfilling a role similar to that of *SQN*.

Based on the findings, we propose a novel AKA protocol for 5G, namely 5G-RNAKA , which uses a trusted USIM to generate a random number N and sends it along with its identity to the Serving Network (SN) during the initiation phase. Since N is regenerated each time the protocol runs, it serves as a fresh value per session, replacing the *SQN* counters and providing defense against replay attacks. In addition, 5G-RNAKA eliminates synchronization between the UE and the HN, thus not susceptible to synchronization failure attacks. Moreover, since the protocol does not require *SQN* and only has one type of authentication failure message, attackers cannot distinguish UEs based on different failure messages. This makes it immune to the AKA linkability attack and the *SQN* exposure attack, thereby better protecting user privacy.

However, the design of 5G-RNAKA still faces other challenges: (i) It should enable the UE to authenticate the SN, which prevents attackers from impersonating the SN to the UE and ensures both parties negotiate the same anchor key K_{SEAF}; and (ii) It should be efficient and low-cost due to frequent use in 5G networks [44]. Therefore, 5G-RNAKA should use at most the same number of messages as 5G-AKA for authentication. The UE may use asymmetric encryption functions to encrypt the *SUPI*, while only using symmetric keyed one-way functions in other processes, the SN should only use one-way functions, and the HN may use the above-mentioned functions and asymmetric decryption.

To address the first challenge, in 5G-RNAKA the SN generates a *MAC* (*SNMAC*) containing the identity of the SN for UE authentication. Since *SNMAC* also includes the random number N and confidential information from the HN via a secure channel, attackers cannot replay or forge it. To address the second challenge, 5G-RNAKA adheres to limitations on message quantity and cryptographic algorithms, satisfying requirements for efficiency and low cost.

In summary, our contributions extend beyond the replacement of *SQN*, as 5G-RNAKA systematically addresses core limitations of 5G-AKA in terms of security, protocol complexity, and deployability, as detailed below:

Replacement of SQN with USIM-Generated Randomness. 5G-RNAKA is the first practical authentication protocol to completely eliminate SQN counters, replacing them with UE-generated randomness. It remains compatible with key 5G procedures like SUCI-based identity protection and GUTI-based session continuity. This design renders the protocol stateless and resilient to SQN-related attacks, resulting in a cleaner and more robust authentication framework.

Mutual Authentication Including UE-to-SN Verification. In contrast to 5G-AKA, which lacks explicit UE verification of the SN, 5G-RNAKA ensures true mutual authentication. The *SNMAC* allows the UE to verify the SN's legitimacy, thereby preventing impersonation attacks and ensuring that both entities derive the same anchor key K_{SEAF}, strengthening key agreement integrity.

Formal Security and Privacy Guarantees. We conduct a formal symbolic analysis of 5G-RNAKA using the Tamarin Prover [13]. The results demonstrate that 5G-RNAKA achieves stronger authentication and privacy properties than 5G-AKA. In particular, the combination of UE-to-SN authentication and the elimination of

SQN makes the protocol resilient to SQN-related attacks and replay attacks.

Real-World Implementation and Performance Evaluation. We implement 5G-RNAKA in a practical over-the-air testbed using Universal Software Radio Peripheral (USRP) and the open-source projects Free5GC [38], srsUE [57], and srsRAN [58]. Experimental results show that 5G-RNAKA significantly reduces computational, storage, and communication overhead compared to 5G-AKA, confirming its suitability for lightweight mobile devices.

We open our codes for formal verification, experiments, and performance evaluation at https://github.com/TACSL/5G-RNAKA.

2 Background

In this section, we discuss how mutual authentication and key establishment are achieved in the 5G-AKA based on 3GPP TS 33.501 [9]. We also analyze known security and privacy issues in 5G-AKA, focusing on the role of *SQN* and related concerns. Table 1 lists the abbreviations and notations used in this paper.

Table 1: Abbreviations and Notations

Abbreviations	Meaning
SUPI	SUbscription Permanent Identifier
SUCI	SUbscription Concealed Identifier
GUTI	Globally Unique Temporary Identifier
K	The permanent key shared between a UE and HN
AK	Anonymity Key
K_{SEAF}	The anchor key derived from 5G-AKA
(PK_{HN}, SK_{HN})	The HN's public-private key pair
R	The random number generated by HN as a challenge
N	The random number generated by UE as a Nonce
ID_{SN}	Serving Network Identifier
ID_{HN}	Home Network Identifier
SQN_{UE}	The UE's sequence number
SQN_{HN}	The HN's sequence number

2.1 Architecture

We abstract the network nodes or functional components involved in 5G-AKA into three entities for clarity, as shown in Figure 1:

(1) The UE refers to the subscriber's mobile device, such as smartphones and tablets. Each UE is uniquely identified by its *SUPI*, stored in the USIM, a cryptographic chip within the UE, replacing the "*IMSI*" used in the pre-5G standard.

(2) The HN is the network that the subscriber belongs to, where the Authentication credentials Repository and Processing Function (ARPF) processes data securely, while the Authentication Server Function (AUSF) handles authentication requests.

(3) The SN provides network access to UEs and offers services such as calling or messaging upon successful authentication via 5G-AKA. The gNodeB (gNB) in the SN handles wireless communication with the UE, while the SEcurity Anchor Function (SEAF) manages the keys for subsequent security communication.

Due to the inherent openness of wireless communication, messages between the UE and SN are exposed to risks on an insecure wireless channel. In contrast, the channel between the SN and HN is classified as secure, as specified in TS 33.501 [9], assuming it

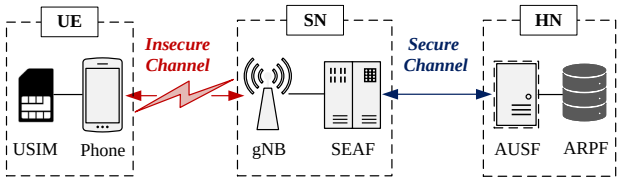

Figure 1: The architecture of 5G-AKA

is protected from eavesdropping and tampering. Communication within network components, such as between the USIM and the phone, gNB and SEAF, or AUSF and ARPF, is also considered secure. In the following discussions on protocol flows and formal modeling, the UE, SN, and HN will each be abstracted as single logical entities, without considering their internal components.

2.2 5G-AKA Protocol

To achieve mutual authentication between UEs and core networks and to establish a shared anchor key (K_{SEAF}), the 3GPP has standardized 5G-AKA as the primary AKA protocol in 5G. It consists of an initiation phase and a challenge-response phase.

The Initiation Phase. As shown in Figure 2, the UE initiates the protocol by identifying itself to the HN while preserving the privacy of the UE's identity so that it cannot be traced. This can be done in two different ways according to TS 24.501 [6]:

(1) Send its *SUCI* to the SN when the UE lacks a Globally Unique Temporary Identity (*GUTI*), such as during the initial registration with the SN. The *SUCI* is generated by encrypting the *SUPI* with randomized asymmetric encryption (aenc): $SUCI = \langle \text{aenc}(\langle SUPI, R_s \rangle, PK_{HN}), ID_{HN} \rangle$, where R_s is a random nonce and ID_{HN} uniquely identifies the HN. This encryption process not only conceals the UE's identity but also ensures that the *SUCI* is unique for each session, as R_s is randomly generated for each session, thereby preventing an adversary from linking different sessions. The ID_{HN} allows the SN to request authentication material from the appropriate HN. The SN then forwards the *SUCI* along with its identity to the HN.

In this context, R_s is a parameter whose exact meaning depends on the specific encryption algorithm used. For instance, in the Elliptic Curve Integrated Encryption Scheme (ECIES) as suggested in C.3 of TS 33.501 [9], R_s corresponds to the Initial Counter Block (*ICB*). The *ICB* is a value derived from the temporary shared key used in ECIES and functions as the initialization vector (IV) for the symmetric encryption process. Its role is to ensure that each encryption operation generates distinct ciphertexts (i.e., the *SUCI*) by introducing randomness, thereby preventing identical encryptions even if the same data is used.

(2) Send *GUTI* to the SN if the UE was assigned one. The SN uses *GUTI* to retrieve the UE's *SUPI* from its (*GUTI*, *SUPI*) database and sends it along with its identity to the HN. If retrieval fails, the SN then requests the *SUCI* from the UE (as shown by the dashed line in Figure 2). Using *GUTI* instead of always using *SUCI* avoids running asymmetric encryption, which saves computational time.

After the HN receives a message containing either the *SUCI* or the *SUPI*, it extracts the UE's *SUPI* or decrypts the *SUCI* using the

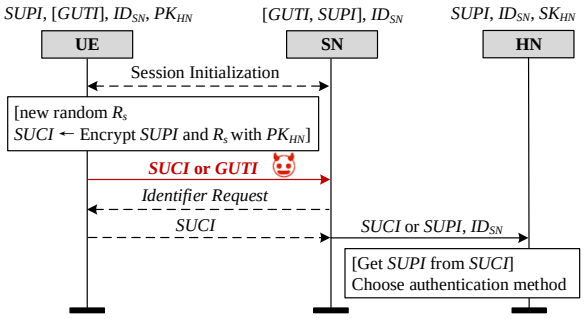

Figure 2: The initiation phase of 5G-AKA. [] indicates optional parameters or operations.

HN's private key to obtain the *SUPI*. The HN then determines the appropriate authentication protocol based on *SUPI* and retrieves the corresponding K and SQN_{HN} from its database.

The Challenge-response Phase. In this phase, the UE and core networks mutually authenticate and derive the anchor key K_{SEAF}, as shown in Figure 3. They employ a set of cryptographic algorithms including `f1`-`f5`, `f1*`, and `f5*` for computing authentication parameters based on independent one-way keyed functions, alongside two complex Key Derivation Functions (`KDFs`) named `Challenge` and `KeyGeneration` (`KG`), with $\oplus$ representing Exclusive-OR. Firstly, using these algorithms, the HN generates an authentication vector $AV = (R, AUTN, HxRES^*)$ and an anchor key K_{SEAF}. These parameters have the following roles:

- R is a random number used as a challenge value;
- $AUTN$ is an authentication token that concatenates a MAC with a concealed SQN_{HN}, where the concealment of SQN_{HN} is accomplished through an anonymous key AK. SQN_{HN} ensures the freshness of the challenge, and MAC is used for the authentication of the UE to the HN;
- $HxRES^*$ is the hash of the expected response value, used for the SN authentication to the UE.
- K_{SEAF} is the anchor key negotiated between the UE and core network, which is used to derive keys used for securing subsequent communication.

