Privacy-Preserving and Standard-Compatible AKA Protocol for 5G

Yuchen Wang\textsuperscript{1,2} \quad Zhenfeng Zhang\textsuperscript{1,†} \quad Yongquan Xie\textsuperscript{3,†}

wangyuchen@tca.iscas.ac.cn \quad zhenfeng@iscas.ac.cn \quad yqxie_oscca@163.com

\textsuperscript{1}TCA of State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
\textsuperscript{2}Alibaba Group
\textsuperscript{3}Commercial Cryptography Testing Center of State Cryptography Administration

Abstract

The 3GPP consortium has published the Authentication and Key Agreement protocol for the 5th generation (5G) mobile communication system (i.e., 5G-AKA) by Technical Specification (TS) 33.501. It introduces public key encryption to conceal the so-called SUPIs so as to enhance mobile users’ privacy. However, 5G-AKA is only privacy-preserving at the presence of passive attackers, and is still vulnerable to the linkability attacks from active attackers. An active attacker can track target mobile phones via performing these attacks, which puts the privacy of users at risk.

In this paper, we propose a privacy-preserving solution for the AKA protocol of 5G system denoted by 5G-AKA\textsuperscript{′}. It is resistant to linkability attacks performed by active attackers, and is compatible with the SIM cards and currently deployed Serving Networks (SNs). In particular, we first conduct an analysis on the known linkability attacks in 5G-AKA, and find out a root cause of all attacks. Then, we design a countermeasure with the inherent key encapsulation mechanism of ECIES (i.e., ECIES-KEM), and use the shared key established by ECIES-KEM to encrypt the challenges sent by a Home Network (HN). With this measure, a target User Equipment (UE) who receives a message replayed from its previously attended sessions behaves as non-target UEs, which prevents the attacker from distinguishing the UE by linking it with its previous sessions. Moreover, 5G-AKA\textsuperscript{′} does not raise additional bandwidth cost, and only introduces limited additional time costs from 0.02\% to 0.03\%. Finally, we use a state-of-the-art formal verification tool, Tamarin prover, to prove that 5G-AKA\textsuperscript{′} achieves the desired security goals of privacy, authentication and secrecy.

1 Introduction

Nowadays, the mobile communication system has become an integral part of daily activities. According to the investigation report published by GSM Association (GSMA) [54], over 5 billion people have subscribed to mobile services by the end of 2018, which accounts for 67\% of global population. It is also expected that the scale of the mobile communication system will keep increasing in the next 5 years with the global deployment of mobile network infrastructures and Internet-of-Things (IoT) devices.

In Technical Specification (TS) 33.501 [20], 3GPP describes new versions of Authentication and Key Agreement (AKA) protocols for 5G (i.e., 5G-AKA), which enables an User Equipment (UE) and a Home Network (HN) to authenticate each other and establish key materials (a.k.a., anchor keys) for subsequent 5G procedures. 5G-AKA inherits many of the design characteristics from the AKA protocols for 3G and 4G, including the usage of a challenge-response procedure and the employment of sequence numbers.

Compared with the AKA protocols for 3G and 4G, 5G-AKA makes progress on protecting the privacy of users by disallowing the unsecure plaintext transmission of permanent identifiers of subscribers (i.e., SUPIs). TS 33.501 states that an SUPI must be concealed (i.e., encrypted) by the Elliptic Curve Integrated Encryption Scheme (ECIES) [3] algorithm with HN’s public key when it is sent over the radio, which prevents the notorious IMSI-catching attack [53]. With this measure, a passive attacker who can only monitor the wireless traffic will neither access the SUPI in plaintext, nor trace a UE across its 5G-AKA sessions via SUPI.

However, recent research [24,26,43] also find that 5G-AKA is still vulnerable to a series of attacks on privacy performed by active attackers. Compared with a passive attacker, an active attacker can furthermore emit radio signals actively (e.g., using rogue base stations). In particular, these attacks enable the attacker to distinguish a target UE from a set of UEs via replaying the messages from its previously attended AKA sessions, and are also known as linkability attacks since the attacker can link the target UE with its previous AKA sessions. With these attacks, the attacker can monitor or track the target UE and even infer an user’s real-world identity from the mobile activity pattern of its UE [37].

Moreover, these attacks can also be exploited to track the
target UE across mobile communication protocols of different generations (e.g., 4G and 5G), as they also exist in 3G and 4G AKA protocols [22, 27, 35]. In particular, the attacker can link the target UE in 5G with its 3G or 4G AKA sessions, on the premise that its SIM card\(^1\) remains unchanged. Currently, many major mobile providers (e.g., China Mobile [6], Three [10] and Vodafone [14]) have announced that their 5G services will not require mobile users to change legacy 4G SIM cards, which makes the premise realistic. To be more specific, we also present the cross-protocol linkability attack in Figure 1: First, the attacker captures a 3G or 4G AKA session of the target UE, which includes its identifier SUPI, and then designs an elaborate attack vector with the recorded messages. Next, in 5G network, the attacker uses the vector to launch linkability attacks on all UEs in the attack area, and distinguishes the target UE with its unique response. Such threat scenario does not only enable the attacker to track high-value target users (such as spy on embassy officials and journalists [26]), but also leaks the SUPI of a 5G-AKA session, which breaks the premise realistic. To be more specific, we also present the cross-protocol linkability attack in Figure 1: First, the attacker captures a 3G or 4G AKA session of the target UE, which includes its identifier SUPI, and then designs an elaborate attack vector with the recorded messages. Next, in 5G network, the attacker uses the vector to launch linkability attacks on all UEs in the attack area, and distinguishes the target UE with its unique response. Such threat scenario does not only enable the attacker to track high-value target users (such as spy on embassy officials and journalists [26]), but also leaks the SUPI of a 5G-AKA session, which breaks the premise realistic.

Currently, active attackers have been regarded as realistic threats for most 5G use cases [24, 26, 50] with the rapid development of open-source solutions of 5G communication [7–9]. Thus, it is reasonable to assume that a real-world attacker can mount active attacks with acceptable cost in the 5G era as in 4G [36, 44, 51] upon the completion of 5G standards, which makes the improvement of 5G-AKA important and urgent.

However, improving the privacy of 5G-AKA is not a trivial task, as: 1) There exists several kinds of linkability attacks. A satisfying solution must fix all of them “in one shot”. 2) The proposed fix should be compatible with 3GPP’s current specifications for 5G network (e.g., the SIM card commands defined by TS 31.102 [12]). Except for the effort to modify all involved standards, a non-compatible proposal would also require the communication provider to change the SIM card for all users, and all SNs to modify their implementations accordingly, which makes it cumbersome to be deployed in practice due to the high migration cost.

1.1 Our Contributions

In this paper, we propose 5G-AKA’ as a privacy-preserving solution for the AKA protocol of 5G system. It is able to prevent the linkability attacks mounted by active attackers, and is compatible with the standard AUTHENTICATE SIM card command [12] and the specifications of 5G network [15, 16, 18, 20], and thus can be deployed by reusing the current SIM cards and SNs’ implementations. The contributions of this paper are listed as follows:

- **An In-Depth Analysis on Known Linkability Attacks.** We first conduct an in-depth analysis on all known linkability attacks, and surprisingly find that all these attacks can be ascribed to the same root cause. In a nutshell, these attacks are all raised by the two-step check that a UE performs on the HN’s challenge. The attacker can thus use the target UE’s previous session to elaborate an attack vector which includes a replayed message. The replayed message can pass the target UE’s first check on the Message Authentication Code (MAC) of the challenge, as the message contains a valid MAC value calculated with the key shared by the target UE and HN, but cannot pass the other UEs’ checks as the MACs are incorrect, which makes them behave differently (i.e., respond with different messages).

- **Fix the Privacy Issues in 5G-AKA.** To fix this issue, we propose a countermeasure of encrypting the challenge sent by HN with a temporary shared key established from the key encapsulation mechanism inherent in ECIES. This key varies in different sessions, and enables the UE to check the message’s validity and freshness simultaneously via checking its MAC. A replayed message fails target UE’s check on MAC as non-target UEs, as a replayed challenge encrypted under the key in a previous session will be decrypted by the UE with a new key in current session and result in a different challenge from the original one.

