information

Article

5GAKA-LCCO: A Secure 5G Authentication and Key Agreement Protocol with Less Communication and Computation Overhead

Yuelei Xiao [1,2,*] and Shan Gao [3]

[1] School of Modern Posts, Xi'an University of Post and Telecommunications, Xi'an 710061, China
[2] Shaanxi Provincial Information Engineering Research Institute, Xi'an 710075, China
[3] School of Computer, Xi'an University of Post and Telecommunications, Xi'an 710061, China; gaoshan8165@163.com
* Correspondence: xiaoyuelei@xupt.edu.cn; Tel.: +86-029-85-383-289

Abstract: There are still some shortcomings in the latest version of the 5G authentication and key agreement (AKA) protocol, which is specified by the third-generation partnership project (3GPP). To overcome these shortcomings, an improved primary authentication and key agreement protocol for 5G networks (5G-IPAKA) were proposed. However, one of the shortcomings of the 5G AKA protocol has not been completely overcome in the 5G-IPAKA protocol, resulting in denial of service (DoS) attacks against both the serving network (SN) and the home network (HN). In addition, the 5G AKA protocol has large communication and computation overhead, while the 5G-IPAKA protocol has an even larger communication and computation overhead. These will lead to a great deal of energy consumption. To solve these problems, a secure 5G authentication and key agreement protocol, with less communication and computation overhead (5GAKA-LCCO) is proposed. Then, the 5GAKA-LCCO protocol is proven secure in both the strand space model and the Scyther tool. Further discussion and comparative analysis show that the 5GAKA-LCCO protocol can completely overcome the shortcomings of the latest version of the 5G AKA protocol and is better than the recently improved 5G AKA protocols in overcoming these shortcomings. Additionally, the 5GAKA-LCCO protocol has less communication and computation overhead than the 5G AKA protocol and the recently improved 5G AKA protocols.

Keywords: authentication and key agreement (AKA); 5G AKA; 5G-IPAKA; 5GAKA-LCCO; strand space model; communication and computation overhead; energy consumption

check for updates

Citation: Xiao, Y.; Gao, S. 5GAKA-LCCO: A Secure 5G Authentication and Key Agreement Protocol with Less Communication and Computation Overhead. *Information* **2022**, *13*, 257. https://doi.org/10.3390/info13050257

Academic Editor: Lorenzo Mucchi

Received: 6 April 2022
Accepted: 14 May 2022
Published: 16 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

With the continuous popularization of 5G communication technology, in the near future, the 5G network, as an important communication infrastructure, will penetrate diverse vertical fields, such as the transportation and medical treatment industries, and will also support various information interactions between people, people and things, and things and things [1]. In the 5G network, three different primary authentication and key agreement protocols are defined in the related third-generation partnership project (3GPP) specifications [2–4], including the 5G authentication and key agreement (AKA) protocol, the improved extensible authentication protocol method for third-generation authentication and key agreement (EAP-AKA'), and the 5G extensible authentication protocol method for transport layer security (EAP-TLS). The first two protocols are based on the shared key cryptography, while the last one is based on the public key cryptography. These protocols all aim to provide mutual authentication of subscribers and networks. Currently, they are in the process of standardization.

The 5G AKA protocol [2–4] was developed directly from the evolution packet system (EPS)-AKA protocol of the long-term evolution (LTE)/4G network [3], so it inherited

certain security vulnerabilities from the EPS-AKA protocol, such as impersonation attacks, man-in-the-middle attacks (MitM), and denial of service (DoS) attacks [5–11]. In [12], the authors analyzed the 5G AKA protocol of the technical specification (TS) 33.501 v0.7.0. They discovered a protocol vulnerability that could enable an attacker to impersonate another user in a serving network (SN). Based on the Tamarin model checker [13], Basin et al. [14] investigated the security properties of the 5G AKA protocol of TS 33.501 v15.1.0, and several major issues were revealed. These issues are related to user localization, the leakage of activity, the impact of active attackers, and the presence of malicious SN while roaming. In [15], the authors pointed out that the 5G AKA protocol suffers from link–ability attacks and proposed a new authentication scheme by making use of the Diffie–Hellman key exchange algorithm to generate the session key. This scheme was successful in preventing link–ability attacks along with an MitM attack.

For the more recent 5G AKA protocol, the authors in [16] found a new attack type. They claimed that the protection mechanism of the sequence number (SQN) can be defeated under specific replay attacks due to its use of exclusive-OR (XOR) and a lack of randomness. In [17], the authors modeled all key components of the 5G AKA protocol (i.e., the user equipment, the serving network, and the home network) according to the definition in the 3GPP specification document. They discovered an attack that exploits a potential race condition and additionally showed that solving the race condition for the honest case does not necessarily prevent the attack. In [18], the authors investigated the privacy properties of the 5G AKA protocol using the Bana–Comon logic [19,20]. They discovered a novel de-synchronization attack and proved that their proposed protocol guarantees these privacy properties. In [21], the authors proposed a novel version of the 5G AKA protocol to prevent active attacks and gain resistance against malignant serving networks. Unfortunately, there is a possibility of an SN impersonation, so this scheme does not eliminate the vulnerability toward a MitM attack. Further, Gharsallah et al. [22] also attempted to launch a revised version of the 5G AKA protocol. However, their proposed protocol suffers from privacy preservation, as the device identities are clearly transmitted in the air, which leads to numerous security attacks.

As time goes on, more attacks on the 5G AKA protocol were found due to the insecure channel between different network domains in the legacy mobile network. In [23], the authors discovered an attack exploiting the subscription concealed identifier (SUCI) to track a subscriber in the 5G network, which is directly caused by the insecure air channel. To cover this issue, they proposed a secure authentication scheme by utilizing the existing public key infrastructure (PKI) mechanism. Further, they found a location-sniffing attack, which can be implemented by an attacker through inexpensive devices [24]. Similarly, they proposed a fixed scheme based on the existing PKI mechanism. In [25], the authors modeled the 5G AKA protocol by using ProVerif based on three- and four-entity models and then proposed their security consideration. Further, Mariya et al. [26] proposed an enhanced version of the authentication and key agreement protocol for the 5G system that surmounts the limitations existing in the 5G AKA protocol. Parne et al. [27] introduced a protocol that preserves the privacy of the user identity and overcomes the identified problems of the 5G AKA protocol.

Similarly, 3GPP has also been enhancing the security of the 5G AKA protocol [2–4]. However, there are still some shortcomings in the latest version of the 5G AKA protocol [28]. To overcome these shortcomings, an improved primary authentication and key agreement protocol for 5G networks (5G-IPAKA) was proposed in [28].

The main contributions of this paper are as follows:

- We point out that one of the shortcomings of the 5G AKA protocol has not been completely overcome in the 5G-IPAKA protocol. This means that DoS attacks against both the SN and the HN can be formed, resulting in a great deal of energy consumption of both the SN and the HN.

- We point out that the 5G AKA protocol has large communication and computation overhead. This makes that a lot of energy is consumed whether the authentication is successful or failed. However, the 5G-IPAKA protocol has larger communication and computation overhead than the 5G AKA protocol.
- We propose a secure 5G authentication and key agreement protocol with less communication and computation overhead (5GAKA-LCCO) from seven aspects. Then, we formally analyze the security of the 5GAKA-LCCO protocol by using both the strand space model [29–31] and the Scyther tool [32,33]. As a result, the 5GAKA-LCCO protocol is secure in both the strand space model and the Scyther tool.
- Through further discussion and comparative analysis, the 5GAKA-LCCO protocol can completely overcome the shortcomings of the latest version of the 5G AKA protocol and is better than the recently improved 5G AKA protocols in overcoming these shortcomings. In addition, the 5GAKA-LCCO protocol has less communication and computation overhead than the 5G AKA protocol and the recently improved 5G AKA protocols.

The rest of this paper is organized as follows. Section 2 provides overviews of both the 5G AKA protocol and the 5G-IPAKA protocol. In Section 3, we give our motivation for this paper. Section 4 describes our proposed 5GAKA-LCCO protocol. Section 5 provides formal verification of the 5GAKA-LCCO protocol in both the strand space model and the Scyther tool. In Section 6, we present the discussion and analysis, and conclude the paper in Section 7.

2. Overviews of Both the 5G AKA Protocol and the 5G-IPAKA Protocol

2.1. The 5G AKA Protocol

According to [2–4], the steps of the latest version of the 5G AKA protocol in the 3GPP standard version v17.4.0 of TS 33.501 are illustrated in Figure 1.