Then, upon receiving AV, the SN stores R and $HxRES^*$, and sends $(R, AUTN)$ to the UE. Next, the UE parses $AUTN$ as $(CONC, MAC)$, calculates AK using K and R, XORs AK with $CONC$ to obtain SQN_{HN}, and calculates $xMAC$ based on SQN_{HN} and R. Subsequently, the UE authenticates the HN by comparing the calculated $xMAC$ with the received MAC and checks the freshness of the challenge by comparing the received SQN_{HN} with its SQN_{UE}. There are three possible results (the first two indicating UE authentication failure and the last one indicating success):

- If $xMAC$ does not match MAC, it indicates that the UE fails to authenticate the HN. As a result, the UE replies 'MAC_Failure' to SN and the authentication is directly terminated (see Case 1 in Figure 3).
- If the check on SQN_{HN} fails, the UE sends 'Sync_Failure' and performs a re-synchronization with the HN to keep SQN consistent (see Case 2 in Figure 3).

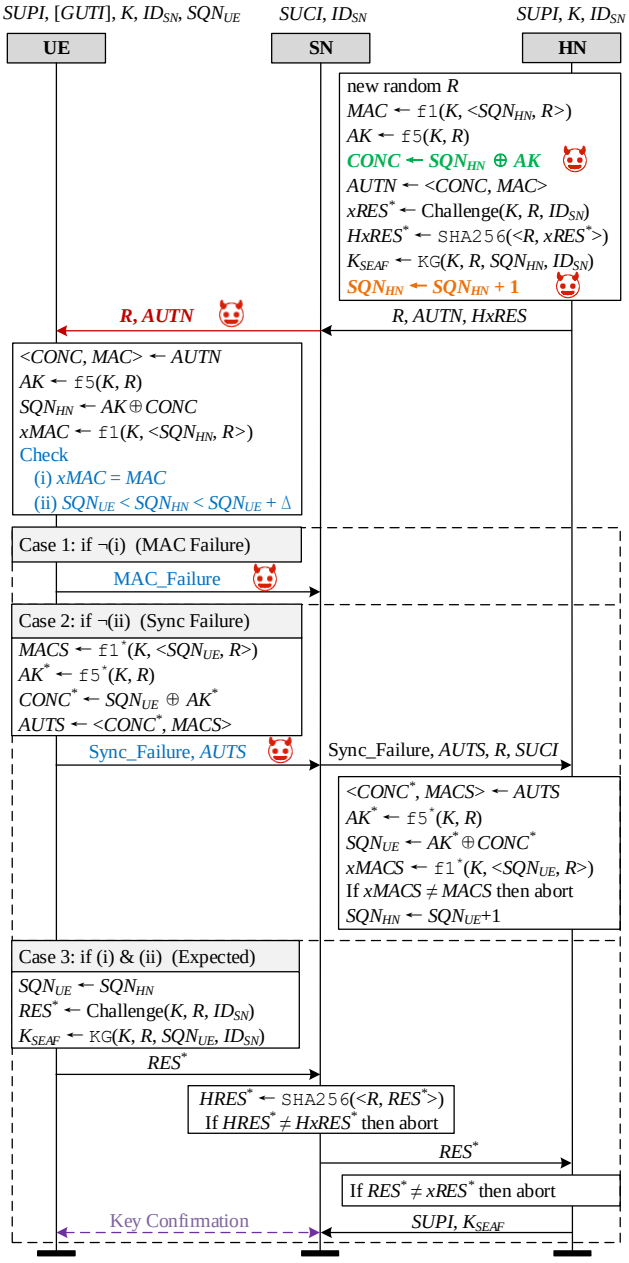

Figure 3: The challenge-response phase of 5G-AKA

- If both the MAC and SQN checks hold, indicating successful authentication on the UE side, the UE computes RES^* and K_{SEAF}, and subsequently returns RES^* to the SN (see Case 3 in Figure 3).

Upon successful authentication on the UE side (i.e., Case 3), the SN computes $HRES^*$ with RES^*, and checks if it matches the stored $HxRES^*$. A successful match indicates that the UE is authenticated by the SN, which proceeds to forward RES^* to the HN, who validates

it. If it succeeds, then the HN successfully authenticates the UE and sends *SUPI* and K_{SEAF} to the SN.

In the procedure of re-synchronization, the UE calculates *AUTS* and replies with ('Sync_Failure', *AUTS*), where *AUTS* contains a concealed SQN_{UE} ($SQN_{UE} \oplus AK$). *AUTS* is finally sent to the HN and undergoes authentication, confirming its origin from the intended UE by comparing the *MACS* value with one calculated using identical methods and parameters. Therefore, the SQN_{UE} is extracted and incremented to replace the SQN_{HN}.

Compared to EPS-AKA in 4G [4], 5G-AKA introduces two key improvements. First, it enhances privacy against passive attackers by encrypting *SUPI* into *SUCI* with a randomized public key cipher, which can prevent identity capture attacks on *IMSI* in 4G [33, 60]. Second, it strengthens authentication by verifying the complete response (RES^*) in the HN instead of the SN, thereby preventing malicious SNs from impersonating legitimate networks to UE.

2.3 Security Issues in 5G-AKA

5G-AKA has been identified to have vulnerabilities that can lead to the following three types of attacks:

AKA Linkability Attack [11, 14, 37, 44]. In carrying out this attack, the attacker records the (*R*, *AUTN*) message sent to the target UE and replays it to all UEs within the target area (see the red part of Figure 3). Upon receiving this replayed message, the target UE will reply with the 'Sync_failure' message because it has the correct *K* and it can pass the *MAC* check, but it cannot pass the *SQN* freshness check since the message is replayed. The non-target UEs will reply with the 'MAC_failure' message as they cannot pass the *MAC* check without the correct key *K* (see the blue part of Figure 3). This attack can further lead to a SUCI-catching attack. If an attacker replays a captured *SUCI* to trigger authentication (see the red part in Figure 2), the attacker can distinguish between the target UE and non-target UEs due to these differing responses, enabling the tracking of subscribers [26].

Synchronization Failure Attack [41]. This attack causes the UE to fail to verify the *SQN* of the network side, ultimately resulting in service interruptions for the victim UE. The AKA mechanism originally utilized the freshness of *SQN* to prevent attackers from replaying HN messages, and the SQN_{HN} and SQN_{UE} both increment by 1 after each authentication (see the orange part in Figure 3). But when the adversary constructs multiple legitimate initial messages to send to HN, the SQN_{HN} is increased multiple times, which is inconsistent with the *SQN* of the target UE. As a result, the target UE fails to verify the *SQN* of the HNs, resulting in service disruptions.

***SQN* Exposure Attack** [19]. This attack builds upon the AKA linkability attack by exploiting the concealment of *SQN* through $SQN \oplus AK$ in *AUTS* and the deterministic nature of *SQN*, which is incrementally updated. When the attacker replays the same (*R*, *AUTN*) to the target UE at different times, the UE returns different $AUTS_1$ and $AUTS_2$ due to the failure of the *SQN* freshness checks. In $AUTS_1$, $CONC_1^* = SQN_{UE1} \oplus AK^*$, and in $AUTS_2$, $CONC_2^* = SQN_{UE2} \oplus AK^*$(refer to the green part of Figure 3). Since $AUTS_1$ and $AUTS_2$ are based on the same random number *R*, their *AKs* are identical. Therefore, $CONC_1^* \oplus CONC_2^* = SQN_{UE1} \oplus SQN_{UE2}$. By replaying authentication challenges multiple times, the attacker can deduce partial least significant bits of the target UE's *SQN*, revealing the typical service usage of subscribers, such as the number of calls or SMSs sent over a period.

In the aforementioned attacks, the AKA linkability attack and the *SQN* exposure attack can compromise subscriber privacy, while the synchronization failure attack can impact service availability. These attacks exploit almost all related *SQN* processes, including the *SQN* encryption mechanisms, *SQN* incremental methods, and *SQN* validation results, to extract subscriber privacy information and even monitor subscriber behavior. Therefore, these attacks can be collectively referred to as SQN-related attacks.

2.4 Issues with Recently Proposed Protocols

We summarize the recently proposed protocols to mitigate SQN-related attacks in Table 2.

DH-based Proposal. Liu et al. [46] and Arkko et al. [12] introduced asymmetric algorithms such as the Diffie-Hellman (DH) key exchange algorithm into the 5G-AKA to resist the AKA linkability attack and the *SQN* exposure attack. However, this measure does not defend against the synchronization failure attack. Additionally, the adoption of DH increased the protocol's complexity [24].

Enhance the *SQN* Concealment. Borgaonkar et al. enhanced the *SQN* concealment mechanism in 5G-AKA to resist the *SQN* exposure attack by symmetrically encrypting SQN_{UE}, correctly randomizing *AUTS*, and asymmetrically encrypting SQN_{UE} [19]. However, these schemes cannot resist the synchronization failure attack and the AKA linkability attack caused by replaying *SUCI* [61].

Noval-AKA. Braeken et al. [21] proposed the Novel-AKA protocol as a well-motivated enhancement to 5G-AKA that addresses security weaknesses such as user traceability by replacing the *SQN* with random numbers. However, the protocol assumes that the SN cannot retrieve the *SUPI* via the *GUTI*, which may limit its compatibility with standard 5G flows. Furthermore, the requirement for the HN to encrypt both the *SUPI* and K_{SEAF}, combined with the absence of explicit RES^* confirmation by the HN, may introduce additional processing complexity and hinder the establishment of injective agreement between the UE and HN.

Symmetric Key-based. Braeken proposed a lightweight two-phase variant of 5G-AKA [20] to improve efficiency. Distinct from 5G-AKA, it requires two extra identity-related parameters during the registration phase between the UE and HN. Utilizing symmetric keys, XOR for obfuscation, and hash functions for authentication, it effectively resists the synchronization failure and the *SQN* exposure attacks. However, it remains vulnerable to AKA linkability attack due to differing message formats for UEs in synchronized and desynchronized states [51].

5G-AKA'. Wang et al. proposed the 5G-AKA' [61], in which a temporary shared key established from the key encapsulation mechanism inherent in the asymmetric encryption algorithm `ECIES` is used to encrypt the challenge sent by the HN to address the privacy issues. However, the generation of this temporary key requires the use of *SUCI* as the identity of the UE in each authentication, contradicting the 5G-AKA process described in Section 2.2, which prefers to use *GUTI* over *SUCI*, thus avoiding running high-computational algorithms such as asymmetric encryption to keep efficiency. Additionally, since attackers can still cause SQN_{HN} to increment continuously by replaying multiple legitimate initial messages, it makes

Table 2: Related improved protocols for 5G-AKA. "✓/ ✘" indicates whether the method can resist the attack or possesses the property (e.g., compatibility with *GUTI* and *SUCI*). "High" means the method uses asymmetric cryptography during the challenge-response phase, while "Low" means it does not.