We integrate this countermeasure to 5G-AKA, and denote the fixed protocol by 5G-AKA’. It can protect users’ privacy against the aforementioned linkability attacks, with the following subtleties: 1) It does not introduce new primitive to 5G system, as the ECIES algorithm have been included in TS 33.501 [20]. 2) It is compatible with the standard AUTHENTICATE SIM command without modification. This command has been provided by legacy 3G and 4G SIM cards, which enables 5G-AKA’ to be deployed in a way of reusing SIM cards. This feature will be useful in the migration to 5G, as swapping all SIM cards is commonly considered as a costly and cumbersome operation for mobile communication providers. 3) It is compatible with 3GPP’s specifications for 5G networks (e.g., TS 23.502 [15] and

---

\(^1\)In this paper, we use SIM card to refer to the 3G, 4G and 5G USIM applications and the physical smart card that carries them.
In this section, we first review the works which analyze 5G-AKA with formal methods, and then present the works that try to fix the weakness of linkability attacks for 5G-AKA. We summarize the proposals that improve the privacy of 5G-AKA in Table 2, and compare them with 5G-AKA'.

**Formal verification on 5G-AKA.** Formal methods have been widely accepted for evaluating the security of 3GPP AKA protocols and their variants for 3G and 4G systems [1, 22, 55]. In [24], Basin et al. formally refine the security and privacy properties required by 5G-AKA from 3GPP’s specifications [5, 19, 20] and evaluate 5G-AKA with Tamarin Prover [47]. They provide missing security assumptions which are necessary for achieving the desired security goals, such as key confirmation and channel binding, and prove the existence of Failure Message Linkability Attack in 5G-AKA. In a concurrent work by Cremers et al. [31], a fine-grained analysis is performed. It points out an attack raised by the potential race condition between the components residing within an HN, and also discusses various compromising scenarios and trust assumptions in 5G-AKA.

**Improving the privacy of 5G-AKA.** To improve the privacy of mobile subscribers, a series of pseudonym mechanisms have been designed and suggested as proposals for 5G-AKA [41, 42, 55]. These schemes protect the confidentiality of user identities via using changing pseudonym identifiers instead of the persistent ones (i.e., SUPIs), but can not completely fix the privacy issues of 5G-AKA as the linkability attacks performed by active attackers can not be prevented.

Arapinis et al. [22] suggest encrypting the reason of failure so as to avoid the failure message linkability attack. But the proposed scheme involves the trouble of changing all SNs, as an SN must decrypt the failure message first. Borgaonkar et al. [26] find a variant of failure message linkability attack and denote it by sequence number inference attack. To avoid this attack, they propose three countermeasures to enhance the SNQ concealment mechanism while preserving the compatibility with SN, which include encrypting $SQN_{UE}$ with symmetric and asymmetric encryption schemes, as well as using a freshly generated random number to conceal $SQN_{UE}$. However, these fixes cannot prevent the encrypted SUPI replay attack given by Fouque et al. [35] and Koutsos [43].

To eliminate all linkability attacks, Koutsos [43] proposes an AKA+ protocol for 5G communication, which is resistant to all known privacy threats by re-arranging the message flow of 5G-AKA. However, AKA+ changes the protocol flow and terms of messages of 5G-AKA significantly. For UE, $SQN_{UE}$ is encrypted together with SUPI, and $SQN_{HN}$ is no longer parsed and checked. The UE’s operation of $AKA^+$ cannot be implemented with the standardized commands provided by SIM cards, which implies that all subscribers’ SIM cards must be replaced. The terms of messages in $AKA^+$ cannot be

<table>
<thead>
<tr>
<th>Abbreviations &amp; Notations</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>HN</td>
<td>Home Network</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>SN</td>
<td>Serving Network</td>
</tr>
<tr>
<td>KEM</td>
<td>Key Encapsulation Mechanism</td>
</tr>
<tr>
<td>DEM</td>
<td>Data Encapsulation Mechanism</td>
</tr>
<tr>
<td>SUPI</td>
<td>Subscriber Permanent Identifier</td>
</tr>
<tr>
<td>SUCI</td>
<td>Subscriber Concealed Identifier</td>
</tr>
<tr>
<td>$k$</td>
<td>The permanent key shared between a UE and HN</td>
</tr>
<tr>
<td>$K_{seaf}$</td>
<td>The anchor key derived from 5G-AKA</td>
</tr>
<tr>
<td>$k_{UE}$</td>
<td>The UE’s shared key established by ECIES-KEM</td>
</tr>
<tr>
<td>$k_{HN}$</td>
<td>The HN’s shared key established by ECIES-KEM</td>
</tr>
<tr>
<td>$(PK_{HN}, sk_{HN})$</td>
<td>The HN’s ECIES public-private key pair</td>
</tr>
<tr>
<td>$SQN_{UE}$</td>
<td>The UE’s sequence number</td>
</tr>
<tr>
<td>$SQN_{HN}$</td>
<td>The HN’s sequence number</td>
</tr>
<tr>
<td>RAND</td>
<td>The HN’s challenge message</td>
</tr>
</tbody>
</table>

Table 1: A summary of abbreviations and notations

TS 24.301 [17]), and can be developed on the top of current implements of SNs. 4) It provides the desired properties of authentication and secrecy as defined by TS 33.501 [20]. 5) Compared with 5G-AKA, it does not raise additional bandwidth cost, and only raise additional computation costs from 0.02% to 0.03%.

• **Formal Verification on 5G-AKA’.** We formally analyze 5G-AKA’ in the symbolic model with Tamarin Prover. In particular, we first prove that 5G-AKA’ satisfies the goals of authentication and secrecy, based on the script proposed by Basin et al. [24]. Then, we implement a new script that captures the desired privacy goal and prove that 5G-AKA’ is privacy-preserving against active attackers. Our formal analysis models ECIES abstractly, which makes the results also fit for the variants of 5G-AKA’ that use the other asymmetric encryption schemes following the KEM/DEM paradigm (e.g., post-quantum KEMs [21, 28]).

## 1.2 Organization

In Section 2, we briefly review the related works and compare our result with the previous works which also focus on linkability attacks of 5G-AKA. In Section 3, we present the ECIES algorithm, 5G-AKA protocol and the details of known linkability attacks. In Section 4, we give the threat model and security goals that we consider throughout this work. In Section 5, we present the analyze on linkability attacks, the corresponding countermeasure, and give the detail of 5G-AKA’ with a performance evaluation. We describe the formal verification in Section 6, and conclude in Section 7.
This design idea also refers to the well-known Mechanism (KEM) and a Data Encapsulation Mechanism (DEM) [52]. This design idea also refers to the well-known Mechanism (KEM) and a Data Encapsulation Mechanism (DEM) [52]. In particular, it is a “hybrid” ECIES for frequently used abbreviations and notations.

In this section, we first present an important component of 5G-AKA, ECIES, in a component based manner. Then, we give out a detailed description on 5G-AKA following with the known linkability attacks. We refer the readers to Table 1 for frequently used abbreviations and notations.

| Pseudonym-based proposals [41,42,55] | - | - | - | - |
| Encrypt the failure reason [22] | - | - | - | - |
| Enhance the SQN concealment mechanism [26] | - | - | - | - |
| AKA+ [43] | - | - | - | - |
| DH-based proposals [23,45] | - | - | - | - |
| 5G-AKA’(This work) | √ | √ | √ | √ |

Table 2: The proposals for improving the privacy of 5G-AKA

<table>
<thead>
<tr>
<th>UE</th>
<th>HN</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initiation phase.</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>The challenge-response phase.</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1. Check the validity of RAND</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2. Check the freshness of RAND</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3. Calculate RES and $K_{sdf}$</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(RAND, AUTN)</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(Sync_Failure, AUTS)</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(RES)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: An overview of 5G-AKA

handled by the currently deployed SNs. The migration from 5G-AKA to AKA+ requires that all users must change their SIM cards and all SNs are needed to modify their implementations accordingly. In the proposals for 5G-AKA by Arkko et al. [23] and Liu et al. [45], a Diffie-Hellman (DH) key exchange procedure is introduced on the basis of 4G-AKA for the purpose of privacy-preserving. However, they require extra round trips for key exchange, which does not only significantly increase latency, but is also incompatible with the SIM cards and SNs’ implementations.

3 Background

In this section, we first present an important component of 5G-AKA, ECIES, in a component based manner. Then, we give out a detailed description on 5G-AKA following with the known linkability attacks. We refer the readers to Table 1 for frequently used abbreviations and notations.

3.1 ECIES

ECIES is an asymmetric encryption algorithm that can handle message of arbitrary length. In particular, it is a “hybrid” encryption scheme which consists of a Key Encapsulation Mechanism (KEM) and a Data Encapsulation Mechanism (DEM) [52]. This design idea also refers to the well-known KEM/DEM paradigm, which uses KEM to establish shared keys between the sender and recipient with asymmetric crypto, and uses DEM to encrypt and decrypt the actual payload with that shared key using symmetric crypto. This paradigm has been extensively used in practice and standards [2, 38, 48].