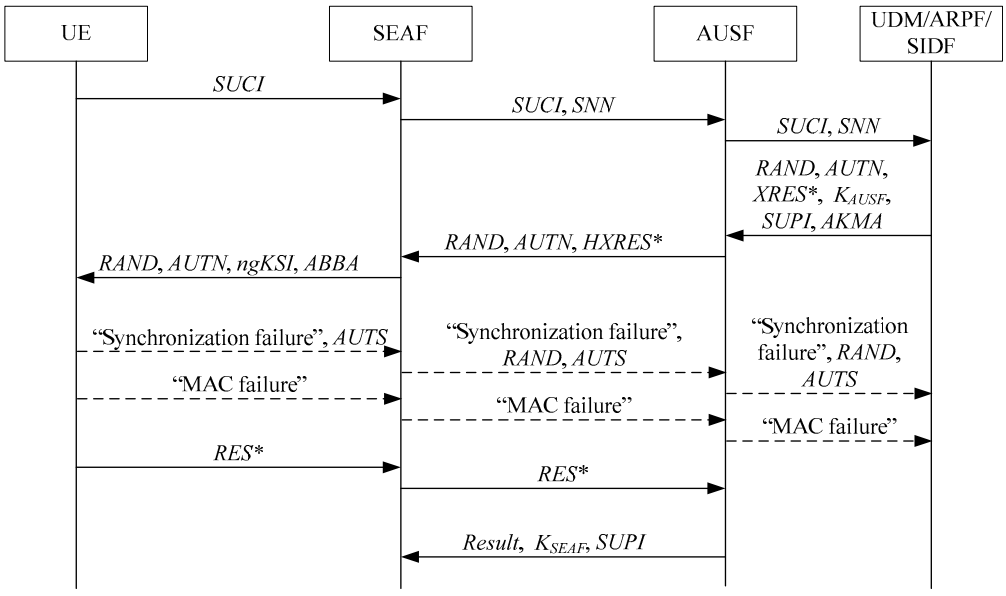

Figure 1. The steps of the latest version of the 5G AKA protocol.

In Figure 1, the universal subscriber identity module (USIM) and the mobile equipment (ME) are located in the user equipment (UE), and the security anchor function (SEAF) is located in the SN. The authentication server function (AUSF), the unified data management (UDM), the authentication credential repository and processing function (ARPE), and the subscriber identity de-concealing function (SIDF) are located in the home network (HN). The messages between the SN and the HN are usually protected. The detailed steps of the latest version of the 5G AKA protocol are as follows:

1. When the SEAF initiates authentication with the UE, the UE sends $SUCI$ to the SEAF, where the UE includes the ME and the USIM. $SUCI$ denotes a SUCI of the UE and $SUCI = x \cdot G||\{SUPI\}_{EK}||MAC_{UE}$, where $SUPI$ denotes the subscription permanent identifier (SUPI) of the UE, $x \cdot G$ and x are an ephemeral public–private key pair of the UE for Diffie–Hellman exchange, $y \cdot G$ and y are the ephemeral public–private key pair of the HN for Diffie–Hellman exchange, $EK||ICB||MK = KDF(x \cdot y \cdot G)$ and $MAC_{UE} = HMAC(MK, \{SUPI\}_{EK})$, EK is an encryption key, ICB is an initial counter block (ICB), MK is a message authentication code (MAC) key, MAC_{UE} is a MAC of the UE, $KDF()$ is a key derivation function, $HMAC()$ is a hash function for computing MAC, and $||$ denotes a concatenation.
2. Upon receiving $SUCI$, the SEAF sends $SUCI$ and SNN to the AUSF. SNN denotes the serving network name (SNN) of the SN.
3. If the SEAF is entitled to use SNN, then the AUSF stores the received SNN and sends $SUCI$ and SNN to the UDM.
4. The UDM invokes the SIDF when $SUCI$ is received. Then, the SIDF de-conceals $SUCI$ to gain $SUPI$ before the UDM can process the request. Based on $SUPI$, the UDM/ARPF chooses one authentication method.
5. When 5G AKA is selected, the UDM/ARPF generates $RAND$, calculates $AUTN$ and $XRES^*$, and derives K_{AUSF}, and then creates a 5G home environment authentication vector (5G HE AV) from $RAND$, $AUTN$, $XRES^*$, and K_{AUSF}. $RAND$ is an unpredictable challenge of the HN. $AUTN$ is an authentication token of the HN and $AUTN = SQN \oplus AK||AMF||MAC$, where SQN is a fresh sequence number generated by the HN, AK is an anonymous key and $AK = f_5(K, RAND)$, AMF is the authentication management field (AMF) and the separation bit of the AMF is set to 1, MAC is a MAC of the HN and $MAC = f_1(K, SQN||RAND||AMF)$, K is a long-term key between the UE and the HN, $f_1()$ is a message authentication function, and $f_5()$ is a key-generating function. Here, $XRES^* = KDF(CK||IK, SNN||RAND||XRES)$, where CK is a cipher key and $CK = f_3(K, RAND)$, IK is an integrity key and $IK = f_4(K, RAND)$, $XRES$ is an expected response and $XRES = f_2(K, RAND)$, $f_2()$ is a message authentication function, and $f_3()$ and $f_4()$ are two key-generating functions. K_{AUSF} is a key derived from CK and IK, and $K_{AUSF} = KDF(CK||IK, SNN||SQN \oplus AK)$.
6. The UDM sends the 5G HE AV to the AUSF together with $SUPI$. When an authentication and key management for applications (AKMA) subscription is used, the UDM also sends $AKMA$ to the AUSF. $AKMA$ denotes the AKMA indication and routing indicator.
7. The AUSF stores the received $XRES^*$ temporarily together with the received $SUPI$.
8. The AUSF generates a 5G AV from the 5G HE AV received from the UDM/ARPF by computing $HXRES^*$ from $XRES^*$, computing K_{SEAF} from K_{AUSF}, replacing $XRES^*$ with $HXRES^*$, and replacing K_{AUSF} with K_{SEAF} in the 5G HE AV, where $HXRES^* = SHA256(RAND||XRES^*)$, $K_{SEAF} = KDF(K_{AUSF}, SNN)$, and $SHA256()$ is a hash function.
9. The ASUF creates a 5G serving environment authentication vector (5G SE AV) by removing K_{SEAF} from the 5G AV, then sends the 5G SE AV (i.e., $RAND$, $AUTN$, and $HXRES^*$) to the SEAF.
10. The SEAF stores $HXRES^*$, and then sends $RAND$, $AUTN$, $ngKSI$, and $ABBA$ to the UE. Here, $ngKSI$ is used by the UE and the access and mobility management function (AMF) to identify the K_{AMF} and the partial native security context that is created if the authentication is successful. $ABBA$ denotes the anti-bidding down between architecture (ABBA) parameter.
11. In the UE, the ME forwards $RAND$ and $AUTN$ to the USIM. Upon receipt of $RAND$ and $AUTN$, the USIM first computes the anonymous key AK and retrieves the sequence number $SQN = (SQN \oplus AK) \oplus AK$. Next, the USIM computes $XMAC = f_1(K, SQN||RAND||AMF)$ and compares this with MAC, which is included in $AUTN$. Then, the USIM verifies that the received SQN is in the correct range.

If $XMAC$ is the same as MAC and SQN is in the correct range, then the USIM computes a response $RES = f_2(K, RAND)$, CK and IK, and then returns RES, CK, and IK to the ME. The ME then computes $RES^* = KDF(CK||IK, SNN||RAND||RES)$, K_{AUSF} and K_{SEAF}.

12. The UE sends RES^* to the SEAF.
13. The SEAF computes $HRES^* = SHA256(RAND||RES^*)$ and compares this with $HXRES^*$. If they coincide, then the SEAF considers that the authentication is successful from the serving network point of view; if not, then the SEAF considers that the authentication is unsuccessful.
14. The SEAF sends the received RES^* to the AUSF.
15. The AUSF compares the received RES^* with the stored $XRES^*$. If RES^* and $XRES^*$ are equal, then the AUSF considers that the authentication is successful from the home network point of view. Then, the AUSF informs the UDM of the authentication result.
16. The AUSF indicates to the SEAF whether the authentication was successful or not from the home network point of view (i.e., *Result*). If the authentication was successful, then the ASUF also sends K_{SEAF} and $SUPI$ to the SEAF.

In step 11, if $XMAC$ and MAC are different, then the USIM indicates to the ME a MAC failure of $AUTN$. Then, the UE sends a "MAC failure" indication to the SEAF. Further, the SEAF sends the "MAC failure" indication to the AUSF. Finally, the ASUF sends the "MAC failure" indication to the UDM/ARPF.

In step 11, if SQN is not in the correct range, then the USIM computes $AUTS = SQN_{UE} \oplus AK^*||MAC - S$, and sends $AUTS$ with a "Synchronization failure" indication to the ME, where SQN_{UE} denotes the highest sequence number the USIM has accepted, $AK^* = f_5^*(K, RAND)$, $MAC - S = f_1^*(K, SQN_{UE}||RAND||AMF_0)$, AMF_0 is a dummy value of all zeros, $f_1^*()$ is a message authentication function, and $f_5^*()$ is a key-generating function. Then, the UE sends $AUTS$ with a "Synchronization failure" indication to the SEAF. Further, the SEAF sends $RAND$ and $AUTS$ with a "Synchronization failure" indication to the AUSF. Finally, the ASUF sends $RAND$ and $AUTS$ with a "Synchronization failure" indication to the UDM/ARPF.