	Year	AKA linkability attack	Synchronization failure attack	*SQN* exposure attack	Compatibility with *GUTI* and *SUCI*	Computational cost
DH-based proposal [12, 46]	2018	✓	✘	✓	✓	High
Enhance the *SQN* concealment [19]	2019	✘	✘	✓	✓	High
Noval-AKA [21]	2019	✓	✓	✓	✘	High
Symmetric key-based [20]	2020	✘	✓	✓	✓	Low
5G-AKA' [61]	2021	✓	✘	✓	✘	Low
EIMSI-AKA [36]	2023	✘	✘	✓	✓	High
5G-RNAKA (This work)	2025	✓	✓	✓	✓	Low

the UE fail to verify the *SQN* of HN, thus causing the protocol vulnerable to synchronization failure attack.

EIMSI-AKA. Fei et al. proposed the EIMSI-AKA protocol [36], which uses asymmetric encryption and digital signatures to encrypt authentication failure messages, resulting in higher computation costs for this protocol. Moreover, although this protocol can resist the *SQN* exposure attack, it cannot defend against the AKA linkability attack caused by replaying Encrypted *IMSI* (*EIMSI*) and the synchronization failure attack.

Yu et al. proposed the AAKA scheme [65], which protects against mobile tracking by honest-but-curious mobile network operators using anonymous credentials. It prevents linkability across sessions while still allowing lawful de-anonymization when necessary. Although AAKA aligns with the 3GPP architecture and SIM standards, it requires major changes to AMF and USIM to support advanced primitives (e.g., zero-knowledge proofs, Boneh-Boyen signatures, and Decisional Diffie-Hellman), resulting in significantly higher overhead (52.92 ms vs. 3.69 ms in 5G-AKA). Due to its distinct trust model and infrastructure requirements, AAKA is not included in Table 2, which focuses on SQN-related proposals within conventional 5G-AKA assumptions.

In short, the currently proposed enhanced protocols either cannot fully support the various scenarios of 5G, require significant modifications to the existing 5G architecture, or still fail to address security and privacy issues in 5G-AKA. 5G-RNAKA resolves all of the above issues, and that is why we continue to compare the proposed 5G-RNAKA with 5G-AKA in the following analysis.

3 THREAT MODEL AND SECURITY GOALS

In this section, we first describe the threat model and then list the security goals. The threat model and security goals align with those proposed by Basin et al. [14], with references to the work of Wang et al. [61] and Cremers et al. [29], and meet the requirements of the 3GPP standards [9].

3.1 Threat Model

Security Assumptions on Channels. Security assumptions about channels depend on their guaranteed communication security. The channel between the UE and SN is insecure and active attackers are realistic threats by acceptable cost [39, 56]. Thus, the channel allows to carry out eavesdropping by passive attackers and manipulation, interception, and injection of messages by active attackers. This channel is considered unbound. The channel between the HN and SN is defined by TS 33.501 [9] as a secure channel for "end-to-end core network interconnection" to provide secrecy, integrity, authenticity, and replay protection for message transmission. Thus, attackers cannot perform passive attacks or active attacks on this channel. Additionally, each message is bound to a unique session ID within these channels [14].

Security Assumptions on Cryptographic Primitives. We assume the cryptographic algorithms are secure, which means the message authentication or key derivation functions implemented by `f1-f5`, `f1*`, `f5*`, and *KDFs* can preserve the confidentiality and integrity of their inputs [14, 61].

Security Assumptions on Entities. Based on TS 33.501 [9], it is assumed that attackers do not control entities within the 5G core network (such as the HN). Furthermore, long-term secrets (e.g., K, SK_{HN}) and temporary secrets (e.g., K_{SEAF}) within these entities are expected to remain confidential and not be leaked to the attacker. However, attackers can potentially lure the UE to connect to their fake base stations through the fake base station attack [40, 49]. In addition, it is assumed that the USIM in the honest UE is trusted, so attackers cannot access their protected permanent key K and anchor key K_{SEAF}, and can only obtain the keys stored within the UE under their control.

3.2 Security Goals

We outline the authentication, secrecy, and privacy goals based on proposals by Basin et al. [14] and Cremers et al. [29], aligning with the 5G standard's security goals for 5G-AKA [9]. The formal expressions of security goals are indicated in purple text.

Authentication Properties. To formalize 5G-RNAKA 's authentication properties, we use Lowe's taxonomy [47] to define goals linked to security definitions in Tamarin Prover [54]. This taxonomy outlines four authentication levels from A's perspective: Aliveness (ensuring B has run the protocol), weak agreement (confirming B's prior interaction with A), non-injective agreement (achieving data consistency between both parties), and injective agreement (ensuring a one-to-one relationship between A and B).

Authentication between UE and HN. Firstly, the UE/HN must obtain weak agreement with the HN/UE (A1/A2). The weak agreement here is equivalent to the non-injective agreement of entity identity identifiers. Then, to ensure that the SN is authorized by the HN, the UE must obtain non-injective agreement on ID_{SN} with the

HN (A3). To prevent the SN from faking authentication requests with the HN for UE not attached to its base stations, the HN must obtain non-injective agreement on ID_{SN} with the UE (A4). Finally, to ensure that both parties have the same K_{SEAF} and that the same K_{SEAF} is not established twice, the UE/HN must obtain injective agreement on K_{SEAF} with the HN/UE (A5/A6).

Authentication between UE and SN. Firstly, the UE/SN must obtain weak agreement with the SN/UE (A7/A8). Additionally, for the same reasons as before, the UE/SN must obtain injective agreement on K_{SEAF} with the SN/UE (A9/A10).

Authentication between SN and HN. Firstly, to ensure that the UE authenticated by the SN is authorized by the HN, the SN must obtain non-injective agreement on *SUPI* with the HN (A11). Since the HN does not require authorization from the SN when authenticating the UE, there is no need to establish the non-injective agreement in the reverse direction. Additionally, for similar reasons, the SN/HN must obtain injective agreement on K_{SEAF} with the HN/SN (A12/A13).

Secrecy Properties. Firstly, if the permanent key K is compromised, attackers can impersonate an honest UE to communicate with other entities, therefore the secrecy of K must be ensured (S1). Similarly, if K_{SEAF} is compromised, attackers can derive other sub-keys through K_{SEAF}, allowing them to decrypt all subsequent communications of the honest UE, therefore the secrecy of K_{SEAF} must be ensured (S2). Note that S2 differs from forward secrecy and post-compromise secrecy [27], which require session key K_{SEAF} secrecy even when long-term key material K is compromised. Since K is stored and used in trusted components (the USIM in the UE and the AUSF in the HN), forward and post-compromise secrecy of K_{SEAF} are not considered as security goals.

Privacy Properties. According to TS 33.102 and TR 33.899 [1, 5], 3G already had security requirements for subscriber identity confidentiality, anonymity, and unlinkability. Therefore, the *SUPI* used as an identifier of UE in the protocol flow, must be kept secrecy (P1). Moreover, the proposed AKA protocol shall provide unlinkability for the subscriber to prevent the leakage of their location information and identification by attackers. According to the definition of indistinguishability [23, 25, 61, 66], unlinkability can be expressed as UE indistinguishability that given two honest UEs, an active attacker cannot distinguish which UE it is interacting with (P2).

In the correspondence between security goals and attacks outlined in Section 2.3, the AKA linkability attack and the *SQN* exposure attack correspond to security goal *P2*, as they are related to the attacker's ability to distinguish between target and non-target UEs. Meanwhile, the synchronization failure attack is associated with the value of *SQN* in the protocol. Since the proposed improved protocol eliminates the *SQN* counters, security goals related to *SQN* are not within the scope of this paper, essentially eliminating the existence of the synchronization failure attack.

In addition, the prevention of bidding down attacks is a fundamental security requirement defined by 3GPP for the entire security context establishment between UE and the core network. According to TS 33.501 [9], these attacks are mitigated by introducing the Anti-Bidding down Between Architectures (ABBA) parameter during the procedures of security algorithm selection, key establishment, and the security mode command procedure, including authentication and key agreement. Therefore, the new AKA protocol adopts this approach instead of considering additional measures.

4 5G-RNAKA

In this section, we introduce the proposed 5G-RNAKA to enhance 5G-AKA security without using *SQN*. It resists all SQN-related attacks and provides security features such as unlinkability, mutual authentication, and secrecy. We first explain the reasons for removing *SQN* and the feasibility of our approach in the design concept part. Then, we describe the complete message flow of 5G-RNAKA .

4.1 Design Overview

In 5G-AKA, the *SQN* counters are used to provide the replay protection for the UE, but as analyzed in Section 2.3, the processes related to *SQN* have been found to have security vulnerabilities and can be exploited by attackers to compromise user privacy.

Firstly, due to the use of the *SQN* mechanism, the UE needs to check the freshness of the *SQN*, resulting in different responses from the UE to authentication challenge messages (R, $AUTN$), i.e. 'MAC_Failure' and 'Sync_Failure', which can be distinguished by attackers, thereby causing AKA linkability attack. Secondly, because the HN performs an incremental operation on the SQN_{HN} stored in HN during the authentication vector preparation stage, attackers can construct a legitimate initial message, causing the SQN_{HN} to be inconsistent with the *SQN* of the target UE, which also leads to synchronization failure attack. Finally, due to the use of XOR in hiding *SQN* within *AUTS*, attackers can easily obtain the XOR values of multiple *SQNs* at different times by repeatedly sending the same initial message, while SQN_{HN} lacks randomness since the HN increments SQN_{HN} by 1, leading to the *SQN* exposure attack.

Based on research into relevant standards, the choice of *SQN* to provide replay protection for the subscriber is for historical reasons, i.e. the USIMs in 3G and 4G lacked the ability to generate random numbers [14]. With the development of technology, this issue no longer exists. According to TS 31.102 [5], the USIMs in 5G should support the randomized cryptographic algorithm for encrypting *SUPI*. To maintain the freshness and randomness of *SUCI* [26], the 5G USIMs now have the ability to generate random numbers.

Therefore, 5G-RNAKA resists SQN-related attacks by replacing the *SQN* in 5G-AKA with a random nonce, denoted as N, which can be generated by the USIM in the UE and sent to the SN. The SN can then calculate an *SNMAC* with N, the SN's identity identifier ID_{SN}, and the secret information *HNMAC* obtained from the HN. *SNMAC* is used by the UE to authenticate the SN's identity and verify the freshness of challenge messages.