The ECIES-KEM consists of the following algorithms:

- **KeyGen**(pp): It takes a public parameter pp as input, and outputs a private-public key pair (sk, PK) such that $PK = sk \cdot G$, where pp is commonly a standardized parameter such as secp256r1 [4], and $G \in pp$ is a base point.
- **Encap**(PK): It takes a public key PK as input, generates an ephemeral private-public key pair (r, R) such that $R = r \cdot G$, and outputs a ciphertext $C_0 = R$ and a key $k_s = KDF(r \cdot PK)$, where KDF is a key derivation function.
- **Decap**(sk, C0): It takes a ciphertext $C_0$ and a private key sk as input, and outputs $k_s = KDF(sk \cdot C_0)$ as the shared key.

The ECIES-DEM consists of the following algorithms:

- **SEnc**(ks, M): It takes a key $k_s$ and a message M as input, parses $k_s$ as $k_1||k_2$, computes $C_1 = ENC(k_1, M)$ and $C_2 = MAC(k_2, C_1)$, and outputs ($C_1, C_2$), where ENC is the encryption operation of a symmetric encryption scheme.
- **SDec**(ks, C1, C2): It takes a ciphertext ($C_1, C_2$) and a key $k_s$ as input, parses $k_s$ as $k_1||k_2$, outputs ⊥ if $C_2 \neq MAC(k_2, C_1)$, and outputs $M = DEC(k_1, C_1)$ otherwise, where DEC is the decryption operation of a symmetric encryption scheme.

In TS 33.501 [20], ECIES refers to the mechanism specified by SEC1 [3]. The ECIES encryption algorithm takes PK and M as inputs, sequentially runs Encaps(PK) and SEnc(ks, M), and outputs a ciphertext $C = (C_0, C_1, C_2)$; the ECIES decryption algorithm takes sk and C as inputs, sequentially runs Decaps(sk, C0) and SDec(k1, C1, C2), and outputs M or ⊥. We denote the ECIES-KEM scheme by $\mathcal{KEM}_{ECIES} = \{\text{KeyGen}_{ECIES}, \text{Encaps}_{ECIES}, \text{Decaps}_{ECIES}\}$, and denote the DEM scheme by $\mathcal{DEM}_{ECIES} = \{\text{SEnc}_{ECIES}, \text{SDec}_{ECIES}\}$, and refer to the corresponding specifications [3, 20] for details.
3.2 The 5G-AKA Protocol

Next, we present the 5G-AKA protocol [20] in detail, which consists of an initiation phase and a challenge-response phase. Our description does not distinguish the components that reside within an HN as some related works that also focus on the privacy of 5G-AKA [26, 40, 42]. Nonetheless, it includes a detailed description on the root cause of known linkability attacks of 5G-AKA, and also provides explicit boundaries of the compatibility with legacy SIM cards. We begin with an overview, and then present the details of each phase.

Overview. We present an overview of the 5G-AKA protocol by Figure 2. In the initiation phase, the UE encrypts SUPI with the HN’s public key using ECIES, and sends the ciphertext (i.e., SUCI) to the HN through the radio channel via a base station. In the challenge-response phase, the HN chooses a random challenge (i.e., RAND), and calculates AUTN. In particular, AUTN contains MAC and concealed SQNHN. The UE uses MAC to verify the authenticity and integrity of RAND (for simplicity, we also say the UE utilizes MAC to verify the validity of RAND), and uses SQNHN to check the freshness of RAND. Upon receiving the RAND, AUTN, the UE first checks the message’s validity, and returns a MAC failure message if this check fails. Then, it checks the message’s freshness via comparing SQNHN with SQNUE, and returns a Sync Failure, AUTN message if this check fails, where UE uses AUTS to re-synchronize with the HN. When all checks pass, the UE generates a response RES for RAND, calculates the key material for subsequent procedures (i.e., Kseaf), and sends RES to the HN.

When a UE is unable to communicate to its HN directly (e.g., in roaming scenarios where the HN’s base station is not available), it may attach to a Serving Network (SN) who provides local mobile communication services. In such scenarios, the messages shown in Figure 2 are transmitted with the help from SN, where the UE communicates with the SN (i.e., the SN’s base station) over the radio channel, and the SN communicates with the HN via a wired channel provided by the 5G Core network (5GC).

We next give the details for each phase, where we use IDSN (resp., IDHN) as the unique identifier of SN (resp., HN), and denote the SHA-256 cryptographic hash function by HSHA256.

The Initiation Phase. This phase is shown by Figure 3. After the session between UE and SN has been initialized, the UE encrypts its SUPI with PKHN using ECIES, where we denote the shared key by kSUPI. Upon receiving the message from UE, the SN sends SUCI to the HN. Then, it sends SUCI to the HN. The HN decrypts SUCI with its private key and retrieves the corresponding k and SQNHN from its database, where we denoted the shared key by kHN.

According to TS 33.501 [20], the encryption of SUPI can either be performed with the SIM cards of next generation [12], or outside the SIM cards. In this paper, we follow the option that the encryption is carried out by the UE outside the SIM card, which satisfies the specification of TS 33.501 [20], and is more friendly to legacy 3G and 4G compatible SIM cards which only support the AUTHENTICATE command as specified by TS 31.102, Release-14 [13].

The Challenge-Response Phase. In this phase, the UE and HN mutually authenticate each other via a challenge-response procedure, and establish anchor keys (i.e., Kseaf) together with the SN, as shown in Figure 4. This phase contains a series of cryptographic functions f1, f2, f3, f4, f5, f6 and f7 as specified by TS 33.501 [20]. Furthermore, we also denote the derivation processes of anchor keys by a KeyDerivation function for the sake of simplicity. It takes k, RAND, IDSN and SQNUE (or SQNHN) as inputs and includes the calculations of f3 and f4.

At the beginning of this phase, the HN generates an Authentication Vector AV = (RAND, AUTN, HXRES, Kseaf):

- Choose a 128-bit nonce RAND as challenge.
\( \text{RAND} \leftarrow \{0, 1\}^{128} \)

\( \text{MAC} \leftarrow f_1(k, \text{SQN}_{\text{HN}}, \text{RAND}), \text{AK} \leftarrow f_5(k, \text{RAND}) \)

\( \text{AUTN} \leftarrow \{\text{AK} \oplus \text{SQN}_{\text{HN}}, \text{MAC}\} \)

\( \text{XRES} \leftarrow f_2(k, \text{RAND}), \text{HXRES} \leftarrow H_{\text{SHA256}}(\text{RAND}, \text{XRES}) \)

\( K_{\text{seaf}} \leftarrow \text{KeyDerivation}(k, \text{RAND}, \text{ID}_{\text{SN}}, \text{SQN}_{\text{HN}}) \)

\( \text{SQN}_{\text{HN}} \leftarrow \text{SQN}_{\text{HN}} + 1 \)

Figure 4: The Challenge-Response Phase of 5G-AKA, where the implicit key authentication is expressed by a Key Confirmation round-trip.
AUTHENTICATE(RAND, AUTN):

\[ AK \leftarrow f_5(k, RAND) \]

Parse AUTN as \( \{AK \oplus SQN_{HN}, MAC\} \)

Check \( f_1(k, RAND, SQN_{HN}) = MAC \)

If this check does not pass, return \( \perp \)

Check \( SQN_{UE} < SQN_{HN} < SQN_{UE} + \Delta \)

If this check does not pass:

\[ MAC^* \leftarrow f_1(k, RAND, SQN_{UE}) \]

Return \( AUTS \leftarrow \{f_2(k, RAND) \oplus SQN_{UE}, MAC^*\} \)

\( SQN_{UE} \leftarrow SQN_{HN} \)

\( K_{seaf} \leftarrow \text{KeyDerivation}(k, RAND) \)

\( RES \leftarrow f_2(k, RAND) \)

Return \( (K_{seaf}, RES) \)

Figure 5: The AUTHENTICATE SIM command, where the condition marked by \( \dagger \) is optional following the non-normative Annex C of TS 33.102 [19]. \( k \) and \( SQN_{UE} \) are secrets stored by the SIM card.

- Calculate AUTN, which includes the concealed \( SQN_{HN} \) and \( MAC \). Particularly, \( SQN_{HN} \) is concealed with an anonymous key \( AK \) derived from RAND and \( k \), and \( MAC \) is computed with \( k \), RAND and \( SQN_{HN} \)
- Calculate \( HXRES \) by hashing RAND and XRES, where XRES is the expected response computed with \( k \) and RAND.
- Derive \( K_{seaf} \) with \( k \), RAND, \( IDS_N \) and \( SQN_{HN} \).
- Increase \( SQN_{HN} \) by 1.