According to the analysis of the above 5G AKA protocol, there are still some shortcomings in the latest version of the 5G AKA protocol [28], as follows:

- $SUCI$ can be replayed without being found. The HN cannot find whether $SUCI$ is a replayed message because $SUCI$ does not contain the challenge of the HN. Similarly, the UE cannot find whether $SUCI$ is a replayed message because $AUTN$ does not contain the challenge of the UE (i.e., x), which is included in $SUCI$ generated by the UE.
- Mutual authentication between the UE and the SN cannot be established. The UE cannot authenticate the SN because $AUTN$ does not contain SNN. Similarly, the SN cannot authenticate the UE for the following three reasons. Firstly, the SN does not verify $SUCI$, $AUTN$, $HXRES^*$, RES^*, and $AUTS$. Secondly, the second received message of the SN does not contain $SUPI$ to match with $SUCI$ in the first received message of the SN. Finally, the last received message of the SN does not contain $RAND$, meaning that $SUPI$ in the last received message of the SN cannot match the UE's identity in $AUTN$ and $HXRES^*$, which are included in the second received message of the SN.
- K_{SEAF} cannot reach an agreement. The last received message of the SN does not contain $RAND$, so this message can be a replayed message, meaning that K_{SEAF} on the SN is not equal to K_{SEAF} on the HN. As a result, K_{SEAF} on the SN is also not equal to K_{SEAF} on the UE.
- Location privacy of the UE can be compromised. Because $AUTN$ does not contain the challenge of the UE (i.e., x), the first received message of the UE can be a replayed message. If $SQN \subset AUTN$ is in the correct range, then the location of the UE can be compromised by reidentifying RES^*. If $SQN \subset AUTN$ is not in the correct range, then the location privacy of the UE can be compromised by identifying the "Synchronization

failure" indication. That is to say, when the first received message of the UE is replayed, the legitimate UE responds to RES^* or a "Synchronization failure" indication, but any other UE responds to a "MAC failure" indication. As a result, the location privacy of the legitimate UE can be compromised.

- DoS attacks against the SN can be formed. Because the received messages of the SN do not contain the challenge of the SN, these messages can be some replayed messages. As a result, the penetrator can impersonate both the UE and the HN to complete the entire 5G AKA protocol with the SN, forming DoS attacks against the SN.
- Attacks based on MAC failure can be performed. Firstly, the penetrator can forge or tamper with the first received message of the UE to make the UE respond to a "MAC failure" indication, resulting in authentication failure. Secondly, the penetrator can directly send a "MAC failure" indication to the SN, causing authentication failure. Finally, the penetrator can also replay a "MAC failure" indication between the SN and the HN, causing authentication failure.
- Perfect forward secrecy cannot be provided. In the latest version of the 5G AKA protocol, if K is leaked, then the penetrator can calculate K_{AUSF} and K_{SEAF} based on those messages transmitted in the past run of the protocol. As a result, the penetrator can decrypt those encrypted communication messages transmitted in the past run of the protocol. Therefore, the latest version of the 5G AKA protocol cannot provide perfect forward secrecy.

2.2. The 5G-IPAKA Protocol

In order to overcome the above shortcomings of the latest version of the 5G AKA protocol, we proposed the 5G-IPAKA protocol in [28], which is illustrated in Figure 2.

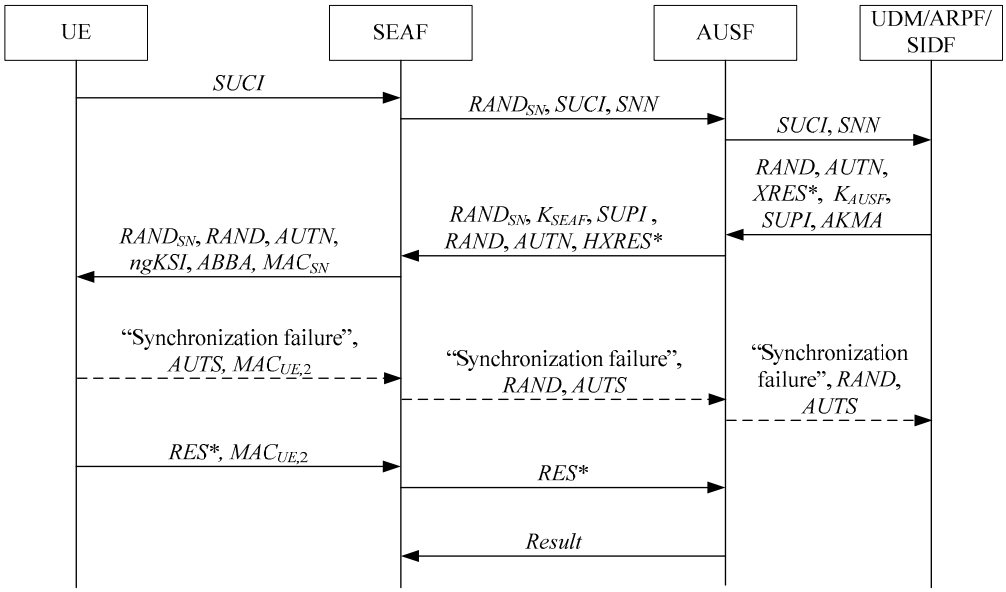

Figure 2. The 5G-IPAKA protocol.

In Figure 2, the detail steps of the 5G-IPAKA protocol are as follows:

1. When the SEAF initiates an authentication with the UE, the UE sends $SUCI$ to the SEAF.
2. Upon receiving $SUCI$, the SEAF generates $RAND_{SN}$ and then sends $RAND_{SN}$, $SUCI$, and SNN to the AUSF, where $RAND_{SN}$ is an unpredictable challenge of the SEAF.
3. If the SEAF is entitled to use SNN, then the AUSF stores the received SNN and sends $SUCI$ and SNN to the UD.

4. The UDM invokes the SIDF when $SUCI$ is received. Then, the SIDF de-conceals $SUCI$ to gain $SUPI$ before the UDM can process the request. Based on $SUPI$, the UDM/ARPF chooses one authentication method.
5. When 5G-IPAKA is selected, the UDM/ARPF generates $RAND$, calculates $AUTN$ and $XRES^*$, and derives K_{AUSF}, and then creates a 5G HE AV from $RAND$, $AUTN$, $XRES^*$, and K_{AUSF}, where $AUTN = SQN \oplus AK||AMF||MAC$, $AK = f_5(BK, RAND)$, $MAC = f_1(BK, SQN||RAND||AMF)$, $CK = f_3(BK, RAND)$, $IK = f_4(BK, RAND)$, $XRES = f_2(BK, RAND)$, $XRES^* = KDF(CK||IK, SNN||RAND||XRES)$, $K_{AUSF} = KDF(CK||IK, SNN||SQN \oplus AK)$, and $BK = KDF(K, x \cdot y \cdot G||SNN)$.
6. The UDM sends the 5G HE AV to the AUSF together with $SUPI$. When an AKMA subscription is used, the UDM also sends $AKMA$ to the AUSF.
7. The AUSF stores the $XRES^*$ temporarily together with the received $SUPI$.
8. The AUSF generates a 5G AV from the 5G HE AV received from the UDM/ARPF by computing $HXRES^*$ from $XRES^*$, computing K_{SEAF} from K_{AUSF}, replacing $XRES^*$ with $HXRES^*$, and replacing K_{AUSF} with K_{SEAF} in the 5G HE AV.
9. The ASUF creates a 5G SE AV by adding $SUPI$ to the 5G AV, then sends the 5G SE AV (i.e., $RAND$, $AUTN$, $HXRES^*$, K_{SEAF}, and $SUPI$) together with $RAND_{SN}$ to the SEAF.
10. The SEAF stores $HXRES^*$, computes MAC_{SN}, and then sends $RAND_{SN}$, $RAND$, $AUTN$, $ngKSI$, $ABBA$, and MAC_{SN} to the UE, where MAC_{SN} is a MAC of the SEAF and $MAC_{SN} = HMAC(K_{SEAF}, RAND_{SN}||RAND||AUTN||ngKSI||ABBA)$.
11. In the UE, the ME forwards $RAND$ and $AUTN$ to the USIM. Upon receipt of $RAND$ and $AUTN$, the USIM first computes $BK = KDF(K, x \cdot y \cdot G||SNN)$ and the anonymous key $AK = f_5(BK, RAND)$ and retrieves the sequence number $SQN = (SQN \oplus AK) \oplus AK$. Next, the USIM computes $XMAC = f_1(BK, SQN||RAND||AMF)$ and compares this with MAC that is included in $AUTN$. Then, the USIM verifies that the received SQN is in the correct range. If $XMAC$ is the same as MAC and SQN is in the correct range, then the USIM computes a response $RES = f_2(BK, RAND)$, $CK = f_3(BK, RAND)$, and $IK = f_4(BK, RAND)$, and then returns RES, CK, and IK to the ME. The ME then computes $RES^* = KDF(CK||IK, SNN||RAND||RES)$, K_{AUSF}, and K_{SEAF}. Finally, the ME verifies MAC_{SN} using K_{SEAF}. If the verification fails, then the ME aborts.
12. The UE computes $MAC_{UE,2}$, and then sends RES^* and $MAC_{UE,2}$ to the SEAF, where $MAC_{UE,2} = HMAC(K_{SEAF}, RAND_{SN}||RES^*)$ is another MAC of the UE.
13. The SEAF verifies $MAC_{UE,2}$. If the verification fails, then the SEAF aborts. Otherwise, the SEAF computes $HRES^* = SHA256(RAND||RES^*)$ and compares this with $HXRES^*$. If they coincide, then the SEAF considers that the authentication is successful from the serving network point of view. If not, then the SEAF considers that the authentication is unsuccessful.
14. The SEAF sends the received RES^* to the AUSF.
15. The AUSF compares the received RES^* with the stored $XRES^*$. If RES^* and $XRES^*$ are equal, then the AUSF considers that the authentication is successful from the home network point of view. Then, the AUSF informs the UDM of the authentication result.
16. The AUSF indicates to the SEAF whether the authentication was successful or not from the home network point of view (i.e., *Result*).