SNMAC is a critical component of 5G-RNAKA that distinguishes it from 5G-AKA. Below is an explanation of its computation. Firstly, the use of the random nonce N serves to resist replay attacks, playing a similar role to *SQN*. This is because N is regenerated for each authentication process, and discarded by the UE after authentication, effectively identifying the current authentication. Thus, when the UE verifies *SNMAC*, it is actually verifying the freshness of N. Secondly, the use of *HNMAC* prevents attackers from forging *SNMAC*, as *HNMAC* is concealed over the wireless channel through AK, making it a secret information for attackers. Lastly, the use of ID_{SN} enables the UE to authenticate the SN. Through analysis of Section 2.2 of 5G-AKA, it can be found that 5G-AKA only implements authentication of HN and freshness checking of challenges on the UE side, lacking authentication of SN. This is

achieved in 5G-RNAKA through UE checking of *SNMAC*, which contains the identity identifier $I\ D_{SN}$ for SN.

Furthermore, due to the removal of the *SQN* counters in the proposed 5G-RNAKA , it can resist SQN-related attacks as outlined in Section 2.3, with detailed explanations provided in Section 5.4. And because the protocol does not necessitate synchronization between the UE and the HN, the re-synchronization process is also eliminated, thereby simplifying the protocol's complexity.

4.2 Protocol Flow

We now describe the protocol flow in the initiation phase and the challenge-response phase of 5G-RNAKA:

The Initiation Phase. The SN triggers this phase to send the UE's identity to the HN while transferring the *N* generated by UE to SN, as shown in Figure 4. Similar to the initialization process of 5G-AKA, the UE will proceed in two different ways.

(1) If the UE possesses a *GUTI*, it will generate a new random number *N* and send <*GUTI*, *N*> to the SN.

(2) If a *GUTI* is not available for the UE, the UE will use `aenc` to encrypt its *SUPI*. More specifically, the UE first uses `ECIES` to freshly generate a public-private key pair <*PK*, *sk*>. Second, it computes k_{UE} = `KDF`($sk \cdot PK_{HN}$). Third, it parses k_{UE} as $EK \parallel ICB \parallel MK$, computes *C1* = `ENC`(*EK*, *SUPI*, *ICB*), C2 = `MAC`(*MK*, *C1*), where `ENC` is the symmetric encryption algorithm and `MAC` is the function used to calculate the MAC tag value. Finally, it constructs the *SUCI* = {*PK*, *C1*, *C2*}. In this case, *ICB* can be used as *N*, rather than generating an additional random number, thus simplifying the process and improving efficiency. The UE then sends <*SUCI*, *N*> to the SN.

After the SN receives <*GUTI*/*SUCI*, *N*>, it first stores *N* for the challenge-response phase. Then, if it receives *SUCI*, *SUCI* and ID_{SN} are sent to the HN; if it receives an valid *GUTI*, ID_{SN} and *SUPI* corresponding to *GUTI* are sent to the HN; if it receives a invalid *GUTI*, the *SUCI* request will be sent to the UE, which will then follow step 2 of the above process.

Subsequently, if *SUCI* is received, the HN employs its private key SK_{HN} to decrypt *SUCI* and retrieve *SUPI*. More specifically, the HN first computes k_{HN}=`KDF`($PK \cdot sk_{HN}$). Then, it parses k_{HN} as $EK \parallel ICB \parallel MK$, computes `MAC`(*MK*,*C1*), and verifies whether it matches *C2*; If the MAC check passes, it derives *SUPI* by computing `DEC`(*EK*, *C1*), otherwise it aborts, where `DEC` denotes the symmetric decryption algorithm corresponding to `ENC`. Finally, based on *SUPI*, the HN selects the appropriate authentication protocol and retrieves the permanent key *K* from its database.

The Challenge-response Phase. In this phase, the different entities participating in the protocol authenticate each other via a challenge-response procedure and derive the anchor key K_{SEAF} as depicted in Figure 5. The main process of this phase is as follows:

First, the HN generates authentication vector *AV* = (*R*, *HNMAC*, *AUTN*, $HxRES^*$) and K_{SEAF} using the permanent key *K* retrieved during the initiation phase, where the cryptographic algorithms `f1-f5`, `Challenge`, and `KG` used are the same as those in 5G-AKA:

- *R*: a random number as the challenge from HN.
- *HNMAC*: a *MAC* of *R* and *K* using `f1`, utilized for verifying the identity of HN.
- *AUTN*: a concealed *HNMAC* using an anonymous key *AK* derived from *K* and *R*.

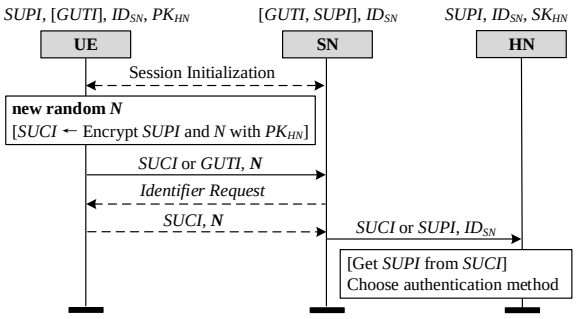

Figure 4: The initiation phase of 5G-RNAKA . Bold indicates the differences from 5G-AKA. [] indicates optional parameters or operations.

- $HxRES^*$: a hash value of the expected response, obtained by hashing *R* cascaded with $xRES^*$, where $xRES^*$ is the expected response computed with *K*, *R*, and ID_{SN}.

Then, the HN sends the generated *AV* to the SN. Upon receiving *AV*, the SN stores $HxRES^*$ and calculates *SNMAC* using `SHA256` over *HNMAC*, *N*, and ID_{SN}. Next, the SN sends (*R*, *AUTN*, *SNMAC*) to the UE. The UE completes authentication of the SN and the HN, and generates RES^* through the following computation:

- Obtain *HNMAC* from *AUTN* by computing $AUTN \oplus AK$, where *AK* is derived from *K* and *R* using `f5`.
- Compute *xHNMAC* using *K* and *R*.
- Calculate *xSNMAC* using *HNMAC*, *N*, and ID_{SN}.

Next, the UE checks whether *xHNMAC* matches *HNMAC* and *xSNMAC* matches *SNMAC*. The former indicates that the UE has successfully authenticated the HN, while the latter indicates that the UE has successfully authenticated the SN and completed the freshness check for the challenge. If either of these checks fails, UE will reply with a 'MAC_Failure' message to the SN. Once the UE successfully completes authentication of the SN and the HN, it calculates RES^*, derives the anchor key K_{SEAF} from `KG`(*K*, *R*, ID_{SN}), and sends RES^* to the SN.

Finally, the SN calculates $HRES^*$ by hashing *R* concatenated with RES^*, and if $HRES^*$ matches $HxRES^*$, the SN successfully authenticates the UE and sends RES^* to the HN. Upon receipt of RES^*, If RES^* matches $xRES^*$, which means the HN completes the authentication of the UE, then the HN sends the UE's *SUPI* and K_{SEAF} to the SN.

Compared with 5G-AKA, 5G-RNAKA differs in three ways: First, both the UE and the HN no longer utilize the *SQN* counter for freshness checks. Instead, freshness checks in 5G-RNAKA are accomplished using a nonce *N*, which is fresh and generated in the UE. Second, since *SNMAC* includes the SN's identifier ID_{SN}, the UE can authenticate the SN by checking *SNMAC* generated by the SN, which relies on the freshness of *N* and the secret information *HNMAC* sent by the HN. Third, because 5G-RNAKA does not employ the *SQN* counters, it can eliminate the requirement of the re-synchronization phase (Case 2) in 5G-AKA, making the protocol more concise and more secure.

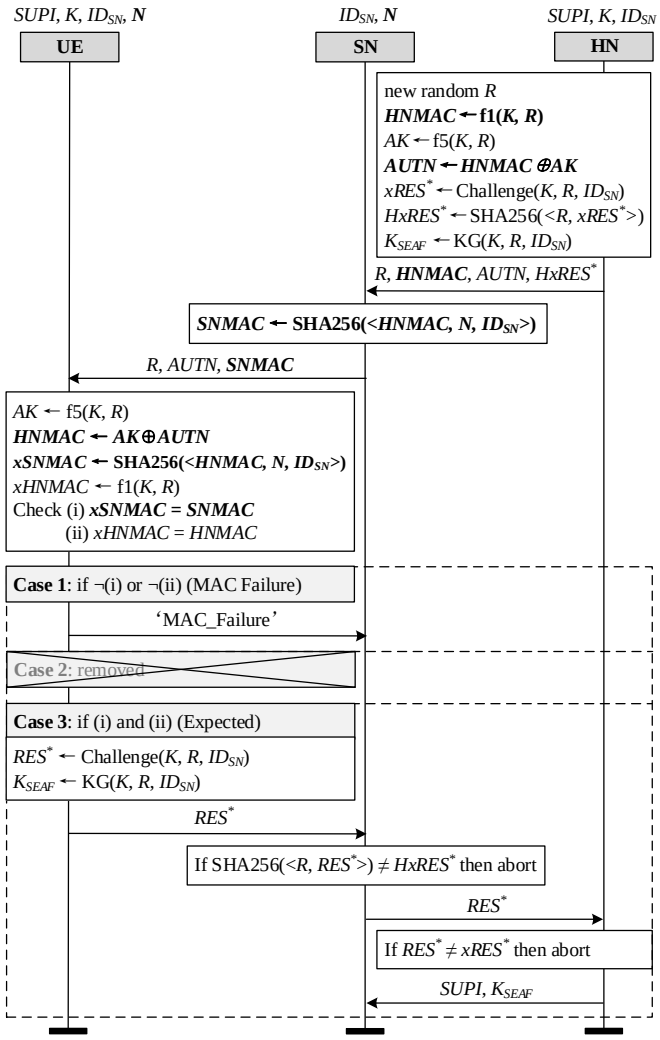

Figure 5: The challenge-response phase of 5G-RNAKA . Bold indicates the differences from 5G-AKA.

5 Security Analysis

We prove the security of 5G-RNAKA using the state-of-the-art formal analysis tool Tamarin Prover [13], which has been employed to verify many critical protocols, such as TLS 1.3 [30, 31], WPA2 [32], SPDM1.2 [28], SOAP [45] and the credit card protocol EMV [16, 17]. Notably, Basin et al. utilized Tamarin to conduct a detailed analysis of 5G-AKA [14], allowing us to draw from their abstraction and modeling methods for protocol entities, security goals, protocol sequences, and compromise scenarios as well as to facilitate comparing the analysis results of 5G-RNAKA with those of 5G-AKA. Therefore, in this section, we first introduce Tamarin, then describe the major modeling choices for 5G-RNAKA in Tamarin, next present the formal verification results, and finally explain how 5G-RNAKA mitigates related attacks.