Then, the HN sends AV to SN. Upon receiving AV, the SN stores HXRES, RAND and \( K_{seaf} \), and sends (RAND, AUTN) to the UE. Next, the UE checks the message and calculates the response by calling the SIM card’s AUTHENTICATE command with (RAND, AUTN). This command is shown in Figure 5, and also described as follows:

- The SIM card calculates \( AK \) with \( k \) and RAND via \( f_5 \), parses AUTN as \( \{CONC, MAC\} \), and de-conceals \( SQN_{HN} \).
- Then, it checks the validity of RAND and \( SQN_{HN} \) with \( MAC \). If this check fails, the SIM card responds with a failure message (denoted by \( \perp \)). Then, the UE sends a Mac_Failure message to SN (See Case i in Figure 4).
- Next, it checks the freshness of AV with \( SQN_{HN} \). If this check fails, the SIM card responds with an AUTS message which conceals \( SQN_{UE} \). Then, the UE re-synchronizes with HN by sending Sync_Failure and AUTS to the SN (See Case ii in Figure 4).
- If all checks hold, the SIM card sets \( SQN_{UE} \) by \( SQN_{HN} \), derives \( K_{seaf} \) with \( k \), RAND, \( IDS_N \) and \( SQN_{HN} \), calculates a response \( RES \) using \( k \) and RAND, and finally returns \( (K_{seaf}, RES) \). The UE stores \( K_{seaf} \), and sends RES to the SN (See Case iii in Figure 4).

Upon receiving RES, the SN checks its validity by calculating the hashed value of RES and RAND, and comparing it with HXRES. It then forwards RES to the HN. Next, the HN authenticates UE by comparing RES with its stored XRES, and sends SUPI to the SN if they are matching. The SN continues the protocol only when both checks hold, and rejects the authentication otherwise.

When all checks pass, the SN and UE communicate with the session keys derived from anchor keys (i.e., \( K_{seaf} \)) in subsequent 5G procedures. TS 33.501 [20] also specifies that the UE and SN should confirm the keys agreed and the identities of each other implicitly through the successful use of keys in subsequent procedures, which can be expressed by a key-confirmation round trip with \( K_{seaf} \).

Re-synchronization between UE and HN. In the following, we give a more detailed description on the re-synchronization mechanism of sequence numbers between the UE and HN. It allows the UE to verify the freshness of message and reject a replayed message.

A UE checks the freshness of (RAND, AUTN) via verifying \( SQN_{UE} < SQN_{HN} \), and optionally checks \( SQN_{HN} < SQN_{UE} + \Delta \). The former condition ensures that a replayed message can be detected and rejected, and the latter is designed to prevent the wrap around of \( SQN_{UE} \). Moreover, 3GPP also provides a recommended value of \( \Delta = 2^{28} \) in TS 33.102 [19] so as to decrease the failure rate due to synchronization failure.

If this check fails, the UE re-synchronizes with the HN by sending a concealed \( SQN_{UE} \) in an authenticated manner with an AUTS, where RAND is used to generate \( AK^* \) and \( MAC^* \) as shown by Figure 5. Upon the reception of UE’s re-synchronization message, the SN and HN interact as in Figure 4. In particular, the SN sends (Sync_Failure, AUTS, RAND, SUCI) to the HN. Then, the HN de-conceals \( SQN_{UE} \) with the anonymity key derived from \( k \) and RAND, and checks its authenticity with \( MAC^* \). If the check holds, the HN re-sets \( SQN_{HN} \) by \( SQN_{UE} + 1 \).

3.3 The Linkability Attacks in 5G-AKA.

Currently, three types of linkability attacks have been found in 5G-AKA, which are described as follows:

- Failure Message Linkability Attack [22, 24]. In this attack, the attacker records a (RAND, AUTN) message that the HN sends to the target UE, and replays it to all UEs in the attack area. Upon receiving such a message, the target UE passes the check on MAC as it is generated with the correct \( k \), but fails the next check on freshness since the message is replayed, and replies with a Sync_Failure message, while the other UEs all fail the check on MAC and reply with MAC_Failure messages.
• Sequence Number Inference Attack [26]. This attack is performed in the same way as the first attack, where a \((RAND, AUTN)\) is replayed. But it furthermore enables the attacker to obtain the increase pattern or even particular digits of the target UE’s \(SQN_{UE}\). In particular, the attacker replays a \((RAND, AUTN)\) several times, where each time the target UE replies with a synchronization failure message containing \(CONC_{SQN_{UE}} \leftrightarrow SQN_{UE} \oplus f'_{1}(k, RAND)\). Then, denoting the target UE’s \(SQN_{UE}\) in two different tests by \(SQN_{UE}^{1}\) and \(SQN_{UE}^{2}\), the attacker can learn \(SQN_{UE}^{1} \oplus SQN_{UE}^{2}\) with \(CONC_{SQN_{UE}}\), as \(SQN_{UE}^{1}\) and \(SQN_{UE}^{2}\) are concealed with the same key via the XOR operation.

• Encrypted SUPI Replay Attack [35, 43]. In this attack, the attacker records an \(SUCI\) sent by the target UE and replays it to the HN in all UEs’ sessions, and waits for the UEs to reply to the HN’s challenge messages. The target UE will reply without failure message (i.e., both checks hold), while the others will all send \(MAC\_Failure\) messages as the HN uses the \(k\) shared with the target UE to calculate their MACs.

4 Threat Model and Security Goals

In this section, we present the threat model that we consider in this paper as well as the desired goals for the AKA protocol in 5G system. In particular, the threat model is based on previous works by Basin et al. [24], Borgaonkar et al. [26] and Cremers and Dehnel-Wild [31]. For the security goals, we give out a specified goal for privacy by a mean of indistinguishability, and follow Basin et al. [24] and Cremers and Dehnel-Wild [31] for the goals of secrecy and authentication.

4.1 Threat Model

The presented threat model does not only include the requirements according to TS 33.501 [20], but also contains the supplementary assumptions provided by Basin et al. [24] and Cremers and Dehnel-Wild [31], since they have been formally proved to be necessary for the security of 5G-AKA, and submitted to 3GPP for future standardization.

Assumptions on Channels. We next present the assumptions on both channels in 5G network. For the radio channel, we allow the existence of both passive and active attackers, as TS 33.501 [20] does not present any security assumption or requirement for this channel. In particular, an active attacker can eavesdrop, manipulate, and inject messages on this channel, and is also allowed to command UEs to identify themselves by actively starting new AKA sessions.

For the wired channel on which an SN communicates with an HN, TS 33.501 [20] explicitly specifies its security requirements as “e2e core network interconnection” channel. This channel guarantees the confidentiality and integrity of messages transferred in a mutually authenticated manner, and is resistant to message replay. Besides the requirements specified by TS 33.501 [20], we furthermore assume that this channel is binding, where each message is bound to a session identified by an unique session ID, since previous works of Basin et al. [24] and Cremers and Dehnel-Wild [31] have shown the necessity of such an assumption in 5G-AKA.

Assumptions on Functions. The attacker is allowed to execute all functions involved in 5G-AKA with its chosen inputs. We assume that \(f_1, f_2, f_3, f_4, f_5, f_6, f_7\) protect both confidentiality and integrity of their inputs following Basin et al. [24], and \(KEM_{ECIES}, DEM_{ECIES}\) are secure w.r.t., the standard security definitions of KEM and DEM by Shoup [52].

Assumptions on Components. We do not allow the attacker to compromise any component that resides within 5GC (i.e., SNs and HNs) according to TS 33.501 [20], which implies that the attacker can neither steal their long term secrets (e.g., \(k\) and \(sk_{HN}\)) nor temporary secrets (e.g., \(K_{saf}\)). Furthermore, we do not allow the attacker to steal the long-term key \(k\) as well as \(SQN_{UE}\) from an honest user’s UE, and also assume that the UE can protect all temporary secret information established in an AKA session such as \(K_{saf}\). We only allow the attacker to compromise the keys and secrets of UEs in its possession.

4.2 Security Goals

In the following, we first provide a more specific goal for privacy, as 3GPP’s specifications only present weak, or “underspecified” privacy goals [24, 26], which are unable to cover the cases of linkability attacks or protect the users’ privacy in practice. Then we present the goals of secrecy and authentication by reusing the ones proposed by Basin et al. [24] and Cremers and Dehnel-Wild [31].

Privacy. We first review the privacy goals desired by 3GPP for 5G-AKA in order to find out their drawbacks, and then give out a more specified goal from the view of practical attack scenarios. In TS 33.102 [19], 3GPP has identified three privacy requirements related to the privacy of mobile users including user Identity confidentiality, user location confidentiality and user untraceability, but only in the presence of passive attackers. Basin et al. [24] interprets these privacy requirements into three individual goals:

• The \(SUPI\) must remain secret.
• The values of \(SQN_{UE}\) and \(SQN_{HN}\) must remain secret.
• The untraceability of user must be provided.