In step 11, if $XMAC$ and MAC are different, then the UE directly discards the first received message of the UE without responding to a "MAC failure" indication, so the HN will initiate a new authentication procedure towards the UE when the HN does not receive an authentication response message or a synchronization failure message within a certain period of time.

In step 11, if SQN is not in the correct range, then the USIM computes $AUTS = SQN_{UE} \oplus AK^*||MAC - S$, and then sends $AUTS$ with a "Synchronization failure" indication to the ME, where $AK^* = f_5^*(BK, RAND)$ and $MAC - S = f_1^*(BK, SQN_{UE}||RAND||AMF_0)$. Then, the ME computes $MAC_{UE,2} = HMAC(K_{SEAF}, RAND_{SN}||Syncf||AUTS)$, and then

sends $AUTS$ and $MAC_{UE,2}$ with a "Synchronization failure" indication to the SEAF, where $Syncf$ = "Synchronization failure". Further, the SEAF verifies $MAC_{UE,2}$. If the verification fails, then the SEAF aborts, otherwise the SEAF sends $RAND$ and $AUTS$ with a "Synchronization failure" indication to the AUSF. Finally, the ASUF sends $RAND$ and $AUTS$ with a "Synchronization failure" indication to the UDM/ARPF.

Compared with the latest version of the 5G AKA protocol, the main improvements of the 5G-IPAKA protocol are as follows:

- Replace the pre-shared key between the UE and the HN with a derivation key of the pre-shared key. In detail, K is replaced with $BK = KDF(K, x \cdot y \cdot G||SNN)$ on both the UE and the HN.
- Add the challenge-response mechanism for the SN. Firstly, $RAND_{SN}$ is added to the first sent message of the SEAF as a challenge and is added to the second received message of the SEAF as a response. Then, $RAND_{SN}$ is added to the second sent message of the SEAF as a challenge and is added to the third received message of the SEAF as a response (i.e., $RAND_{SN}$ in $MAC_{UE,2}$).
- Add the mutual authentication and key confirmation between the UE and the SN. Firstly, K_{SEAF} and $SUPI$ are moved to the second sent message of the AUSF. Then, the UE and the SN perform a mutual authentication and key confirmation process based on MAC_{SN} and $MAC_{UE,2}$, which are generated by using K_{SEAF}.
- Replace the MAC failure procedure with the timeout mechanism on the HN. If $XMAC$ and MAC are different, then the UE directly discards the first received message of the UE without responding to a "MAC failure" indication, so the HN will initiate a new authentication procedure towards the UE when the HN does not receive an authentication response message or a synchronization failure message within a certain period of time.

3. Motivation

In [28], the 5G-IPAKA protocol was proven secure in the mixed strand space model [29–31]. However, the above first shortcoming of the 5G AKA protocol has not been completely overcome. In the 5G-IPAKA protocol, whether $SUCI$ is replayed can be found, but only the UE can find whether $SUCI$ is replayed, while both the SN and the HN cannot find whether $SUCI$ is replayed. This will lead to DoS attacks against both the SN and the HN, as shown in Figures 3 and 4.

In Figure 3, the penetrator P replays a large amount of messages to the SEAF, which include $SUCI$, $SUCI'$, etc., for different UEs. Then, the SEAF, AUSF, UDM, ARPF, and SIDF must respond if these UEs have not been authenticated, and the penetrator discards the response messages of the SEAF. As a result, DoS attacks against both the SN and the HN are formed, resulting in a great deal of energy of both the SN and the HN being consumed.

In Figure 4, the penetrator P replays a large amount of messages to the AUSF, which include $SUCI$, $SUCI'$, etc., for different UEs. Then, the AUSF, UDM, ARPF, and SIDF must respond if these UEs have not been authenticated, and the penetrator discards the response messages of the AUSF. As a result, Dos attacks against the HN are formed, resulting in a great deal of energy of the HN being consumed.

Additionally, the 5G AKA protocol has large communication and computation overhead. As a result, whether the authentication is successful or failed, this will lead to a great deal of energy consumption. Compared with the 5G AKA protocol, the 5G-IPAKA protocol adds some calculations and fields, so it has larger communication and computation overhead than the 5G AKA protocol.

Therefore, it is necessary to propose a novel 5G AKA protocol, which can completely overcome the above shortcomings of the 5G AKA protocol and has less communication and computation overhead than the 5G AKA protocol.

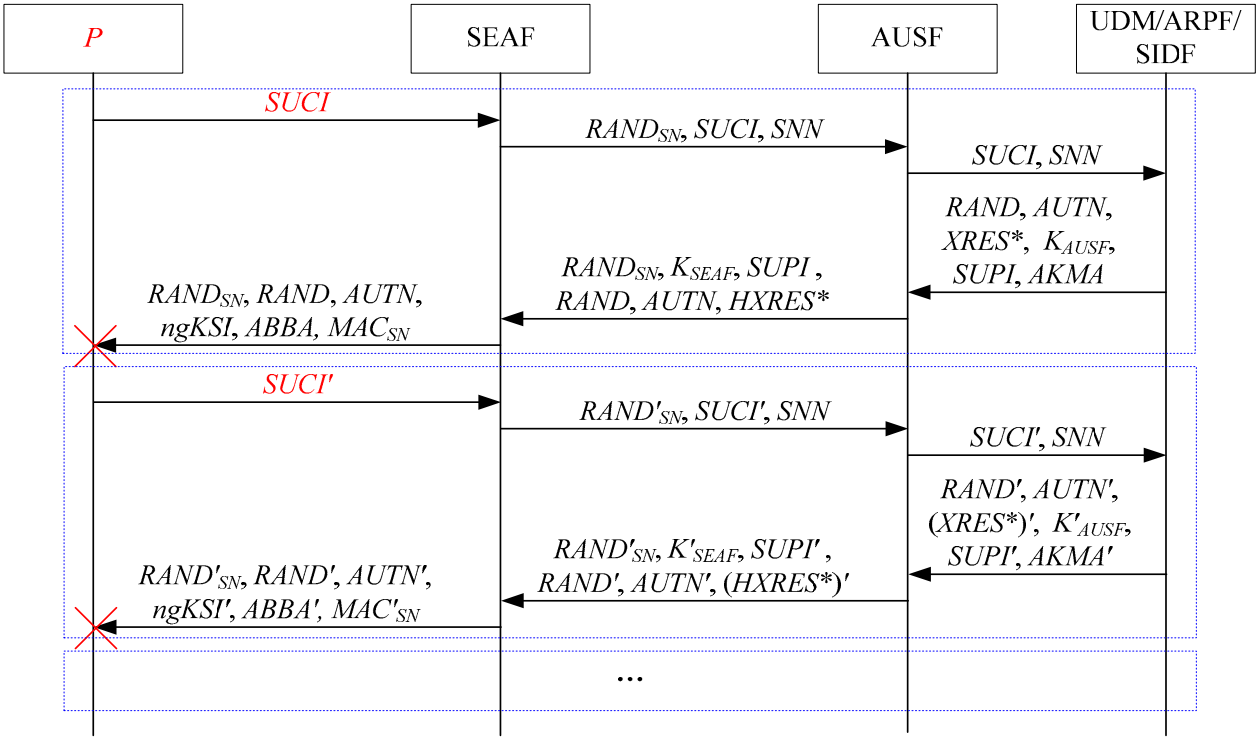

Figure 3. DoS attacks against both the SN and the HN for the 5G-IPAKA protocol.

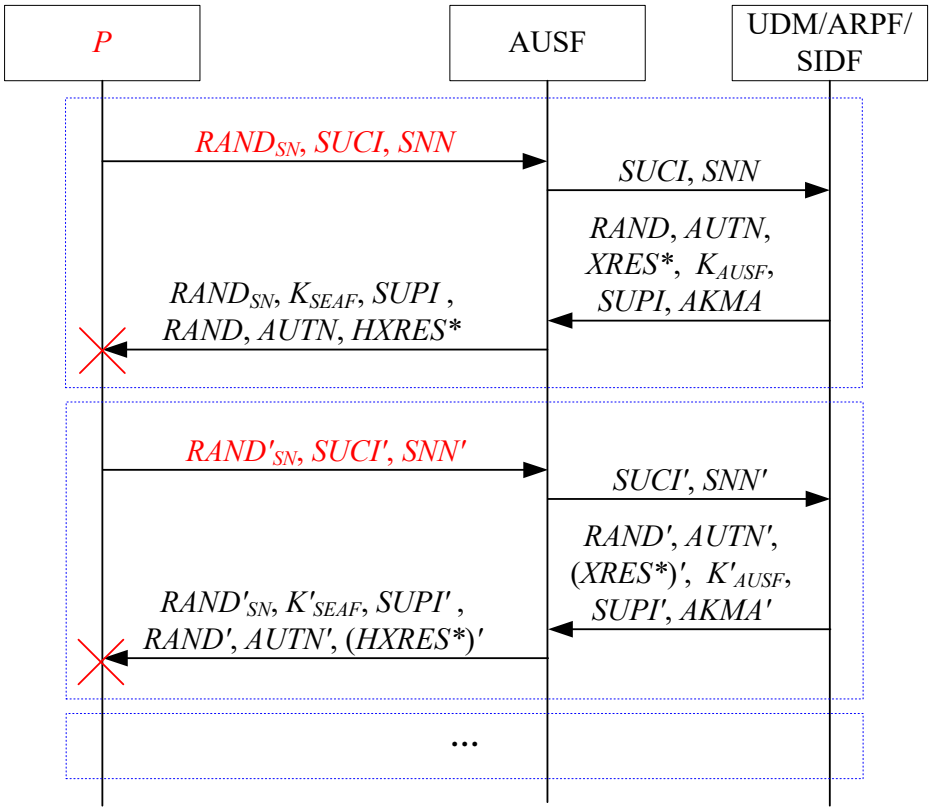

Figure 4. DoS attacks against the HN for the 5G-IPAKA protocol.