5.1 Tamarin Prover

Tamarin is a symbolic protocol verifier specifically designed for analyzing the security goals of cryptographic protocols. Unlike computational model approaches requiring manual proof construction and significant computational resources [18], Tamarin provides a flexible framework supporting both fully automated verification and guided proof construction [13]. This dual-mode capability enables users to tackle a wide range of protocol complexities. Tamarin precisely models state transitions, cryptographic operations, and protocol properties using rules, equational theories, and lemmas [48]. By supporting the analysis of protocols with an unbounded number of sessions and simulating strong adversaries, it ensures scalability and rigorous vulnerability assessment for real-world applications. Furthermore, Tamarin's support for equivalence properties [15] and advanced cryptographic primitives, such as XOR [34, 35], makes it well-suited for addressing intricate privacy and authentication requirements of 5G-RNAKA.

5.2 Modeling Choices in Tamarin

We construct a comprehensive formal model of 5G-RNAKA, incorporating an additional key confirmation process. The model consists of approximately 500 Lines of Code (LoC), with 40 lemmas for verifying authentication and secrecy, considering both passive and active attackers for verifying privacy. The following section details the modeling of UE indistinguishability, key confirmation, compromise scenarios, and proof strategies.

Modeling the Indistinguishability of the UE. We verify whether an attacker can distinguish between two instances of the UE using observational equivalence, as described in [15] and implemented by Tamarin. This approach models the protocol using `diff-terms`, which take two arguments, essentially defining two instances of the protocol that only differ in some terms. Tamarin then compares the two resulting systems and verifies that the attacker cannot distinguish between them during any protocol execution, regardless of the adversarial actions.

To specifically verify this indistinguishability, we check whether an attacker can tell `UE1` apart from `UE2` after interacting with one of them (e.g., `UE1`) and recording the exchanged messages. We model this scenario using two systems: the Left Hand System (LHS), which includes two `UE1` instances, and the Right Hand System (RHS), which includes `UE1` and `UE2`. Both systems represent the execution of the 5G-RNAKA protocol.

The evaluation considers both passive and active attack models: (i) in the passive attacker model, the UE, SN, and HN are merged into a single entity, and messages exchanged between the UE and SN are exposed to the attacker for eavesdropping only; (ii) in the active attacker model, the SN and HN are combined into one entity, and the communication channel between the UE and SN uses the Dolev-Yao attacker model provided by Tamarin, which allows the attacker to manipulate, intercept, and inject messages. Additionally, our model also accounts for cases of *SNMAC* failure and *HNMAC* failure, which are key steps in the protocol and critical for proving the indistinguishability of the UE.

After verifying the correctness of the model through executable lemmas, we use Tamarin to automatically analyze indistinguishability under both attack models. If Tamarin detects that the attacker

can distinguish between LHS and RHS, it implies a potential vulnerability. Such a scenario could enable the attacker to exploit the protocol for linkability attacks in practice.

Modeling Key Confirmation. To illustrate the security of 5G-RNAKA with key confirmation, we model the process as follows: after receiving ($SUPI$, K_{SEAF}) from the HN, the SN computes $\mathtt{f1}(K_{SEAF}$, "SEAF") and sends it to the UE. The UE verifies it using its own K_{SEAF} and replies with $\mathtt{f1}(K_{SEAF}$, "UE"), which the SN verifies to complete the confirmation.

Modeling Compromised Scenarios. To obtain the minimal assumptions required to achieve each security goal, we also model various compromised scenarios, including: (i) the attacker controls multiple compromised UEs, gaining access to secret values such as $SUPI$ and K; (ii) the attacker controls multiple compromised SNs, enabling them to manipulate the channel between SNs and HNs; (iii) the attacker controls multiple compromised HNs, gaining access to sensitive values such as K, SK_{HN} and $SUPI$.

Proof Strategies. The model inclusion of various compromise scenarios, multiple parameters, XOR operation, and unbounded parallel entities execution leads to a state-space explosion and non-termination in Tamarin, which we address using customized proof strategies. These strategies are implemented using tactics, a feature introduced in Tamarin version 1.8.0, which enables guided proof searches. We developed 23 custom tactics, totaling over 600 LoC, to efficiently guide Tamarin's search process.

5.3 Verification Results

We conduct a systematic analysis of 5G-RNAKA using Tamarin, driven by the formalized model described above. The minimal assumptions required for achieving authentication and secrecy properties are shown in Table 3. To clearly demonstrate the security improvements of 5G-RNAKA over 5G-AKA, we present the comparison results for each security goal outlined in Section 3.2 in Table 4, noting that the results for 5G-AKA are referenced from Basin et al [14]. Therefore, we first present the results of 5G-RNAKA from the perspectives of privacy, authentication, and secrecy, and then provide a detailed analysis of the reasons why 5G-RNAKA satisfies three security goals that 5G-AKA fails to fully achieve.

Privacy. When proving privacy, we first prove the secrecy of $SUPI$ from each party by writing the code for the secrecy lemma, which shows that an attacker cannot obtain the identity identifier of the honest UE. Next, we prove the indistinguishability of the UE through the method given in Section 5.2. We find that either passive attackers or active attackers cannot find an attack path to distinguish between LHS and RHS, indicating that the attacker cannot threaten the privacy of the subscriber by implementing the AKA linkability attack. Therefore, the above two proofs demonstrate that 5G-RNAKA achieves its privacy goals.

Authentication. When proving the security goals of 5G-RNAKA, we refer to the lemmas provided by Basin et al. for 5G-AKA [14]. The proof results in Table 3 show that the 5G-RNAKA can achieve all authentication goals except for the UE obtain weak agreement with SN (A7), the UE obtain injective agreement on K_{SEAF} with SN (A9) and the HN obtain injective agreement on K_{SEAF} with the SN (A13). However, A7 and A9 can be satisfied if a key confirmation process is introduced.

Firstly, we explain why achieving goals of A7 and A9 requires key confirmation. For the protocol itself, after the UE sends RES^* and ends the protocol, the SN has not yet received the message containing the $SUPI$ from the HN. This makes it impossible to achieve mutual agreement on the identities from the UE's perspective, thus failing to meet the weak agreement of the UE with the SN (A7). Since A9 builds upon A7, it also cannot be satisfied. However, with the addition of key confirmation, a new interaction occurs between the UE and SN after the protocol execution, enabling the fulfillment of these two security goals.

Next, we discuss why A13 cannot be satisfied but is acceptable in practical systems. In 5G-RNAKA , as K_{SEAF} is sent to the SN by the HN in the last message, no evidence can be used to confirm that the SN has received K_{SEAF}. However, precisely because the channel between the HN and SN is secure, the violation of this target does not threaten the protocol's security. Meanwhile, sending $SUPI$ and K_{SEAF} together can eliminate the issue of the SN incorrectly assigning K_{SEAF} to the wrong UE. Therefore, this modification in 5G-RNAKA enhances security, and it is consistent with the current version of 5G-AKA (TS 33.501 version 18.5.0 [9]).

Secrecy. The results demonstrate that 5G-RNAKA can achieve all secrecy goals that we conduct a comprehensive verification from the perspectives of UE, SN, and HN. This means attackers cannot obtain the $SUPI$, K, K_{SEAF} from honest entities.

Comparison. Table 4 presents the results of security goals in the formal analysis of 5G-AKA [14] and 5G-RNAKA, along with the verification time for each security goal on a workstation (with a 2.10GHz Intel (R) Xeon (R) Gold 5218R CPU, 20cores, running Linux). Compared with the analysis results of 5G-AKA [14], 5G-RNAKA differs in three aspects:

(1) 5G-RNAKA achieves A3, which is not achieved by 5G-AKA. This is attributed to the generation of an $SNMAC$ on the SN, computed using cryptographic mechanisms that prevent forgery by potential adversaries. The UE verifies the SNMAC to authenticate the SN, thereby enhancing resistance to impersonation attacks.

(2) 5G-RNAKA achieves A5 without requiring key confirmation but 5G-AKA relies on it. In 5G-RNAKA, the UE verifies ID_{SN} that is embedded in K_{SEAF} using the $SNMAC$, ensuring consistency between the UE and HN. In contrast, 5G-AKA does not verify ID_{SN}, allowing attackers to tamper with it and causing inconsistencies in K_{SEAF} between the UE and HN. Consequently, 5G-AKA requires an additional key confirmation step.

(3) 5G-RNAKA achieves P2, but 5G-RNAKA cannot achieve. This is because 5G-RNAKA uses random number N instead of SQN mechanism to resist message replay attack in the wireless channel. This eliminates Case 2 from 5G-AKA, thereby preventing linkability attacks by active attackers and enhancing user privacy protection.

According to the formal analysis results presented in Table 3 and Table 4, as well as the explanations provided above, 5G-RNAKA offers advantages over 5G-AKA in both authentication and privacy.

5.4 Defending Against Known Attacks

Based on the formal analysis results provided in Section 5.3, we combine the protocol flow to demonstrate the ability of 5G-RNAKA to resist well-known classical attacks (as required by the 3GPP standard [9]) and SQN-related attacks as described in Section 2.3.

Table 3: Minimal assumptions required for 5G-RNAKA to achieve authentication, secrecy properties. We use "kc" indicates a key confirmation is needed, "$\neg K$" indicates honest UEs and honest HNs shared K is not revealed, "$\neg ch$" indicates an uncompromised channel between honest SN and honest HN, "$\neg ch^*$" indicates there are not compromised SNs, "$\neg SK_{HN}$" indicates honest HNs's SK_{HN} is not revealed, "wa" means that weakagreement already implies non-injectiveagreement, "-" indicates the property is violated by definition, and "✘" indicates the security goal cannot meet even when all entities are honest.

Point of View	UE		SN		HN	
Partner	SN	HN	UE	HN	UE	SN
Weakagreement	$kc \wedge \neg K \wedge \neg ch$	$\neg K$	$\neg K \wedge \neg ch$	$\neg ch$	$\neg K$	$\neg ch$
Injective agreement on K_{SEAF}	$kc \wedge \neg K \wedge \neg ch$	$\neg K \wedge \neg ch^*$	$\neg K \wedge \neg ch$	$\neg K \wedge \neg ch$	$\neg K$	✘
Non-injective agreement on ID_{SN}	wa	$\neg K \wedge \neg ch^*$	wa	wa	$\neg K$	wa
Non-injective agreement on *SUPI*	wa	wa	wa	$\neg ch$	wa	-
Secrecy on K_{SEAF}	$\neg K \wedge \neg ch$		$\neg K \wedge \neg ch$		$\neg K \wedge \neg ch$	
Secrecy on K	$\emptyset$		-		$\emptyset$	
Secrecy on *SUPI*	$\neg SK_{HN} \wedge \neg ch^*$		$\neg SK_{HN} \wedge \neg ch^*$		$\neg SK_{HN} \wedge \neg ch^*$	

Table 4: Authentication, secrecy, and privacy goals achieved by 5G-AKA and 5G-RNAKA . We use "I" for injective agreement and "NI" for non-injective agreement. "●" indicates that the protocol can meet the security goal. "◐" indicates that this security goal can only be satisfied with the inclusion of key confirmation. "○" indicates that the protocol cannot meet the security goal. The time required to analyze each goal is provided in [].