However, these goals overlap with each other, and are not strong enough to protect the privacy of users in practice. If the attacker is able to obtain the \(SUPI\) for a 5G-AKA session, then it can naturally trace a UE with every AKA session it participates by stealing their \(SUPIs\). The attacker can also trace a UE once the value of \(SQN_{UE}\) or \(SQN_{HN}\) is leaked, since it can
determine the linkability between two AKA sessions with the variation of counters as shown by Borgaonkar et al. [26]. Furthermore, it is necessary to take active attacker into account, as active attackers have been commonly regarded as practical threats in 4G and forthcoming 5G systems [22, 24, 26, 55].

Hence, it is necessary to explore a reasonable way to define the privacy of users for the AKA protocol in 5G. A first approach is directly applying the notion of unlinkability from linkability attacks, which requires that the attacker cannot link the sessions participated by the same UE. Such a property is able to cover the untraceability of users as well as the other privacy goals as specified above. However, it is hard to define the action of “linking” sessions in a formal way, let alone checking whether such a property is actually satisfied. Thus, we leverage the notion of indistinguishability, which is the standard way for anonymous authentication systems [29, 30, 57] to claim privacy-preserving properties. It defines privacy in a strong sense which does not allow the attacker to determine which UE it is interacting with from two UEs. Furthermore, in the case of 5G AKA protocols, it is useful to explicitly allow the attacker to interact with one of the UEs to be distinguished before it actually begins the “game of indistinguishability”. It covers the cases of linkability attacks and is convenient to be formally modelled. If an attacker is able to distinguish a target UE with the others using the data form its previously attended AKA session, then the attacker can link the UE with that session, which is actually the case of linkability attack in practice [22, 24, 26, 43]. We define the privacy goal as follows:

**Goal 1 (UE Indistinguishability)**: Given two UE entities denoted by UE\(_1\) and UE\(_2\), and an AKA session attended by UE\(_1\) (or UE\(_2\)), no active attacker can determine whether it is interacting with UE\(_1\) or UE\(_2\).

**Secrecy.** We mainly focus on the secrecy of K\(_{\text{seaf}}\), as the privacy goal has implied the secrecy of long-term user identifiers and secrets. This goal is essentially identical to the secrecy goals refined by Basin et al. [24] and Cremers and Dehnel-Wild [31], and is presented in a simplified way:

**Goal 2 (Key Secrecy)**: K\(_{\text{seaf}}\) must be kept secret.

**Authentication.** We next list the desired goals for authentication following Basin et al. [24] and Cremers and Dehnel-Wild [31]. These goals are refined from TS 33.501 [20] in the form of agreement following Lowe’s taxonomy [46] of authentication properties, and are provided with corresponding formal definitions with Tamarin Prover.

**Goal 3 (Agreement between UE and SN)**: By the end of protocol execution, the UE and SN must both obtain injective agreement on K\(_{\text{seaf}}\) and weak agreement with each other.

**Goal 4 (Agreement between UE and HN)**: By the end of protocol execution, the UE and HN must both obtain injective agreement on K\(_{\text{seaf}}\) and weak agreement with each other. They also must both obtain non-injective agreement on ID\(_{\text{SN}}\) with each other.

**Goal 5 (Agreement between SN and HN)**: By the end of protocol execution, the SN and HN must both obtain injective agreement on K\(_{\text{seaf}}\) and weak agreement with each other. The SN must obtain non-injective agreement on SUPI with HN.

By weak agreement, we mean that a participant of the protocol has actually executed the protocol with its partner, but they do not have to agree on any data transferred or secret established in this session. Non-injective agreement implies that the participant should agree on the data or secrets with its partner on the basis of weak agreement. Injective agreement furthermore requires that there only exists one partner for the protocol execution and agree on the data or secrets, on the top of non-injective agreement.

### 5 5G-AKA’

In this section, we present our proposal for the AKA protocol of 5G system (i.e., 5G-AKA’). It is able to protect the privacy of users in the presence of active attackers, and only introduces minimal modifications on 5G-AKA in a way of compatible with legacy SIM cards and SNs’ implementations. Moreover, 5G-AKA’ does not involve additional bandwidth cost than 5G-AKA as it reuses the terms of messages, and only raises additional computational cost of less than 0.03%.

In a nutshell, 5G-AKA’ uses the shared keys that are established in ECIES-KEM to encrypt and decrypt RAND at the HN and UE side (See Figure. 7). In this section, we begin with the exploration of the root cause of the linkability attacks via checking the logic of UE, and then propose a targeted fix according to the cause. Such an approach makes our solution more reasonable, and also resistant to the undiscovered attacks raised by the same cause. Next, we describe 5G-AKA’ in detail and evaluate its performance. We do not only present the full message flow of the protocol, but also explain its difference with 5G-AKA and why it is standard compatible.

#### 5.1 Design Idea

We first reason about the root cause of linkability attacks in 5G-AKA, and then present a specific countermeasure against the root cause and explain its rationale. This approach makes it possible to avoid all existing linkability attacks “in one shot”, as well as to prevent undiscovered privacy issues raised by the same cause.

**Root Cause of Linkability Attacks.** In the typical scenario of linkability attacks, an active attacker distinguishes the target UE from a set of UEs when they behave differently to the same attack vector, and then links the target UE to its previously attended session via the association between the vector and the session. To trigger such distinguishable behaviors, the only way is to utilize the conditional statements in the process of a UE’s execution. Thus, it is reasonable to locate the root cause at the AUTHENTICATE SIM command, as it is
only process that includes conditional statements in a UE’s execution of 5G-AKA. In particular, this command involves two conditional statements sequentially for checking MAC and SQN$_HN$ respectively, as shown in Figure 5. Next, we dive into both of them and try to find out the root cause of those linkability attacks.

**Conditional Statement on Checking MAC.** In order to determine whether this conditional statement hides the root cause, an acceptable method is to find out whether its condition can lead to distinguishable behaviors with the same attack vector, which means the condition can only hold for the target UE, but fail for the others. The only way that the attacker can make this happen is to use attack vectors which include valid messages that are generated with a UE’s long term secret key $k$. That is to say, it can only trigger distinguishable behaviors with attack vectors containing replayed messages as it does not know $k$. It also allows the linkability between the target UE and its previously attended sessions, because the attacker can determine that the UE who passes the check on MAC is the same one as in the session where the attack vector comes from, as they have the same $k$.

In other words, this conditional statement only checks the authenticity (and integrity) of a received message, but does not check its freshness, which leaves space for the attacker to create attack vectors using valid but unfresh messages. In 5G-AKA, the check on freshness is postponed to the second conditional statement on checking SQN$_HN$, which raises the Failure Message Linkability Attack [22, 24] and Sequence Number Inference Attack [26], or even not performed in the case of SUCI replay, which raises the Encrypted SUPI Replay Attack [40, 43]. Our observation shows that these seemingly different linkability attacks are actually raised by the same root cause, and can be fixed all at once.

A possible fix for this root cause can be enabling freshness check in addition to the original purpose of this conditional statement, which enables a UE to reject all attack vectors, and behave as the other UEs even if these vectors include messages replayed from its previously attended sessions. However, designing such a fix is not easy, as the commands, message flows and data formats of 5G-AKA must be retained due to the requirement of compatibility.

**Conditional Statement on Checking SQN$_HN$.** When evaluating the second conditional statement, we assume that the checking of MAC has been able to reject a replayed message. With such an assumption, the only way that makes different UEs behave differently is that the target UE is not synchronous with the HN but the others are. However, this cannot be triggered by an attacker via intervening the sessions with the same attack vector, as only messages for the current session can pass the check on MAC, which eliminates the possibility that a UE executes following an attack vector. Furthermore, it also seems impossible for the attacker to link a UE who behaves inconsistently with a re-synchronization message to its previously attended sessions, as re-synchronization can take place due to many reasons such as the out-of-order delivery and re-transmission of messages. It is hard to determine whether a UE who re-synchronizes with the HN is exactly the one that has also re-synchronized with the HN in an old session, or is the one that has responded with a RES in an old session, as the other UEs can also be.

Through the above analysis, we find a root cause that can explain all known linkability attacks: The UE uses two separate conditional statements to check the validity and freshness of a message respectively. A possible fix is using one conditional statement to check both the validity and freshness, as shown in our proposed countermeasure.