4. Our Proposed 5GAKA-LCCO Protocol

According to the above motivation, we propose a 5GAKA-LCCO protocol, which is illustrated in Figure 5.

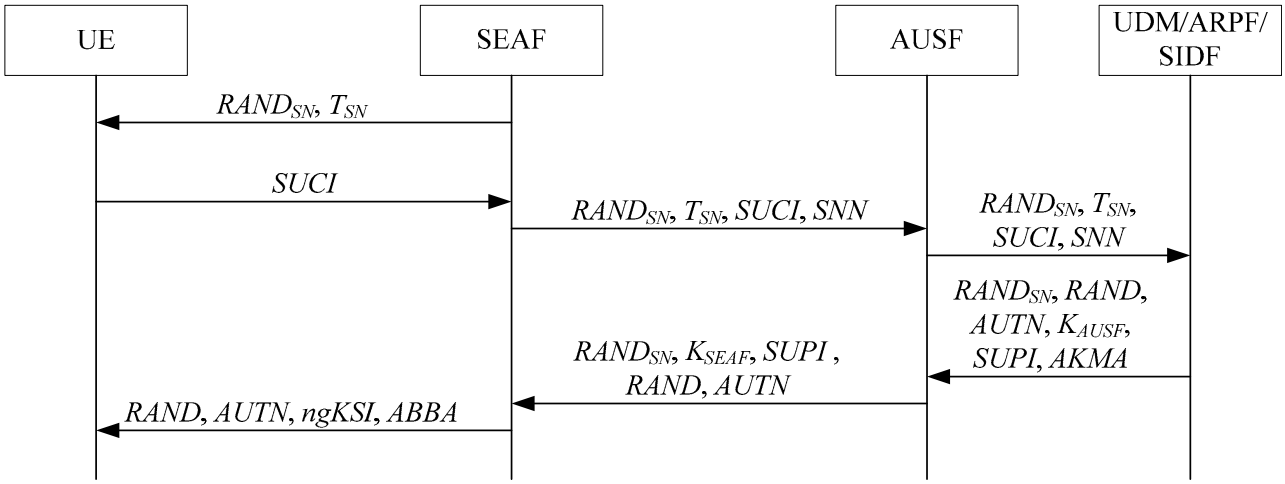

Figure 5. Our proposed 5GAKA-LCCO protocol.

In Figure 5, the detail steps of the 5GAKA-LCCO protocol are as follows:

1. When the SEAF initiates an authentication with the UE, the SEAF generates $RAND_{SN}$ and T_{SN}, and then sends $RAND_{SN}$ and T_{SN} to the UE, where T_{SN} is a timestamp generated by the SEAF.
2. Upon receiving $RAND_{SN}$ and T_{SN}, the UE sends $SUCI = x \cdot G||\{SUPI\}_{EK}||MAC_{UE}$ to the SEAF, where $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$ and $MAC_{UE} = HMAC(MK, \{SUPI\}_{EK})$. Note that the time synchronization only needs to be maintained between the SN and the HN in the 5GAKA-LCCO protocol, so the UE does not verify T_{SN}.
3. Upon receiving $SUCI$, the SEAF sends $RAND_{SN}$, T_{SN}, $SUCI$, and SNN to the AUSF.
4. If the SEAF is entitled to use SNN and T_{SN} is valid, then the AUSF stores the received SNN, and sends $RAND_{SN}$, T_{SN}, $SUCI$, and SNN to the UDM. Otherwise, the AUSF aborts. If $|T_{SN} - T_{AUSF}| < \Delta t_{AUSF}$, then T_{SN} is valid, where T_{AUSF} is the current time of the AUSF and Δt_{AUSF} is the time difference set by the AUSF.
5. The UDM first verifies T_{SN}. If T_{SN} is invalid, then the UDM/ARPF aborts. Otherwise, the UDM invokes the SIDF when $SUCI$ is received. Then, the SIDF de-conceals $SUCI$ to gain $SUPI$ before the UDM can process the request. After the de-concealing process, the SIDF sends MK, K_{AUSF}, and $SUPI$ to the UDM. If $|T_{SN} - T_{UDM}| < \Delta t_{UDM}$, then T_{SN} is valid, where T_{UDM} is the current time of the UDM and Δt_{UDM} is the time difference set by the UDM.
6. The UDM/ARPF generates $RAND$ and calculates $AUTN$, where $AUTN = AMF||MAC$ and $MAC = f_1(MK, RAND||AMF)$.
7. The UDM sends $RAND_{SN}$, $RAND$, $AUTN$, K_{AUSF}, and $SUPI$ to the AUSF. When an AKMA subscription is used, the UDM also sends $AKMA$ to the AUSF.
8. The AUSF calculates K_{SEAF} from K_{AUSF}, then sends $RAND_{SN}$, K_{SEAF}, $SUPIRAND$, and $AUTN$ to the SEAF, where $K_{SEAF} = KDF(K_{AUSF}, SNN)$.
9. The SEAF stores K_{SEAF}, and then sends $RAND$, $AUTN$, $ngKSI$, and $ABBA$ to the UE.
10. The UE verifies $AUTN$ based on MK. If the verification is successful, then the UE calculates K_{SEAF} from K_{AUSF}, and stores K_{SEAF}. Otherwise, the UE aborts.

Compared with the latest version of the 5G AKA protocol, the main improvements of our proposed 5GAKA-LCCO protocol are as follows:

- Modify the key derivation process of the pre-shared key. In detail, $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$, where EK and ICB are used to encrypt $SUPI$, MK is used to calculate MAC_{UE} and MAC, and K_{AUSF} is used to derivate K_{SEAF}.

- Add the challenge-response mechanism for the UE. x is included in $SUCI$ of the first sent message of the UE as a challenge, and x is added to $AUTN$ of the third received message of the UE as a response.
- Add the challenge-response mechanism for the SN. Firstly, $RAND_{SN}$ is added to the first sent message of the SEAF as a challenge and $RAND_{SN}$ is added to $SUCI$ of the first received message of the SEAF as a response. Then, $RAND_{SN}$ is added to the second sent message of the SEAF as a challenge and $RAND_{SN}$ is added to the second received message of the SEAF as a response.
- Add the timestamp mechanism for the SN and the HN. T_{SN} is added to the first four messages of the protocol, but T_{SN} is only verified by the AUSF and the UDM. To verify T_{SN}, time synchronization between the SN and the HN needs to be maintained.
- Remove the synchronization failure procedure. Earlier, SQNs were used in the 5G AKA protocol because strong random number generation was not possible in the USIM, but in the current generation, this is not an issue anymore [21]. Additionally, the SQN concealment mechanism is not sufficiently protected, leading to leakage of SQNs and thus allowing activity monitoring attacks [21]. Hence, we remove SQN from $AUTN$ and use $RAND$ alone.
- Replace the MAC failure procedure with a timeout mechanism on the HN. If $XMAC$ and MAC are different, then the UE directly discards the second received message without responding to a "MAC failure" indication, so the HN will initiate a new authentication procedure towards the UE when the HN does not receive an authentication response message or a synchronization failure message within a certain period of time.
- Reduce the communication and computation overhead of the authentication process. Firstly, the first sent message of the SEAF (including $RAND_{SN}$ and T_{SN}) is added, and the authentication of the UE is advanced to the verification of $SUCI$. Secondly, K_{SEAF} and $SUPI$ are moved to the second sent message of the AUSF. Finally, after receiving the $AUTN$, the UE will no longer respond to the SEAF. This reduces the number of messages in the authentication process, as well as the communication and computation overhead.

Note that the timestamp mechanism for the SN and the HN is added to our proposed 5GAKA-LCCO protocol for the following reasons:

- The timestamp mechanism for the SN and the HN can overcome DoS attacks against the HN. This is because the first received message of the HN in the proposed 5GAKA-LCCO protocol cannot be replayed because T_{SN} is included in this message.
- According to [2–4], the SEAF initiates an authentication with the UE during any procedure for establishing a signaling connection with the UE, according to SEAF's policy. If the random number mechanism is used to overcome DoS attacks against the HN, then the first received message of the HN must be sent from the SEAF, and the HN responds to the SEAF with a random number. However, the first received message of the HN can be replayed, which means that DoS attacks against the HN still cannot be overcome.