Security goals	5G-AKA	5G-RNAKA [time (s)]
A1: UE Weak agreement with HN	●	● [54.64]
A2: HN Weak agreement with UE	●	● [137.48]
A3: UE NI on ID_{SN} with HN	○	● [9.57]
A4: HN NI on ID_{SN} with UE	●	● [11.51]
A5: UE I on K_{SEAF} with HN	◐	● [112.35]
A6: HN I on K_{SEAF} with UE	●	● [9.90]
A7: UE Weak agreement with SN	◐	◐ [114.18]
A8: SN Weak agreement with UE	●	● [8.61]
A9: UE I on K_{SEAF} with SN	◐	◐ [34.33]
A10: SN I on K_{SEAF} with UE	●	● [37.97]
A11: SN NI on *SUPI* with HN	●	● [7.31]
A12: SN I on K_{SEAF} with HN	●	● [17.84]
A13: HN I on K_{SEAF} with SN	○	○ [20.50]
S1: Keep K secret	●	● [15.65]
S2: Keep K_{SEAF} secret	●	● [42.08]
P1: Keep *SUPI* secret	●	● [44.71]
P2: UE Indistinguishability	○	● [9465.56]

AKA Linkability Attack. Regarding the AKA linkability attack caused by replay (*R*, *AUTN*, *SNMAC*), there is only one failure message in 5G-RNAKA . Whether the target UE fails to check *SNMAC* or the non-target UEs fail to check *HNMAC*, they all return 'MAC_Failure' messages. Therefore, attackers cannot launch the AKA linkability attack based on different returned messages. According to Section 3.2, this attack corresponds to security goal P2.

Synchronization Failure Attack. Since 5G-RNAKA does not require synchronization between the UE and the HN, there is no need to maintain the same *SQN* between them. When attackers construct multiple legitimate initial messages sent to the HN, it does not result in desynchronization between the UE and the HN, thereby preventing the failure of UE's freshness verification of challenge messages during normal authentication processes. Hence, it does not result in the synchronization failure attack.

***SQN* Exposure Attack.** 5G-RNAKA replaces the *SQN* counters in 5G-AKA with a random number *N* generated by the UE, eliminating the scenario where attackers could obtain the XOR value of *SQN*. Furthermore, *N* is a random number and does not increment monotonically like *SQN*. Therefore, although attackers can easily obtain *N* transmitted in plaintext over the wireless channel, they cannot infer typical subscriber service usage from it. According to Section 3.2, this attack also corresponds to security goal P2.

Replay Attacks. In 5G-RNAKA, when an attacker replays a message (*R*, *AUTN*, *SNMAC*) sent by the SN to the UE, the value of *N* in the replayed message is outdated. As a result, the UE fails the freshness check when verifying *SNMAC*, preventing the replay attack. At the same time, 5G-RNAKA resists replay attacks against the core network using the same challenge (*R*) and response (RES^*) mechanisms as 5G-AKA. The authentication goals (A5, A6) can be used to examine whether such replay attacks are possible.

Man in the Middle Attacks and Impersonation Attacks. Due to the pre-stored permanent key *K* between the UE and the HN, attackers cannot forge relevant authentication messages over the wireless channel, and mutual authentication between entities is achieved. In addition, compared to 5G-AKA, 5G-RNAKA enables UE to authenticate SN through *SNMAC* that prevents attackers impersonate SN. All these aspects enable 5G-RNAKA to better resist man-in-the-middle attacks and impersonation attacks. From a formal analysis perspective, these two attacks are related to mutual authentication between UE and HN and can be checked through authentication goals (A3, A4, A5, A6).

DoS Attacks. According to the evaluation of computational costs in Section 6.3, 5G-RNAKA has lower overall computational costs compared to 5G-AKA. Moreover, on the most vulnerable HN to DoS attacks, 5G-RNAKA outperforms 5G-AKA. Therefore, 5G-RNAKA provides stronger resistance to DoS attacks than 5G-AKA.

SUCI-catching Attacks. 5G-RNAKA can resist SUCI-catching attacks by introducing a session-specific random number (N) securely bound to the authentication request via *SNMAC*. If a replayed *SUCI* contains an invalid or outdated N, all UEs, including the target UE, uniformly return a 'MAC_failure', thereby preventing the attacker from distinguishing the target. According to Section 3.2, this attack corresponds to security goal P2.

Key Reuse Attacks. According to TS 133.102 [3], Section 6.2.3, session keys must not be reused. In 5G, this session key refers to K_{SEAF}, and its uniqueness can be analyzed through injective agreement property between different parties, especially between the UE and the HN [14]. In 5G-AKA, the generation of K_{SEAF} relies on SN's identity ID_{SN}, which is not verified by the UE. This allows an attacker to tamper with ID_{SN}, potentially causing a mismatch in K_{SEAF} between the UE and the HN [14]. In contrast, 5G-RNAKA ensures that the UE verifies $I\ D_{SN}$via $SNMAC$ (A3), preventing tampering and ensuring consistency in K_{SEAF} derivation (A5).

Compromised but Initially Honest SN Attack. To further strengthen the robustness of 5G-RNAKA, we consider an edge-case scenario where a previously honest SN becomes malicious and reuses a valid tuple <*R*, *HNMAC*, *AUTN*, *HxRES**>. By forging a new *SNMAC*, the compromised SN can send <*R*, *AUTN*, *SNMAC*> to the UE, which may then authenticate and connect to it, since both *SNMAC* and *HNMAC* pass verification. To prevent this, 5G-RNAKA can be optionally extended to include the UE-generated nonce N in the *HNMAC* and K_{SEAF} computations, binding session keys to fresh values and preventing replay by compromised SNs.

6 Experimental Evaluation

In this section, we implement 5G-RNAKA on top of several open-source projects: Free5GC v3.4.1 [38], srsUE v23.11 [57], and srsRAN v24.10 [58] that together provide a complete 5G network stack and widely used in functional tests [10, 43, 55, 62]. Additionally, we use two USRPs [59] to simulate a wireless environment. SrsUE was modified to support UE in generating and transmitting N, as well as receiving and verifying *SNMAC* and *HNMAC*. Meanwhile, Free5GC was adapted to enable the HN and SN to calculate and transmit new *AV*, *HNMAC*, and *SNAMC*. We use these environments to evaluate the correctness of 5G-RNAKA. Finally, we evaluate and compare the performance of 5G-RNAKA and 5G-AKA based on computational, storage, communication, and end-to-end metrics [63, 64].

6.1 Implementation

In our experiment, as shown in Figure 6, we set up a laptop (3.8GHz AMD Ryzen 7 8845H CPU running Linux) to deploy the modified srsUE, simulating a 5G-capable UE. A desktop computer (a 2.8GHz Intel Core i9-10900 CPU running Linux) hosts the modified Free5GC and srsRAN. Free5GC implements the 5G core network defined in 3GPP Release 15 and beyond, while srsRAN provides a complete 5G RAN solution to serve as the core network of both SN and HN. Two USRP B210 devices act as the radio frequency frontends—one for the gNB to broadcast 5G signals, the other for the UE to transmit wireless messages.

We modify both srsUE and Free5GC to implement 5G-RNAKA . For srsUE, we modify: (i) the generation and transmission of the random number N were added; (ii) the reception of *SNMAC* was added; (iii) the calculation process for *xHNMAC* and *xSNMAC* was modified. For Free5GC, we modify: (i) the calculation of authentication vectors *AV* was modified; (ii) the transmission of *HNMAC* was added; (iii) the calculation and transmission of *SNMAC* was added. We remove the synchronization process and other operations related to *SQN* in both srsUE and Free5GC.

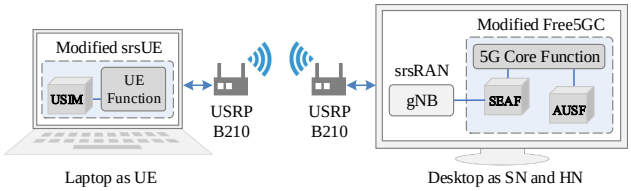

Figure 6: Experimental environment of 5G-RNAKA

6.2 Functional Testing

Our evaluation specifically examines the functional correctness of 5G-RNAKA, as its design is inherently decoupled from mobility management procedures (e.g., handover and cell reselection). For detailed analysis, refer to Section 7.

First, we test Case 1 in 5G-RNAKA. We load a UE with correct configuration data, including its K and *SUPI*, and ensure these values are properly stored in the HN. The UE initiates an authentication request, and the HN generates the corresponding *AV*. Then the UE computes the correct *RES**, matching the expected value in the HN, passes authentication, attaches to the gNB, and establishes secure communication channels with consistent K_{SEAF} generation.

Next, we test Case 3 in 5G-RNAKA. We load a UE with a correct *SUPI* but an incorrect K to trigger that scenario. Since the *HNMAC* or *SNMAC* calculated by the HN with a different key K, their verification can not pass in the UE. As a result, the attachment process is aborted, and no further network resources are allocated to the UE.

These tests demonstrate the correctness of the 5G-RNAKA under both normal and abnormal scenarios, confirming its ability to handle expected functional requirements.

6.3 Performance Evaluation

Computational Cost. We use Free5GC to evaluate the computational cost during the challenge-response phase of 5G-AKA and 5G-RNAKA , focusing on cryptographic execution time, excluding the initiation phase where both protocols have the same cost. The experiment was conducted on a MacBook Pro 2020 with an Apple M1 CPU, and results in Table 5 are averaged over 1,000,000 runs, measured in microseconds. In authentication failure (Case 1) and successful authentication (Case 3), 5G-RNAKA respectively shows 2.61% and 2.13% higher computational cost than 5G-AKA, mainly due to the additional *SNMAC* calculation. However, 5G-RNAKA eliminates the highest overhead of resynchronization (Case 2) in 5G-AKA. Therefore, the computational cost has been reduced by 39.99% overall in 5G-RNAKA.

Storage Cost. To evaluate the storage costs, we provide Table 6, which lists the protocol parameters and their sizes. We assume *SUPI* and *SUCI* each to be 128 bits, following Parne et al. [53], as they have variable lengths [7]. Additionally, *AUTN* size differs between

Table 5: Computational cost (μs).