**Countermeasure.** To solve the privacy issue of 5G-AKA, we propose a countermeasure of using a session key to guarantee the freshness of message and utilizing an one-pass message to establish a temporary session key, which is inspired by traditional security protocols [39, 48, 49]. Interestingly, these ideas coincide with the concealment of SUPI with ECIES introduced by TS 33.501, which allows the countermeasure to be designed in a standard-compatible manner.

The core idea is reusing the symmetric key established by ECIES as a “session key” to encrypt and decrypt the authentication challenge RAND. The decryption is performed by
Figure 7: The Challenge-Response Phase of 5G-AKA', where the implicit key authentication is expressed by a Key Confirmation round-trip. We use underlines to denote the differences with 5G-AKA, and † to denote that the encryption and decryption use AES-128 in ECB mode.
the UE before taking $RAND$ as the input for AUTHENTICATE, which avoids modifying this command. This measure does not require additional round trip as the key has been established and used by both the UE and HN in the encryption and decryption of SUPI, and only requires minimal modification to TS 33.501 [20]. It is also transparent to an SN, since the encrypted challenge does not need to be decrypted by the SN, and can be treated as a challenge in 5G-AKA.

This countermeasure enables freshness checks for the conditional statement on checking $MAC$ beside its original usage, and thus fix the root cause of linkability attacks. If a UE receives an encrypted (or plaintext) challenge that does not belong to the current session, where the challenge can be a replayed one in the cases of Failure Message Linkability Attack and Sequence Number Inference Attack, or is generated by an HN who receives a replayed SUCI in the case of Encrypted SUPI Replay Attack, the decryption algorithm will finally output a challenge that is different from the original one, which can not pass the check on $MAC$. This is due to the fact that the challenge is not encrypted by the correct key encapsulated by the UE in SUCI.

5.2 The Construction of 5G-AKA’

We present the detail of 5G-AKA’ by combining the countermeasure and 5G-AKA as follows:

The Initiation Phase. In this phase, the UE identifies itself to the HN with $SUCI$ as in Figure 6. Compared with 5G-AKA, our protocol only introduces limited modification which requires both the UE and HN to store the established shared keys for the challenge-response phase, where each time a 5G-AKA’ session is initialized, a fresh $k_{UE}$ (i.e., $k_{HN}$) is generated and used. It does not require additional cryptographic operation for both sides, as the key has been established via ECIES, and is also transparent to all 5G network specifications, as the message flow does not change.

The Challenge-Response Phase. In this phase, the involved components authenticate to each other and establish anchor keys as shown in Figure 7, where the differences with 5G-AKA are explained as follows.

First, when the HN generates $AV$, it additionally encrypts the authentication challenge with the shared key established at the initiation phase, and only includes the encrypted one in $AV$. In particular, the HN encrypts $RAND$ with AES-128 in electronic codebook (ECB) mode [34] but not the counter (CTR) mode, since the length of $RAND$ is 128-bit, and 3GPP only allocates a length of 128 bit for this message (See TS 24.501, Section 9.1.3.16 [18] and TS 24.008, Section 10.5.3.1 [16]), which is just coincident with the length of one block of AES-128. Any block cipher work mode which raises expansion on ciphertext requires modifications on the aforementioned specifications. Moreover, any manipulation of $RAND'$, or receiving a $RAND'$ encrypted with an incorrect key, will eventually fail the checking on $MAC$, as the UE only decrypts $RAND'$ with $k_{UE}$ and takes the output, which is different from the one that is used to generate $MAC$, as $RAND$. That is to say, even though we do not employ authenticated encryption, the UE will reject modified $RAND'$.

Another modification on the HN’s side is the way of calculating $HXRES$. Particularly, $HXRES$ is computed by $RAND'$ (i.e., the encrypted $RAND$) rather than $RAND$, since $HXRES$ must be computable by an SN who does not know $RAND$ in order to verify the UE’s response. This change makes our measure transparent to the SN, as it can just take $RAND'$ as $RAND$ and execute as in 5G-AKA [20].

Then, the UE needs to decrypt the authentication challenge before it is taken as an input for AUTHENTICATE. If the encrypted challenge is manipulated, replayed, or even honestly generated by the HN following a different $SUCI$, the SIM card will reply with ⊥ implying the check on $MAC$ does not hold, since the key used by UE to decrypt $RAND'$ does not match the key that encrypts it. Obviously, our approach is compatible with the AUTHENTICATE command as neither its inputs or outputs, nor its execution process is changed (See TS 31.102, Section 7.1 [12]).

Finally, the last necessary modification is presented when the HN checks the re-synchronization message (i.e., $AUTS$) sent from a UE. This message is generated by the SIM card using $RAND$, but the SN only sends $RAND'$ when forwarding this message. Thus, the HN also needs to decrypt $RAND'$ to $RAND$ before it starts to check $AUTS$.

Compared with 5G-AKA, our countermeasure only adds the encryption and decryption of $RAND$ with AES-128, which almost has no impact on efficiency. Furthermore, for practical usage, we suggest vendors and communication providers to apply technique measures to extend the out-of-order delivery of $RAND'$, such as the array scheme presented by TS 33.102 [19], so as to ensure the failure rate due to $MAC$ failure is acceptable.

5.3 Performance Evaluation

We next evaluate the performance of 5G-AKA’ and compare it with 5G-AKA. We focus on the additional time cost raised by 5G-AKA’, as 5G-AKA’ reuses the terms of messages of 5G-AKA and would not raise additional bandwidth cost. In particular, we run the execution processes of UE and HN of both protocols while considering every possible cases.

We use a workstation to run the process of a HN and a mobile phone for a UE. To be more specific, we use a MacBook 2019 workstation to run the process of an HN, and an iPhone 7 plus to run the process of a UE. The workstation equips with an Intel Core-i5 CPU which has 4 cores running at 2.4Ghz each, and runs a macOS Catalina 10.15.3 operating system. The mobile phone has an Apple A10 CPU which has 2 cores running at 2.34Ghz each, and runs an iOS 13.3 operating system. Note that we run the experiments with the Application Processor (AP) of the mobile device but not the
Table 3: The performance evaluation of 5G-AKA’. The superscripts 1, 2 and 3 mean that the UE and HN run in case i, ii, and iii shown in Figure 4 and Figure 7. The time+ line shows the additional time costs and their ratios compared with 5G-AKA.

<table>
<thead>
<tr>
<th></th>
<th>UE1</th>
<th>HN1</th>
<th>UE2</th>
<th>HN2</th>
<th>UE3</th>
<th>HN3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5G-AKA</td>
<td>13124.73</td>
<td>835.10</td>
<td>13158.29</td>
<td>853.44</td>
<td>13132.40</td>
<td>847.46</td>
</tr>
<tr>
<td>5G-AKA’</td>
<td>13128.65</td>
<td>835.27</td>
<td>13162.44</td>
<td>853.71</td>
<td>13136.25</td>
<td>847.64</td>
</tr>
<tr>
<td>time†</td>
<td>3.92 (0.03%)</td>
<td>0.17 (0.02%)</td>
<td>4.15 (0.03%)</td>
<td>0.27 (0.03%)</td>
<td>3.85 (0.03%)</td>
<td>0.18 (0.02%)</td>
</tr>
</tbody>
</table>

Baseband Processor (BP), which is enough for the purpose of comparing the relative difference between 5G-AKA and 5G-AKA’. We use the Crypto++ cryptographic library\(^2\) to implement ECIES with the secp256r1 curve, where we modify the `Encryptor.Encrypt()` and `Decryptor.Decrypt()` interfaces such that they can support the export and import of the shared keys derived by ECIES (i.e., \(k_{HN} \) and \(k_{UE} \)). Furthermore, we use SHA-256 with different prefixes as \{f\}\(_i\)=1, \(f_i^*\) and \(f_i^2\). Our "HN" program is implemented in C++ and compiled with clang 11.0.3 with -O2 flags. Our "UE" program is implemented in Objective-C and C++, which is compiled and deployed to the test device with Xcode 11.4. Both programs implement 5G-AKA and 5G-AKA’, where the implementations of 5G-AKA’ only involve the modifications of less than 20 Lines of Code (LoC) based on the implementations of 5G-AKA. Interestingly, the experimental implementations also imply that the migration from 5G-AKA to 5G-AKA’ can be achieved via only modifying a few LoCs for both endpoints, which makes our proposal easy to be deployed based on current implementations.

The results are shown by Table 3, where the costs for each endpoint in all execution cases are presented in microseconds and are taken the average of 1000 runs with the `chrono` library provided by C++11. The content of Table 3 demonstrates that 5G-AKA’ only brings limited additional time costs than 5G-AKA. For the UE side, the migration from 5G-AKA to 5G-AKA’ will only involves an extra time cost of 0.03%. For the HN side, 5G-AKA’ only brings 0.02% ~ 0.03% additional time cost.