5. Formal Verification of the 5GAKA-LCCO Protocol

To simplify the formal verification of the 5GAKA-LCCO protocol, we assume that:

- The parties of the 5GAKA-LCCO protocol shown in Figure 5 are simplified as the UE, SN, and HN.
- There is a session key $K_{SN,HN}$ between the SN and the HN, and it is secure.
- $ngKSI$ and $ABBA$ do not affect the security of the 5G AKA protocol, so they are ignored here.

According to these assumptions, the 5GAKA-LCCO protocol shown in Figure 5 is simplified as follows:

1. $SN \rightarrow UE$: $RAND_{SN}||T_{SN}$.
2. $UE \rightarrow SN$: $SUCI$.
3. $SN \rightarrow HN$: $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}}$.
4. $HN \rightarrow SN$: $\{RAND_{SN}||K_{SEAF}||SUPI||RAND||AUTN\}_{K_{SN,HN}}$.
5. $SN \rightarrow UE$: $RAND||AUTN$.

In this section, in order to evaluate the security of the 5GAKA-LCCO protocol, we mainly employ two formal verification methods, including security proof by using the strand space model [29,30] and security simulation by the use of the Scyther tool [32,33]. Moreover, we choose the Dolev–Yao attacker model to check the security of the 5GAKA-LCCO protocol. In this attacker model, the attacker can completely control the network and conduct a series of attacks.

5.1. Security Proof Based on the Strand Space Model

The strand space model [29,30] is a well-studied formal analysis method for security protocols. Therefore, we use the strand space model to analyze the security of our proposed 5GAKA-LCCO protocol as follows.

Definition 1. *An infiltrated strand space $\sum, \mathcal{P}$ is a space for the 5GAKA-LCCO protocol if it is the union of four kinds of strands: (1) Initiator strands $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND, AUTN]$ with trace: $< -RAND_{SN}||T_{SN}, +SUCI, -RAND||AUTN >$. The principal associated with this strand is UE; (2) Responder strands $r \in$ Resp$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, H_1, K_{SEAF}, SUPI]$ with trace: $< +RAND_{SN}||T_{SN}, -SUCI, +\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}}, -\{RAND_{SN}||K_{SEAF}||SUPI||RAND||H_1\}_{K_{SN,HN}}, +RAND||H_1 >$. The principal associated with this strand is SN. H_1 is one message that is not inspected by SN; (3) Server strands $t \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$ with trace: $< -\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}}, +\{RAND_{SN}||K_{SEAF}||SUPI||RAND||AUTN\}_{K_{SN,HN}} >$. The principal associated with this strand is HN; (4) Penetrator strands $p \in \mathcal{P}$[29,30].*

Theorem 1. *Suppose: (1) $\sum$ is a space for the 5GAKA-LCCO protocol, and $\mathcal{C}$ is a bundle containing an initiator strand $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND, AUTN]$; (2) $K \notin \mathcal{K}_P$ and $K_{SN,HN} \notin \mathcal{K}_P$; (3) x, $RAND$ and $RAND_{SN}$ are uniquely originating in $\sum$. Then, $\mathcal{C}$ contains a unique server strand $t \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$ and a unique responder strand $r \in$ Resp$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$.*

Proof of Theorem 1. Since $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$, $MK \notin \mathcal{K}_P$ according to assumption (2). $MAC = f_1(MK, RAND||AMF)$, and $RAND$ is uniquely originating in $\sum$, so $MAC \subset AUTN \subset term(< s, 3 >)$ must uniquely originate on a server strand $t \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$. Since $RAND_{SN}$ is uniquely originating in $\sum$, $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}} = term(< t, 1 >)$ must uniquely originate on a responder strand $r \in$ Resp$[UE', SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND', H'_1, K'_{SEAF}, SUPI']$ according to assumption (2). Similarly, $\{RAND_{SN}||K'_{SEAF}||SUPI'||RAND'||H'_1\}_{K_{SN,HN}} = term(< r, 4 >)$ must originate on a server strand $t' \in$ Serv$[UE'', SN, HN, SUCI'', SNN, RAND_{SN}, T''_{SN}, RAND', AUTN', K'_{SEAF}, SUPI']$, where $H'_1 = AUTN'$. By assumption (2), $\{RAND_{SN}||T''_{SN}||SUCI''||SNN\}_{K_{SN,HN}} = term(< t', 1 >)$ must originate on a responder strand $r' \in$ Resp$[UE''', SN, HN, SUCI'', SNN, RAND_{SN}, T''_{SN}, RAND''', H'''_1, K'''_{SEAF}, SUPI''']$. $RAND_{SN}$ is uniquely originating in $\sum$, and $r' = r$, so $SUCI'' = SUCI$ and $T''_{SN} = T_{SN}$. According to t' and Definition 1, $UE'' = UE$, $SUPI' = SUPI$ and $K'_{SEAF} = K_{SEAF}$, so $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}} = term(< t', 1 >)$. By Definition 1, $RAND' = RAND$ and $AUTN' = AUTN$, i.e., $t' = t$. Hence, $r \in$ Resp$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$. □

According to Theorem 1, UE successfully authenticates HN and SN, and can establish an injection agreement [29,30] with them.

Theorem 2. *Suppose: (1) $\sum$ is a space for the 5GAKA-LCCO protocol, and $\mathcal{C}$ is a bundle containing a server strand $t \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$; (2) $K \notin \mathcal{K}_P$ and $K_{SN,HN} \notin \mathcal{K}_P$; (3) x, $RAND$ and $RAND_{SN}$ are uniquely originating in $\sum$. Then, $\mathcal{C}$ contains a unique initiator strand $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND, AUTN]$ and a unique responder strand $r \in$ Resp$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$.*

Proof of Theorem 2. Since $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$, $MK \notin \mathcal{K}_P$ according to assumption (2). $MAC_{UE} = HMAC(MK, \{SUPI\}_{EK})$, and x is uniquely originating in $\sum$, so $MAC_{UE} \subset SUCI \subset term(< t, 1 >)$ must uniquely originate on an initiator strand $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND', AUTN']$. Similarly, $MAC' = f_1(MK, RAND'||AMF) \subset AUTN' \subset term(< s, 3 >)$ must originate on a server strand $t' \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND', AUTN', K_{SEAF}, SUPI]$, so $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}} = term(< t', 1 >)$. By Definition 1, $RAND' = RAND$ and $AUTN' = AUTN$, i.e., $t' = t$. Hence, $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND, AUTN]$. Since $RAND_{SN}$ is uniquely originating in $\sum$, $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}} = term(< t, 1 >)$ must uniquely originate on a responder strand $r \in$ Resp$[UE'', SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND'', H''_1, K''_{SEAF}, SUPI'']$ according to assumption (2). Similarly, $\{RAND_{SN}||K''_{SEAF}||SUPI''||RAND''||H''_1\}K_{SN,HN} = term(< r, 4 >)$ must originate on a server strand $t' \in$ Serv$[UE''', SN, HN, SUCI''', SNN, RAND_{SN}, T'''_{SN}, RAND'', AUTN'', K''_{SEAF}, SUPI'']$, where $H''_1 = AUTN''$. By assumption (2), $\{RAND_{SN}||T'''_{SN}||SUCI'''||SNN\}_{K_{SN,HN}} = term(< t', 1 >)$ must originate on a responder strand $r' \in$ Resp$[UE''''', SN, HN, SUCI''', SNN, RAND_{SN}, T'''_{SN}, RAND''''', H''''_1, K''''_{SEAF}, SUPI''''']$. $RAND_{SN}$ is uniquely originating in $\sum$, and $r' = r$, so $SUCI''' = SUCI$ and $T'''_{SN} = T_{SN}$. According to t' and Definition 1, $UE''' = UE$, $SUPI'' = SUPI$ and $K''_{SEAF} = K_{SEAF}$, so $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}} = term(< t', 1 >)$. By Definition 1, $RAND'' = RAND$ and $AUTN'' = AUTN$, i.e., $t' = t$. Hence, $r \in$ Resp$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$. □

According to Theorem 2, HN successfully authenticates UE and SN, and can establish an injection agreement [29,30] with them.