	5G-AKA			5G-RNAKA		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
UE	0.8402	2.1832	2.3133	0.9346	-	2.3878
SN	0	0	0.0649	0.1045	-	0.1677
HN	2.3465	3.07	2.3109	2.231	-	2.2337
Total	3.1867	5.5532	4.6891	3.2701	-	4.7892

5G-AKA and 5G-RNAKA due to varying calculation methods. In the specific evaluation, we consider the long-term storage parameters of each entity before and after protocol operation, excluding those generated during execution, such as *MAC* and *AUTN*.

Table 6: Parameters and their sizes (bits) in performance evaluation.

Parameters	Size
SQN	48
MAC, *SNMAC*, *HNMAC*, *AUTN* (5G-RNAKA)	64
AUTS	112
K, *SUPI*, *SUCI*, *R*, *N*, RES^*, $HxRES^*$, *AUTN* (5G-AKA)	128
ID_{SN}, ID_{HN}, PK_{HN}, SK_{HN}, K_{SEAF}	256

Furthermore, because the SN and the HN only need to be stored once for variables such as ID_{SN} and ID_{HN}, we introduce the factor n to account for multiple UEs. Table 7 displays the calculated results. Since the storage cost is the same for any entity under different cases of the same protocol, there is no distinction between cases. The results indicate that 5G-RNAKA incurs lower storage costs on both the UE and the HN compared to 5G-AKA, due to the elimination of *SQN* in 5G-RNAKA, which requires long-term storage. Therefore, the storage cost has been reduced by 3.15% overall in 5G-RNAKA.

Table 7: Storage cost (bits).

	UE	SN	HN	Total
5G-AKA	1328n	384n+256	560n+512	2272n+768
5G-RNAKA	1280n	384n+256	512n+512	2176n+768

Communication Cost. We evaluate the communication cost of 5G-RNAKA in terms of both bandwidth and communication round based on Table 6, with the results shown in Table 8. Compared to 5G-AKA, 5G-RNAKA shows a slight increase in bandwidth under successful authentication (Case 3) but a reduction under authentication failure (Case 1). Both protocols require the same number of communication rounds for both success and failure. However, 5G-RNAKA eliminates the need for resynchronization (Case 2). Therefore, the communication cost has been reduced by 37.95% overall in 5G-RNAKA.

Table 8: Communication cost.

	5G-AKA			5G-RNAKA		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
bandwidth (bits)	1480	1888	1792	1280	-	1920
round	3	3	4	3	-	4

End-to-end Cost. To provide a more detailed real-world evaluation, we conducted end-to-end experimental testing on our platform as shown in Figure 6 based on 3GPP TS 23.502 [8], covering the full procedures of registration and authentication. We measured the end-to-end time of each phase, taking into account computational overhead, communication delays, and signal interference. All the time measured are the average of 20 runs and the time differences between the current and previous stages are shown in Table 9.

From the experimental results, 5G-RNAKA exhibits a slightly higher latency than 5G-AKA during the S2–S1 phase, which is due to the use of the `SHA256` algorithm in *xSNMAC* computation, compared to the XOR operation used in SQN_{HN} computation. However, this increase is minimal, only about 0.000008 seconds on average. Interestingly, the real-world evaluation shows that the overall latency of 5G-RNAKA is lower than 5G-AKA in both Case 1 and Case 3. And 5G-RNAKA eliminates Case 2, which had the longest latency. Overall, the end-to-end time of 5G-RNAKA is acceptable. Moreover, 5G-RNAKA successfully completes all protocol stages, demonstrating its compatibility with existing standards.

Table 9: End-to-end Cost (in seconds) from Registration to Security Mode Command. (S0: send registration request; S1: receive authentication request; S2: send authentication response; S3: receive authentication request in resynchronization; S4: send authentication response in resynchronization; S5: receive Security Mode Command)

Stage	5G-AKA			5G-RNAKA		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
S1-S0	0.482167	0.625720	0.572963	0.417251	-	0.470587
S2-S1	0.000023	0.000047	0.000032	0.000026	-	0.000045
S3-S2	-	0.060721	-	-	-	-
S4-S3	-	0.000022	-	-	-	-
S5-S4	-	0.058987	0.085316	-	-	0.048126
Total	0.48219	0.745497	0.658311	0.417277	-	0.518758

7 Discussions

In this section, we discuss the compatibility of 5G-RNAKA with the 5G architecture, impact on mobility, applicability to 6G, as well as limitations and future work.

Compatibility. The foundation of 5G-RNAKA lies in the USIM's capability to generate a random number *N*. According to TS 31.102 [5], this functionality is integrated into 5G USIMs. Given that the USIM is required to generate random numbers while encrypting *SUPI* to obtain the dynamic value of *SUCI* in the initialization of 5G-AKA. Furthermore, since 5G-RNAKA does not introduce any new entities or cryptographic algorithms compared to 5G-AKA, 3G/4G USIMs that can be upgraded to run 5G-AKA can also be upgraded to run 5G-RNAKA , which means that 5G-RNAKA has the same forward compatibility as 5G-AKA.

Impact of Mobility. 5G-RNAKA is designed to handle handover and cell reselection seamlessly. This is because such mobility events are managed by distinct components of the 5G architecture, such as the Access and Mobility Management Function (AMF) and the

Session Management Function (SMF), which are responsible for reliable context transfer, state synchronization, and resilience under dynamic network conditions, as specified in TS 33.501 [9].

First, 5G-RNAKA operates entirely at the Non-Access Stratum (NAS) layer and is independent of lower-layer RRC procedures like RRCConnectionReconfiguration, which govern physical-layer mobility. In practice, for example, srsUE uses a timer to ensure that if the UE does not receive R and $AUTN$ in time, it will resend $SUCI/GUTI$ and N; after five unresponsive resends, the UE clears the authentication data and re-authenticates.

Second, 5G-RNAKA is only triggered during specific signaling events such as initial registration or service requests, consistent with 5G-AKA. The modifications introduced in 5G-RNAKA, namely changing three authentication messages and removing the re-synchronization step, do not alter the conditions under which the authentication is triggered.

Moreover, 5G-RNAKA inherits the key management hierarchy of 5G-AKA. After successful authentication, the derived key K_{SEAF} is consistent with 5G-AKA and can be used in the same way to generate K_{AMF} and K_{gNB} during handovers.

These features ensure that AKA processes remain robust and independent of physical-layer dynamics, allowing us to focus on verifying the logical correctness of 5G-RNAKA in Section 6.2.

Applicability to 6G. Although the 6G architecture is still under development, emerging research has identified several new requirements for future authentication protocols, including stronger privacy guarantees, resistance to quantum attacks, and support for diverse access technologies [52]. 5G-RNAKA aligns well with these trends by removing the synchronization step, eliminating the dependency on sequence number management, and incorporating per-session randomness features that enhance unlinkability, which are critical for privacy-preserving authentication in 6G.

In addition, 5G-RNAKA adopts the design principle of decoupling the protocol logic from the underlying cryptographic algorithms. This separation enables seamless integration of future quantum-resistant cryptographic primitives without altering the core structure of the authentication procedure. Its compatibility with the existing 5G key hierarchy also facilitates long-term adaptability in evolving security environments.

Furthermore, 6G is expected to incorporate physical-layer key generation techniques, where the UE and the base station independently derive a shared symmetric key from the intrinsic randomness of the wireless channel [50]. Since this key is never transmitted and cannot be observed by an adversary, it inherently provides confidentiality and freshness. In future systems, such a physical-layer-derived key could be used directly as the session-specific random value N in 5G-RNAKA, further enhancing its efficiency and robustness in 6G deployments.

Limitations and Future Work. This paper acknowledges three limitations.

Firstly, implementing 5G-RNAKA requires modifications to certain computations and message exchanges among the UE, SN, and HN. To mitigate the impact on these entities, optimization is needed before deployment to balance performance, especially in large-scale or resource-constrained networks.

Secondly, the authentication goals A7 and A9 of 5G-RNAKA are achieved through key confirmation. This design choice implies that the overall security of the authentication process depends on subsequent procedures. Further research is needed to evaluate whether key confirmation should be incorporated into the AKA protocol to strengthen its security guarantees.

Finally, due to the difficulty of modifying the baseband testing protocol on a mobile phone, this study utilizes modified srsUE on a laptop for testing. Future work should involve testing 5G-RNAKA on actual mobile devices within real mobile communication testbeds to ensure compatibility and assess performance under various operating conditions.

8 Conclusion

In this paper, we propose 5G-RNAKA to enhance the security and privacy of 5G-AKA while maintaining its efficiency. This protocol replaces the SQN counters in the traditional AKA protocol with a challenge (N from the UE) and response ($SNMAC$ from the SN) solution. This modification not only mitigates SQN-related attacks but also simplifies the protocol by removing re-synchronization. Additionally, $SNMAC$ is computed by the SN with support from the HN, enabling the UE to authenticate the SN and prevent impersonation attacks. We demonstrate the correctness of 5G-RNAKA in a wireless environment, and performance evaluation results show that, compared to 5G-AKA, 5G-RNAKA enhances communication efficiency while reducing computational and storage costs. These advantages make 5G-RNAKA a promising solution for improving both security and operational performance in mobile networks, suggesting its potential suitability for future 6G environments.

Acknowledgment

The research of Beijing University of Posts and Telecommunications is supported by National Natural Science Foundation of China 62472045 and National Key Research and Development Program of China under Grant 2022YFB2902203.