### 6 Formal Verification

In this section, we evaluate the security of 5G-AKA’ with state-of-the-art symbolic verification tool Tamarin Prover\(^4\). Tamarin Prover is a powerful and efficient symbolic verification tool, and has been employed in the analyses of complex security protocols [24, 31, 32, 56]. To the best of our knowledge, it is also the only tool that can model the properties which are necessary for 5G-AKA [25, 33].

Our formal verification consists of two parts. For the first part, we prove that 5G-AKA’ satisfies the goals of authentication and secrecy, with a modified Tamarin Prover’s script for

\(^2\)https://www.cryptopp.com

\(^4\)https://www.tamarin-prover.org

![Figure 8: The configurations of RHS and LHS, where † denotes that this session can either be a 4G-AKA session or a 5G-AKA’ one.](image)

5G-AKA by Basin et al. [24]. In the second part, we develop a new script to verify that 5G-AKA’ is able to protect the privacy of users even in the presence of active attackers and achieves the desired goal for privacy. It borrows the idea of simplifying the components from the modeling presented by Basin et al. [24] so as to decrease the scale of search space, and is designed following a new idea of dividing the protocol execution with two AKA sessions in order to obtain a reasonable modeling for the goal of privacy.

In this section, we first present the major modeling choices for 5G-AKA’ and then the results of formal verification.

#### 6.1 The Modeling Choices of 5G-AKA’

The formal verification requires the modeling of ECIES in the KEM/DEM paradigm and a comprehensive script for privacy goals, where the major choices are described as follows:

- **Modeling ECIES following the KEM/DEM Paradigm.** In the modeling script, we model a generic asymmetric encryption scheme which is designed following the KEM/DEM paradigm, rather than just model ECIES with the Tamarin’s built-in theory which describes operations over a Diffie-Hellman group. This allows our formal analysis to satisfy a wide range of variants of 5G-AKA’ which are constructed by other asymmetric encryption schemes designed following the KEM/DEM paradigm.
To be more specific, we define four functions and an equation over these functions following Tamarin Prover’s grammars for customized primitives, including encap, getkey, getcipher, and decap, where encap and decap model the key encapsulation algorithm and key decapsulation algorithm respectively. However, encap cannot model the key encapsulation operation alone, since the real-world algorithm outputs a tuple consisting of two elements, but a function in Tamarin Prover only outputs one. To solve this problem, we use getkey to extract the temporary key from the output of encap, and use getcipher for the ciphertext which encapsulates the key. We refer the reader to Appendix A.2 for the details of these functions and equation.

**Modeling Privacy Goals.** Tamarin Prover provides a `diff` operator to model and check privacy-type (or cryptographic indistinguishable) properties. It takes two parameters as input. With this operator, Tamarin automatically generates two systems for one script, namely the Right Hand System (RHS) and the Left Hand System (LHS), where RHS applies the first parameter of `diff`, and LHS uses the second one. Besides RHS and LHS, Tamarin Prover also automatically generates a lemma called observational equivalence, which claims that the attacker cannot distinguish LHS from RHS (resp., RHS from LHS) in its view. A violation of this lemma implies that Tamarin Prover finds a path (i.e., a series of activities that a real-world attacker can follow) that makes the attacker to distinguish LHS from RHS (resp., RHS from LHS), which means an attack vector on privacy has been found.

To verify the goal of privacy, we want to check that whether an attacker can distinguish between $UE_1$ and $UE_2$ even if it has interacted with one of them (say, $UE_1$) and recorded the messages. Thus, we use RHS to model the case of two successive sessions of $UE_1$, and LHS for the case that the second session is attended by $UE_2$. If an attacker can distinguish between LHS and RHS, then it can leverage the detected path to mount linkability attacks in practice. The attacker is allowed to either record a 4G-AKA session or a 5G-AKA one in its first interaction with $UE_1$. The details are shown in Figure 8, and explained as follows:

- For the LHS, we denote the UE instances by $UE'_1$ and $UE'_2$, which corresponds to two successive sessions attended by the same $k$ and $SUPi$, and only differ in $SQN_{UE}$, where $SQN_{UE'_1} = SQN_{UE'_2} + 1$. Here, we consider the extreme case that the attacker knows that the $SQN_{UE'_i}$ has increased by 1.
- For the RHS, the two UEs are different and differ in $k$, $SUPi$ and $SQN_{UE}$.

Thus, the observational equivalence lemma claims that, the attacker cannot distinguish whether it is interacting with $UE'_1$ or $UE'_2$, even if it can capture and replay the messages from an old session attended by $UE'_1$. This implies that the attacker cannot link the AKA sessions participated by the target UE (i.e., $UE'_1$), which eliminates the possibility of UE tracing. Moreover, our modeling adopts Tamarin Prover’s default communication model for the traffic between the UE and SN, which allow the existence of active attacker, and uses the modeling of secure channels for the communication between the SN and HN following the script by Basin et al. [24].

**Modeling MAC Failure.** Our model also covers the case of MAC failure, which is necessary for the proof of privacy, as all attacks rely on this message. We also note that this captures the case that a UE decrypts the challenge sent by the HN with a wrong key, and refer the reader to Appendix A.3 for the detail.

<table>
<thead>
<tr>
<th>Point of View</th>
<th>UE</th>
<th>SN</th>
<th>HN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak agreement</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Agreement on $K_{seaf}$</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>Agreement on ID$_{SN}$</td>
<td>wa</td>
<td>NI</td>
<td>wa</td>
</tr>
<tr>
<td>Agreement on SUPi</td>
<td>wa</td>
<td>wa</td>
<td>wa</td>
</tr>
</tbody>
</table>

Table 4: The authentication and secrecy goals achieved by 5G-AKA'. We use I to denote injective agreement and NI for non-injective agreement. “wa” means the property of non-injective agreement has been implied by the lemma of weak agreement. “−” denotes that this property is violated by definition and is not desired by TS 33.501 [20].

### 6.2 Verification Results

Next, we report the results of the formal analysis w.r.t., the goals for privacy, authentication and secrecy as follows:

**Privacy.** Before the proof for 5G-AKA', we first find the paths of existing linkability attacks in 5G-AKA with our script. This step helps us to establish the confidence on the newly developed model for privacy. For the 5G-AKA' protocol, we confirm that there is no attack in all paths for the LHS and RHS in both settings (i.e., $UE'_j$ can be a 4G-AKA session or a 5G-AKA' session) on the basis of executability. Our result for the privacy goal confirms that 5G-AKA' is able to protect the privacy of users against an active attacker, which means that an active attacker cannot perform linkability attacks that invade the privacy of mobile users, and avoids the leakage of user identifiers though linking a 5G-AKA’ session with a 4G-AKA one.

**Authentication and Secrecy.** To prove the goals for secrecy and authentication, we mainly apply the lemmas provided by Basin et al. [24], and show the achieved properties in Table 4. In particular, the formal verification proves that 5G-AKA' achieves Goals 2, 3, 4 and 5 as described in Section 4.2. We use six lemmas to prove that each pair of partners obtains the
property of weak agreement, which also implies that the cor-
responding properties of non-injective agreement on their identi-
ties hold. We also prove that each pair of partners also obtains
injective agreement on $K_{\text{seaf}}$, and the confidentiality lemmas
hold on the views of every roles. We furthermore prove that
the UE and HN both obtain non-injective agreements on $ID_{SN}$
with each other, and the SN obtains non-injective agreement
on $SUPI$ with HN.

7 Conclusion

In this paper, we present 5G-AKA’ as a privacy-preserving
proposal for the AKA protocol of 5G. It is able to protect the
users’ privacy even in the presence of active attackers, which
provides stronger privacy guarantee than 5G-AKA [20].

Our approach is compatible with 3GPP’s specifications for
5G network and legacy SIM cards, which makes it suitable
to be standardized and deployed in practice. The migration
from 5G-AKA to 5G-AKA’ does not raise extra bandwidth
cost, only involves limited additional time costs, and may only
require software modifications on both endpoints (i.e., UE
and HN). The compatibility with legacy SIM cards enables
it to be deployed in a way of reusing 3G and 4G SIM cards,
which may adjust to mobile communication providers’ inter-
est. The standardization of 5G-AKA’ may include minimal
modifications on TS 33.501. This brings another advantage
that the already deployed SNs’ implementations do not need
to be changed due to the mitigation to 5G-AKA’.

Acknowledgments

The authors would like to thank the anonymous review-
ers of USENIX Security 2020 and 2021 for their help-
ful comments and suggestions. This work is supported by
the National Key Research and Development Program of
China (No.2017YFB0802000, 2017YFB0802500) and the
National Natural Science Foundation of China (No.61802376,
U1536205).