Theorem 3. *Suppose: (1) $\sum$ is a space for the 5GAKA-LCCO protocol, and $\mathcal{C}$ is a bundle containing a responder strand $r \in$ Resp$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, H_1, K_{SEAF}, SUPI]$; (2) $K \notin \mathcal{K}_P$ and $K_{SN,HN} \notin \mathcal{K}_P$; (3) x, $RAND$ and $RAND_{SN}$ are uniquely originating in $\sum$. Then, $\mathcal{C}$ contains a unique server strand $t \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$ and a unique initiator strand $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND, AUTN]$.*

Proof of Theorem 3. By assumptions (2) and (3), $K_{SN,HN} \notin \mathcal{K}_P$, and $RAND$ is uniquely originating in $\sum$, so $\{RAND_{SN}||K_{SEAF}||SUPI||RAND||H_1\}_{K_{SN,HN}} = term(< r, 4 >)$ must uniquely originate on a server strand $t \in$ Serv$[UE, SN, HN, SUCI', SNN, RAND_{SN}, T'_{SN}, RAND, AUTN', K_{SEAF}, SUPI]$. Similarly, $\{RAND_{SN}||T'_{SN}||SUCI'||SNN\}_{K_{SN,HN}} = term(< t, 1 >)$ must originate on a responder strand r'. $RAND_{SN}$ is uniquely originating in $\sum$, and $r' = r$, so $T'_{SN} = T_{SN}$ and $SUCI' = SUCI$ according to assumption (1). According to t and Definition 1, $AUTN' = AUTN$. Hence, $t \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND, AUTN, K_{SEAF}, SUPI]$. Since $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$, $MK \notin \mathcal{K}_P$ according to assumption (2). $MAC_{UE} = HMAC(MK, \{SUPI\}_{EK})$, and x is uniquely originating in $\sum$, so $MAC_{UE} \subset SUCI \subset term(< t, 1 >)$ must uniquely originate on an initiator strand $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND'', AUTN'']$. Similarly, $MAC'' = f_1(MK, RAND''||AMF) \subset AUTN'' \subset term(< s, 3 >)$ must originate on

a server strand $t' \in$ Serv$[UE, SN, HN, SUCI, SNN, RAND_{SN}, T_{SN}, RAND'', AUTN'']$, $K_{SEAF}, SUPI]$, so $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}} = term(< t', 1 >)$. By Definition 1, $RAND'' = RAND$ and $AUTN'' = AUTN$, i.e., $t' = t$. Hence, $s \in$ Init$[UE, SN, HN, SUCI, RAND_{SN}, T_{SN}, RAND, AUTN]$. □

According to Theorem 3, SN successfully authenticates UE and HN, and can establish an injection agreement [29,30] with them.

From Theorems 1–3, mutual authentication between the UE and SN, mutual authentication between the UE and SN, and mutual authentication between the SN and HN are established. Additionally, an injection agreement among the UE, SN and HN is established, so replay and MitM attacks among the UE, SN, and HN are overcome according to the definition of the injection agreement [29,30].

Because MitM attacks among the UE, SN, and HN are overcome, K_{SEAF} can reach an agreement among the UE, SN, and HN. In addition, replay attacks among the UE, SN, and HN are overcome, so all the messages among the UE, SN, and HN cannot be replayed. As a result, $SUCI$ cannot be replayed, the location privacy of the UE cannot be compromised, and DoS attacks against both the SN and HN cannot be formed.

5.2. Security Simulation Based on the Scyther Tool

The Scyther tool [32,33] is a protocol formal analysis tool, which can provide explicit termination for unlimited sessions and infinite state aggregation protocols, and support multiprotocol parallel analysis. The security protocol description language (SPDL) is used in the Scyther tool to describe and analyze security protocols. It provides a set of claims to test various security goals, such as secrecy and authentication. The secret claim is applied to state confidentiality. In order to provide different degrees of authentication strength, several forms of authentication claims including Alive (i.e., aliveness), Weakagree (i.e., weak agreement), Niagree (i.e., non-injection agreement), and Nisynch (i.e., non-injection synchronization) are employed to detect replay, reflection, MitM attacks, and so on. The detailed description of formal definitions for all Scyther claims can be found in [33].

We model our proposed 5GAKA-LCCO protocol in SPDL and specify the security properties of the 5GAKA-LCCO protocol by a series of claims of Scyther, as shown in Figure 6.

From Figure 6, our proposed 5GAKA-LCCO protocol successfully makes certain all Scyther claims and there are no attacks found under the test of the Scyther tool.

According to Figure 6, $SUPI$ (i.e., UE in this figure) is secret, and K_{SEAF} (i.e., the session key established between the UE and HN and distributed from the HN to SN) is also secret. Additionally, aliveness, weak agreement, non-injection agreement, and non-injection synchronization among the UE, SN, and HN are established, so replay and MitM attacks among the UE, SN, and HN are overcome [33].

Similarly, because MitM attacks among the UE, SN, and HN are overcome, K_{SEAF} can reach an agreement among the UE, SN, and HN. In addition, replay attacks among the UE, SN, and HN are overcome, so all the messages among the UE, SN, and HN cannot be replayed. As a result, $SUCI$ cannot be replayed, the location privacy of the UE cannot be compromised, and DoS attacks against both the SN and HN cannot be formed.

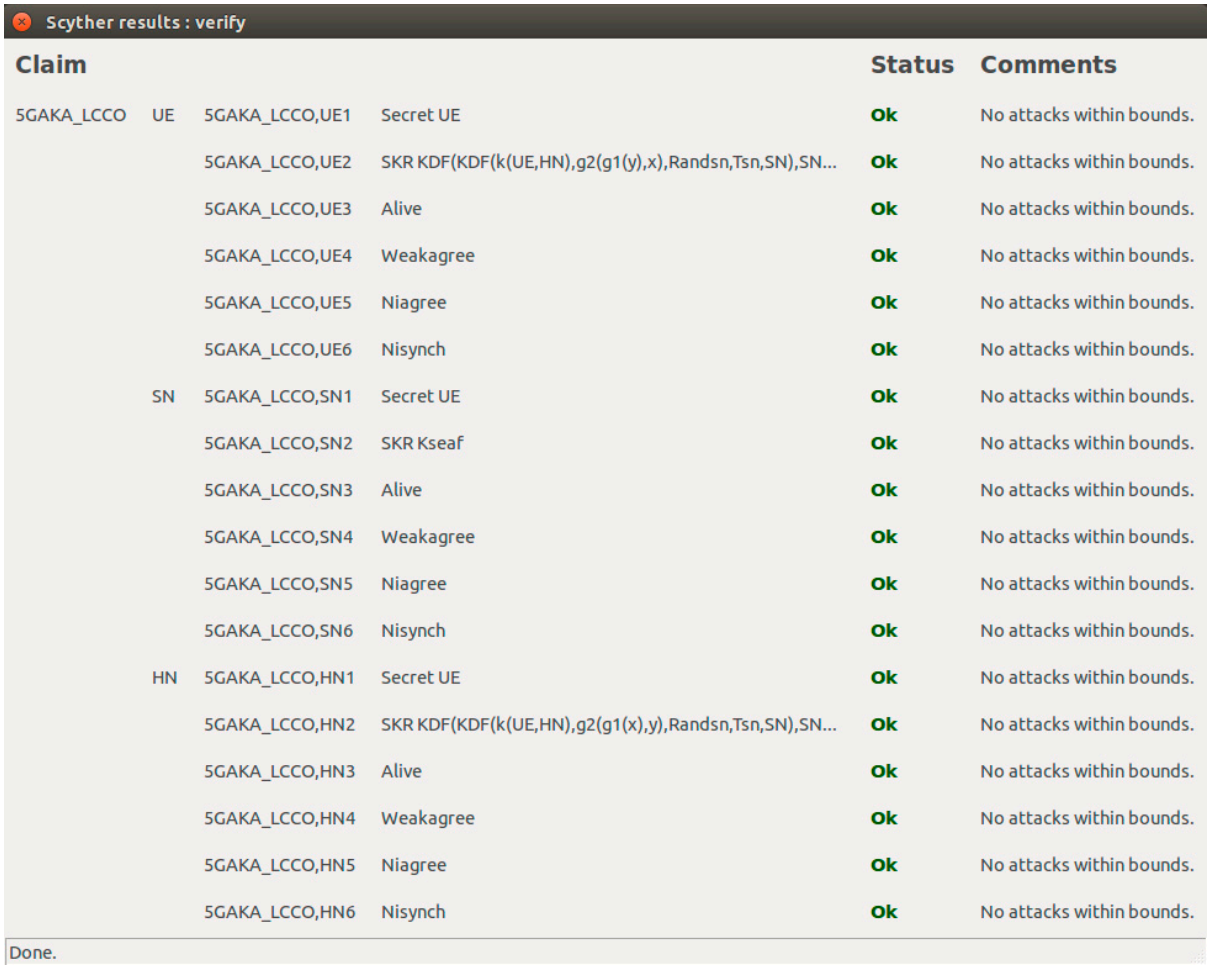

Figure 6. Security simulation results of the 5GAKA-LCCO protocol in the Scyther tool.

6. Discussion

6.1. Security of the 5GAKA-LCCO Protocol

$SUCI = x \cdot G||\{SUPI\}_{EK}||MAC_{UE}$, $MAC_{UE} = HMAC(MK, \{SUPI\}_{EK})$ and $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$, so both $RAND_{SN}$ and T_{SN} are included in $SUCI$, meaning that both the SN and the HN can find whether $SUCI$ is replayed.

$AUTN = AMF||MAC$, $MAC = f_1(MK, RAND||AMF)$ and $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$, so $AUTN$ contains the challenge of the UE (i.e., x). Hence, the second received message of the UE cannot be a replayed message, preventing the location privacy of the UE from being compromised.

Since the received messages of the SN contain the challenge of the SN (i.e., $RAND_{SN}$), these messages cannot be some replayed messages, preventing DoS attacks against the SN. In addition, T_{SN} is included in $\{RAND_{SN}||T_{SN}||SUCI||SNN\}_{K_{SN,HN}}$ and verified by the HN based on maintaining the time synchronization between the SN and HN, so the first received message of the HN cannot be replayed, preventing DoS attacks against the HN.