References

[1] 3GPP. 2017. *Study on the security aspects of the next generation system.* TR 33.899. V1.3.0.
[2] 3GPP. 2018. *Security architecture and procedures for 5G System.* TS 33.501. V15.1.0.
[3] 3GPP. 2024. *3G security; Security architecture.* TS 33.102. V18.0.0.
[4] 3GPP. 2024. *3GPP System Architecture Evolution (SAE); Security architecture.* TS 33.401. V18.0.0.
[5] 3GPP. 2024. *Characteristics of the Universal Subscriber Identity Module (USIM) application.* TS 31.102. Version 18.4.0.
[6] 3GPP. 2024. *Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3.* TS 24.501. V18.6.0.
[7] 3GPP. 2024. *Numbering, addressing and identification.* TS 23.003. V18.5.0.
[8] 3GPP. 2024. *Procedures for the 5G System (5GS).* TS 23.502. V19.1.0.
[9] 3GPP. 2024. *Security architecture and procedures for 5G System.* TS 33.501. V18.5.0.
[10] Mujtahid Akon, Tianchang Yang, Yilu Dong, and Syed Rafiul Hussain. 2023. Formal analysis of access control mechanism of 5G core network. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security.* 666–680.
[11] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. 2012. New privacy issues in mobile telephony: fix and verification. In *Proceedings of the 2012 ACM conference on Computer and communications security.* 205–216.
[12] Jari Arkko, Karl Norrman, Mats Näslund, and Bengt Sahlin. 2015. A USIM compatible 5G AKA protocol with perfect forward secrecy. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 1205–1209.
[13] David Basin, Cas Cremers, Jannik Dreier, and Ralf Sasse. 2022. Tamarin: verification of large-scale, real-world, cryptographic protocols. *IEEE Security & Privacy* 20, 3 (2022), 24–32.
[14] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. 2018. A formal analysis of 5G authentication. In *Proceedings*

of the 2018 ACM SIGSAC conference on computer and communications security. 1383–1396.
[15] David Basin, Jannik Dreier, and Ralf Sasse. 2015. Automated symbolic proofs of observational equivalence. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1144–1155.
[16] David Basin, Ralf Sasse, and Jorge Toro-Pozo. 2021. Card brand mixup attack: bypassing the PIN in non-Visa cards by using them for visa transactions. In *30th USENIX Security Symposium (USENIX Security 21)*. 179–194.
[17] David Basin, Ralf Sasse, and Jorge Toro-Pozo. 2021. The EMV standard: Break, fix, verify. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1766–1781.
[18] Bruno Blanchet. 2012. Security protocol verification: Symbolic and computational models. In *International conference on principles of security and trust*. Springer, 3–29.
[19] Ravishankar Borgaonkar, Lucca Hirschi, Shinjo Park, and Altaf Shaik. 2019. New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols. *Proceedings on Privacy Enhancing Technologies* 3 (2019), 108–127.
[20] An Braeken. 2020. Symmetric key based 5G AKA authentication protocol satisfying anonymity and unlinkability. *Computer Networks* 181 (2020), 107424.
[21] An Braeken, Madhusanka Liyanage, Pardeep Kumar, and John Murphy. 2019. Novel 5G Authentication Protocol to Improve the Resistance Against Active Attacks and Malicious Serving Networks. *IEEE Access* 7 (2019), 64040–64052. doi:10.1109/ACCESS.2019.2914941
[22] Daniel RL Brown. 2009. Sec 1: Elliptic curve cryptography. *Certicom Research* 2 (2009).
[23] Jan Camenisch, Manu Drijvers, and Anja Lehmann. 2017. Anonymous attestation with subverted TPMs. In *Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part III 37*. Springer, 427–461.
[24] Jin Cao, Maode Ma, Hui Li, Ruhui Ma, Yunqing Sun, Pu Yu, and Lihui Xiong. 2019. A survey on security aspects for 3GPP 5G networks. *IEEE communications surveys & tutorials* 22, 1 (2019), 170–195.
[25] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. 2014. Algebraic MACs and keyed-verification anonymous credentials. In *Proceedings of the 2014 acm sigsac conference on computer and communications security*. 1205–1216.
[26] Merlin Chlosta, David Rupprecht, Christina Pöpper, and Thorsten Holz. 2021. 5G SUCI-Catchers: Still catching them all?. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 359–364.
[27] Katriel Cohn-Gordon, Cas Cremers, and Luke Garratt. 2016. On post-compromise security. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. IEEE, 164–178.
[28] Cas Cremers, Alexander Dax, and Aurora Naska. 2023. Formal Analysis of SPDM: Security Protocol and Data Model version 1.2. In *32nd USENIX Security Symposium (USENIX Security 23)*. 6611–6628.
[29] C. Cremers and M. Dehnel-Wild. 2019. Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion. In *Network and Distributed System Security Symposium*.
[30] Cas Cremers, Marko Horvat, Jonathan Hoyland, Sam Scott, and Thyla van der Merwe. 2017. A comprehensive symbolic analysis of TLS 1.3. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1773–1788.
[31] Cas Cremers, Marko Horvat, Sam Scott, and Thyla van der Merwe. 2016. Automated analysis and verification of TLS 1.3: 0-RTT, resumption and delayed authentication. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 470–485.
[32] Cas Cremers, Benjamin Kiesl, and Niklas Medinger. 2020. A Formal Analysis of IEEE 802.11's WPA2: Countering the Kracks Caused by Cracking the Counters. In *29th USENIX Security Symposium (USENIX Security 20)*. 1–17.
[33] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. 2014. IMSI-catch me if you can: IMSI-catcher-catchers. In *Proceedings of the 30th annual computer security applications Conference*. 246–255.
[34] Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, and Ralf Sasse. 2018. Automated unbounded verification of stateful cryptographic protocols with exclusive OR. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 359–373.
[35] Jannik Dreier, Lucca Hirschi, Saša Radomirović, and Ralf Sasse. 2020. Verification of stateful cryptographic protocols with exclusive OR. *Journal of Computer Security* 28, 1 (2020), 1–34.
[36] Teng Fei and Wenye Wang. 2023. The vulnerability and enhancement of AKA protocol for mobile authentication in LTE/5G networks. *Computer Networks* 228 (2023), 109685.
[37] Pierre-Alain Fouque, Cristina Onete, and Benjamin Richard. 2016. Achieving better privacy for the 3GPP AKA protocol. *Cryptology ePrint Archive* (2016).
[38] Free5GC. 2024. https://github.com/free5gc/free5gc.
[39] Nico Golde, Kévin Redon, and Jean-Pierre Seifert. 2013. Let me answer that for you: Exploiting broadcast information in cellular networks. In *22nd USENIX Security Symposium (USENIX Security 13)*. 33–48.
[40] Ke-Wen Huang and Hui-Ming Wang. 2018. Identifying the fake base station: A location based approach. *IEEE Communications Letters* 22, 8 (2018), 1604–1607.
[41] Syed Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A systematic approach for adversarial testing of 4G LTE. In *Network and Distributed Systems Security (NDSS) Symposium 2018*.
[42] Syed Rafiul Hussain, Mitziu Echeverria, Imtiaz Karim, Omar Chowdhury, and Elisa Bertino. 2019. 5GReasoner: A property-directed security and privacy analysis framework for 5G cellular network protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 669–684.
[43] Eunsoo Kim, Min Woo Baek, CheolJun Park, Dongkwan Kim, Yongdae Kim, and Insu Yun. 2023. BASECOMP: A Comparative Analysis for Integrity Protection in Cellular Baseband Software. In *32nd USENIX Security Symposium (USENIX Security 23)*. 3547–3563.
[44] Adrien Koutsos. 2019. The 5G-AKA authentication protocol privacy. In *2019 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 464–479.
[45] Felix Linker and David Basin. 2024. SOAP: A Social Authentication Protocol. In *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, 3223–3240. https://www.usenix.org/conference/usenixsecurity24/presentation/linker
[46] Fuwen Liu, Jin Peng, and Min Zuo. 2018. Toward a secure access to 5G network. In *2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*. IEEE, 1121–1128.
[47] Gavin Lowe. 1997. A hierarchy of authentication specifications. In *Proceedings 10th computer security foundations workshop*. IEEE, 31–43.
[48] The Tamarin Manual. 2024-12-10. https://tamarin-prover.com/manual/index.html.
[49] Ikbal Mawaldi, Tides Anugraha, Ishak Ginting, Nyoman Karna, et al. 2020. Experimental Security Analysis for Fake eNodeB Attack on LTE Network. In *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 140–145.
[50] Miroslav Mitev, Arsenia Chorti, H Vincent Poor, and Gerhard P Fettweis. 2023. What physical layer security can do for 6G security. *IEEE Open Journal of Vehicular Technology* 4 (2023), 375–388.
[51] Jorge Munilla, Mike Burmester, and Raquel Barco. 2021. An enhanced symmetric-key based 5G-AKA protocol. *Computer Networks* 198 (2021), 108373.
[52] Van-Linh Nguyen, Po-Ching Lin, Bo-Chao Cheng, Ren-Hung Hwang, and Ying-Dar Lin. 2021. Security and privacy for 6G: A survey on prospective technologies and challenges. *IEEE Communications Surveys & Tutorials* 23, 4 (2021), 2384–2428.
[53] Balu L Parne, Shubham Gupta, Kaneesha Gandhi, and Shubhangi Meena. 2020. PPSE: Privacy preservation and security efficient AKA protocol for 5G communication networks. In *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 1–6.
[54] Tamarin Prover. 2024. https://github.com/tamarin-prover/tamarin-prover.
[55] Alexander J Ross, Bradley Reaves, Yomna Nasser, Gil Cukierman, and Roger Piqueras Jover. 2024. Fixing Insecure Cellular System Information Broadcasts For Good. In *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*. 693–708.
[56] Altaf Shaik. 2015. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. *arXiv preprint arXiv:1510.07563* (2015).
[57] srsRAN_4G. 2024. https://github.com/srsran/srsRAN_4G.
[58] srsRAN_Project. 2024. https://github.com/srsran/srsRAN_Project.
[59] USRP B210. 2024. Online, Available: https://www.ettus.com/product/details/UB210-KIT.
[60] Fabian Van Den Broek, Roel Verdult, and Joeri De Ruiter. 2015. Defeating IMSI catchers. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. 340–351.
[61] Yuchen Wang, Zhenfeng Zhang, and Yongquan Xie. 2021. Privacy-Preserving and Standard-Compatible AKA Protocol for 5G. In *USENIX Security Symposium*. 3595–3612.
[62] Dongzhu Xu, Anfu Zhou, Guixian Wang, Huanhuan Zhang, Xiangyu Li, Jialiang Pei, and Huadong Ma. 2022. Tutti: coupling 5g ran and mobile edge computing for latency-critical video analytics. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 729–742.
[63] Awaneesh Kumar Yadav, Manoj Misra, Pradumn Kumar Pandey, An Braeken, and Madhusanka Liyange. 2022. An improved and provably secure symmetric-key based 5G-AKA Protocol. *Computer Networks* 218 (2022), 109400.
[64] Awaneesh Kumar Yadav, Pradumn Kumar Pandey, Kuljeet Kaur, Satinder Singh, and Abbas Bradai. 2023. PSLP-5G: A provably secure and lightweight protocol for 5g communication. In *ICC 2023-IEEE International Conference on Communications*. IEEE, 4534–4539.
[65] Hexuan Yu, Changlai Du, Yang Xiao, Angelos Keromytis, Chonggang Wang, Robert Gazda, Y Thomas Hou, and Wenjing Lou. 2023. AAKA: An Anti-Tracking Cellular Authentication Scheme Leveraging Anonymous Credentials. In *Network and Distributed System Security Symposium (NDSS)*.
[66] Zhenfeng Zhang, Kang Yang, Xuexian Hu, and Yuchen Wang. 2016. Practical anonymous password authentication and TLS with anonymous client authentication. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 1179–1191.