References

Protocol (Release 4). Technical specification, 3rd Gener-
ation Partnership Project; Technical Specification Group

techniques – Encryption algorithms – Part 2: Asym-
metric ciphers. ISO/IEC International Standards, May
2006.

[3] SEC 1: Elliptic Curve Cryptography version 2.0. Stan-
dards for efficient cryptography, Certicom Research,
May 2009.

[4] SEC 2: Recommended Elliptic Curve Domain Param-
eters version 2.0. Standards for efficient cryptography,
Certicom Research, July 2010.

[5] TR 33.899: Study on the security aspects of the next
generation system (Release 14). Technical report, 3rd
Generation Partnership Project (3GPP); Technical Spec-
ification Group Services and System Aspects (SA3),
August 2017.

[6] China mobile says 5g networks do not require new sim
cards, increased data usage. https://technode.com
/2018/05/04/5g-china-mobile/, May 2018.

org/, July 2019.

[8] Open5gcore - the next mobile core network testbed plat-

[9] Openairinterface - 5g software alliance for democratis-
ing wireless innovation. https://www.openairinter-
face.org/, November 2019.

[10] Sim only deals - all our sims are 5g ready, at no ex-
tra cost. http://www.three.co.uk/Store/SIM-hub,
November 2019.

the symbolic model. https://tamarin-prover.git

[12] TS 31.102: Characteristics of the Universal Subscriber
Identity Module (USIM) application (Release 16). Tech-
nical specification, 3rd Generation Partnership Project;
Technical Specification Group Core Network and Term-
inals, June 2019.

[13] TS 31.102: Characteristics of the Universal Subscriber
Identity Module (USIM) application version 14.8.0 (Re-
lease 14). Technical specification, 3rd Generation Part-
nership Project; Technical Specification Group Core

[14] Vodafone 5g is here. https://www.vodafone.co.uk
/network/5g, October 2019.

(Release 16). Technical specification, 3rd Generation
Partnership Project (3GPP), July 2020.

[16] TS 24.008: Digital cellular telecommunications system
(Phase 2+) (GSM); Universal Mobile Telecommunica-
tions System (UMTS); LTE; Mobile radio interface
Layer 3 specification; Core network protocols; Stage
3 (Release-16). Technical specification, 3rd Generation
Partnership Project (3GPP), July 2020.


[26] Ravishankar Borgaonkar, Lucca Hirschi, Shinjo Park, and Altaf Shaik. New privacy threat on 3g, 4g, and upcoming 5g AKA protocols. IACR Cryptology ePrint Archive, pages 1175, 2018.


A The Modeling Choices in Detail

A.1 A Brief Introduction to Tamarin Prover

Before presenting the details of our modeling choices, we first present a brief and informal introduction to Tamarin Prover. We also refer the readers to [11] for the details. Particularly, Tamarin uses multiset rewriting rules to specify the execution of protocols, lemmas to model the desired properties, and equations over functions to model the algorithmic operations and cryptographic primitives, which are introduced as follows:

**Rule.** A rule commonly consists of three parts including a premise, a conclusion and a state for labeling the transition between the premise and conclusion, and is used to model one step of a protocol. The premise contains facts that exist in the current state of system, and the conclusion includes the set of facts that appear in the system’s next state, which models the states before and after a step of protocol execution. The intermediate state consists of action facts, which are stored by the system to indicate an execution of this step.

**Lemma.** A lemma claims the (non-)existence of a trace, which consists of the action facts that appear in rules. The order of facts can be arranged by constraints in time sequence.
It is convenient to use lemmas and action facts to model security properties such as secrecy and authentication. To prove a lemma, Tamarin Prover automatically checks all possible traces with backward-searching. It outputs the corresponding path (i.e., the attack) when it finds a violation on the lemma.

**Function and Equation.** Tamarin Prover models cryptographic primitives by functions and equation theories over functions following the so-called black-box cryptography assumption, which means that the primitives are assumed to be secure. Functions model the syntaxes of primitives, and equations model the property of functions (i.e., the functionality of cryptographic primitive). Tamarin Prover has provided a series of built-in message theories which are useful to model real-world protocols. It also allows users to define functions and equations on their own choices, which can be adopted to model primitives outside the scope of built-in theories.

### A.2 Modeling the KEM/DEM Paradigm

To model 5G-AKA faithfully, we have to manually define the asymmetric encryption algorithm (i.e., ECIES) following the KEM/DEM paradigm with functions and theories, as the built-in theory of asymmetric encryption only present a black-box style modeling, which can not describe KEM and DEM precisely. In particular, we present the modeling of a generic asymmetric encryption scheme which is designed following the KEM/DEM paradigm, rather than just model ECIES with the built-in diffie-hellman theory.

To make our modeling more clear, we first give a review on the built-in modeling of asymmetric encryption. Tamarin Prover models asymmetric encryption by aenc/2, adec/2 and pk/1, where the digit indicates the number of parameters. In particular, aenc/2 and adec/2 model encryption and decryption respectively, and pk/1 models the relationship between a private key and public key. Let sk be the private key and m be the plaintext, the built-in equation theory expresses a public-key encryption scheme as follows:

\[
\text{aenc}(\text{aenc}(m, \text{pk}(\text{sk})), \text{sk}) = m.
\]

To model ECIES with the KEM/DEM paradigm, we use four functions and one equation over these functions to define the key encapsulation/decapsulation mechanism. The functions of KEM are presented as follows:

- **encap/2**: It takes two parameters as input, which are the public key of HN and a random number. In spite that the definition of Encapsulation does not contain a random number as input, it is a random algorithm where different runs output different results. Thus, we require the encap function to take a freshly chosen random number as input. Otherwise, Tamarin Prover will treat encap as a deterministic algorithm. This measure has also been applied by Basin et al. [24] for the modeling of ECIES in 5G-AKA.
- **getkey/1**: It takes one parameter as input (i.e., \(\text{encap}(\cdot, \cdot)\)), which outputs the shared secret key generated by the key encapsulation algorithm. This function is executed by the sender (i.e., UE) to extract the shared key.
- **getcipher/1**: This function takes one parameter as input, which is also \(\text{encap}(\cdot, \cdot)\). It outputs the ciphertext which encapsulates the shared key, and is also executed by UE.
- **decap/2**: It takes two parameters as input, which are the recipient’s (i.e., HN’s) private key and the ciphertext that encapsulates the shared key (i.e., getcipher(·)). This function is executed by the recipient to obtain the shared secret key, which models the key decapsulation algorithm.

We use an equation to model the functionality of key encapsulation and decapsulation as follows:

\[
\text{decap}(\text{sk}, \text{getcipher}(\text{encap}(\text{pk}(\text{sk}), \text{R}))) = \\
\text{getkey}(\text{encap}(\text{pk}(\text{sk}), \text{R})),
\]

where \(\text{R}\) is a freshly chosen random number, and \(\text{pk}/1\) is borrowed from the built-in asymmetric-encryption theory to model the relationship between the recipient’s private and public keys. This function guarantees that only the holder of \(\text{sk}\) can establish the same shared key with the sender. Furthermore, we also leverage Tamarin Prover’s built-in theory for symmetric encryption (i.e., senc/2 and sdec/2) to model the encapsulation and decapsulation of data. Note that we do not distinguish a DEM with its underlying symmetric encryption for the sake of simplicity. The equation for senc/2 and sdec/2 are defined as follows:

\[
\text{sdec}(\text{senc}(m, k), k) = m.
\]

### A.3 Modeling MAC Failure

Our modeling of the privacy goal covers the case of MAC failure with the standard Inequality restriction which implies that this rule can only be applied when a UE instance receives a RAND encrypting with a key different from its \(k_{UE}\). The case of MAC failure is not modeled in the scripts by [24], and we also find that the published version of Tamarin Prover 1.4.1\(^1\) is unable to handle this checking condition properly when the diff/2 operator is applied. When searching for a mirror from, e.g., LHS to RHS, Tamarin Prover always captures a mirror which violates the Inequality restriction and outputs it as an attack. In fact, there exists other mirrors that do not violate the restriction. The “attack” is detected due to the incorrect collapsing of two freshly generated terms as equal (e.g., \(\text{Fr}(\neg x)\) and \(\text{Fr}(\neg y)\)). Fortunately, this bug has been reported to the Tamarin Prover team and fixed by commit c3c3cec\(^2\). To model the Mac_Failure message properly, we use a developing version of Tamarin Prover that has applied that commit.

---

\(^1\)https://tamarin-prover.github.io/

\(^2\)https://github.com/tamarin-prover/tamarin-prover/issues/331