The 5GAKA-LCCO protocol does not contain the above "MAC failure" indication, so it can defend against those attacks based on MAC failure. In addition, $EK||ICB||MK||K_{AUSF} = KDF(K, x \cdot y \cdot G||RAND_{SN}||T_{SN}||SNN)$ and $K_{SEAF} = KDF(K_{AUSF}, SNN)$, providing perfect forward secrecy (PFS) based on the Diffie–Hellman exchange.

Depending on the above formal verification and security analysis of the 5GAKA-LCCO protocol, our proposed 5GAKA-LCCO protocol can completely overcome the above shortcomings in the latest version of the 5G AKA protocol.

A comparative analysis between the 5GAKA-LCCO protocol and the recently improved 5G AKA protocols [23,24,26,28] regarding the shortcomings of the latest version of the 5G AKA protocol is given in Table 1.

Table 1. Comparative analysis between the 5GAKA-LCCO protocol and the recently improved 5G AKA protocols [23,24,26,28] regarding the shortcomings of the latest version of the 5G AKA protocol.

Security Issues	5G AKA	[23]	[24]	[26]	[28]	5GAKA-LCCO
$SUCI$ can be replayed without being found	Yes	No	Yes	No	No	No
Mutual authentication cannot be established between the UE and the SN	Yes	Yes	Yes	Yes	No	No
K_{SEAF} cannot be agreed among the UE, he SN and the HN	Yes	Yes	Yes	Yes	No	No
The location privacy of the UE can be compromised	Yes	No	No	No	No	No
Dos attacks against the SN can be formed	Yes	Yes	Yes	Yes	Yes	No
Dos attacks against the HN can be formed	Yes	Yes	Yes	Yes	Yes	No
Attacks based on MAC failure can be performed	Yes	Yes	No	No	No	No
Perfect forward secrecy cannot be provided	Yes	Yes	Yes	Yes	No	No

From Table 1, the recently improved 5G AKA protocols still have some of the shortcomings of the latest version of the 5G AKA protocol, but our proposed 5GAKA-LCCO protocol completely overcomes all the shortcomings of the latest version of the 5G AKA protocol.

In [23], the ephemeral public–private key pair of the UE (i.e., x and $x \cdot G$), the PKI public–private key pair of the SN, and the PKI public–private key pair of the HN are used to ensure the security of the channel between the UE and the SN, the security of channel between the UE and the HN, and the security of the channel between the SN and the HN. The first received message of the UE is encrypted by using the ephemeral public key of the UE, which means that the message can only be decrypted by using the ephemeral private key of the UE, so it cannot be a replayed message, preventing the location privacy of the UE being compromised. In addition, the UE can find whether $SUCI$ is replayed. However, the other parts fully inherit the 5G AKA protocol, so the other shortcomings of the 5G AKA protocol still exist in the protocol of [23].

In [24], both the synchronization failure and the MAC failure are constructed as the format of RES^*, making it impossible to distinguish them, which can prevent the location privacy of the UE being compromised and prevent those attacks based on MAC failure. However, the other parts fully inherit the 5G AKA protocol, so the other shortcomings of the 5G AKA protocol still exist in the protocol of [24].

In [26], $SUCI$ is included in $AUTH_{SEAF}$ of the second received message of the UE, so the UE can find whether $SUCI$ is replayed, where $AUTH_{SEAF}$ is an authentication token of the SEAF. However, both the SN and the HN cannot find whether $SUCI$ is replayed, resulting in DoS attacks against both the SN and HN. Additionally, the protocol in [26] removes the synchronization failure procedure and the MAC failure procedure, preventing the location privacy of the UE from being compromised and defending against those attacks based on MAC failure. Similarly, MAC_{ARPF} is also included in $AUTH_{SEAF}$ of the second received message of the UE, but it does not contain $SEAF_{ID}$, where MAC_{ARPF} is a MAC of the ARPF and $SEAF_{ID}$ is the identity of the SEAF (i.e., SNN mentioned above). This means that the UE cannot authenticate the SN being authenticated by the HN, meaning that mutual authentication between the UE and the SN cannot be established and K_{SEAF}

cannot reach an agreement. In addition, $RAND_{SEAF}$ is included in $RAND'_{UE}$ of the second received message of the SEAF, $HXRES^*$ of the third received message of the SEAF, and RES^* of the fourth received message of the SEAF, but the SEAF does not verify these three fields, so DoS attacks against the SN can be formed, where $RAND'_{UE}$ is calculated based on $RAND_{UE}$ and $RAND_{SEAF}$ (i.e., the challenges of the UE and the SEAF, respectively). Because K_{AUSF} and K_{SEAF} can be calculated when K is leaked, PFS cannot be provided.

In [28], K is replaced with $BK = KDF(K, x \cdot y \cdot G||SNN)$ on both the UE and the HN, so $AUTN$ must contain the challenge of the UE (i.e., x), which is included in $SUCI$ generated by the UE. Hence, the UE can find whether $SUCI$ is replayed. However, both the SN and the HN cannot find whether $SUCI$ is replayed, resulting in DoS attacks against both the SN and the HN. Because the mutual authentication and injection agreement among the UE, SN, and HN are established, K_{SEAF} can reach an agreement among the UE, SN, and HN. Because $AUTN$ contains the challenge of the UE (i.e., x), the first received message of the UE (including $AUTN$) cannot be a replayed message, preventing the location privacy of the UE from being compromised. In addition, the UE directly discards the first received message without responding to a "MAC failure" indication when $XMAC$ and MAC are different, defending against those attacks based on MAC failure. Because K_{AUSF} and K_{SEAF} are generated based on BK, this provides PFS based on the Diffie–Hellman exchange.

Therefore, our proposed 5GAKA-LCCO protocol is better than the recently improved 5G AKA protocols in overcoming the shortcomings of the latest version of the 5G AKA protocol.

6.2. Communication, Computation, and Storage Overhead of the 5GAKA-LCCO Protocol

In order to evaluate the communication overhead, we will compute the transmitted message size. According to [2–4,26,27], the sizes with respect to various fields of the transmitted messages are presented in Table 2.

Table 2. The sizes with respect to numerous fields of the transmitted messages [2–4,26,27].

Fields	Size (Bits)
$K/EK/MK/IK/CK/AK/BK/K_{SN,HN}$	128
$K_{AUSF}/K_{SEAF}/K_{AMF}$	256
$SQN/SQN_{UE}/AMF$	48
$RES/RES^*/XRES/XRES^*/HRES^*/HXRES^*$	128
$MAC/XMAC/MAC_{UE}/MAC_{ARPF}/MAC_{SEAF}$	64
$Syncf/MACf/Result$	16
$RAND/RAND_{SN}/RAND_{SEAF}/RAND_{UE}/RAND'_{UE}$	128
$SNN/SUPI/SEAF_{ID}$	128
T_{SN}/T_1	64
$x/x \cdot G//y \cdot G$	256

A comparative analysis between the 5GAKA-LCCO protocol and the recently improved 5G AKA protocols [23,24,26,28] regarding the communication, computation, and storage overhead is given in Table 3.

In Table 3, the communication overhead represents the total communication overhead of the transmitted messages among the UE, SN, and HN, including the transmitted messages in both the synchronization failure procedure and the MAC failure procedure. For the 5G AKA protocol, the total communication overhead = 448 + 576 + 496 + 368 +128 + 128 + 400 + 208 + 336 + 16 + 16 = 3120 bits. For the protocol in [23], the total communication overhead = 384 + 640 + 496 + 368 + 128 + 128 + 400 + 208 + 336 + 16 + 16 = 3248 bits. For the protocol in [24], the total communication overhead = 448 + 576 + 496 + 368 + 128 + 128 + 400 + 256 + 256 + 208 + 208 = 3472 bits. For the protocol in [26], the total communication overhead = 576 + 192 + 192 + 768 + 1088 +576 + 128 + 128 + 16 = 3664 bits. For the protocol in [28], the total communication overhead = 448 + 704 + 1008 + 560 + 192 + 128 + 16 +

272 + 336 = 3664 bits. For our proposed 5GAKA-LCCO protocol, the total communication overhead = 192 + 448 + 768 + 880 + 368 = 2656 bits.

Table 3. A comparative analysis between the 5GAKA-LCCO protocol and the recently improved 5G AKA protocols [23,24,26,28] regarding the communication, computation, and storage overhead.

Protocols	Communication Overhead (Bits)	Computation Overhead	Storage Overhead (Bits)
5G AKA	3120	$1ECDH + 1ED + 12F + 2XOR$	3328
[23]	3248	$4PED + 1ED + 10F + 2XOR$	7680
[24]	3472	$2PED + 1ECDH + 1ED + 13F + 1XOR$	5504
[26]	3664	$1ECDH + 1ED + 12F$	3392
[28]	3664	$1ECDH + 1ED + 16F + 2XOR$	3328
5GAKA-LCCO	2656	$1ECDH + 1ED + 4F$	3392

In Table 3, *ECDH* denotes the generation and verification of an elliptic curve Diffie–Hellman (ECDH) exchange. *PED* denotes the generation and verification of a public key encryption and decryption process. *ED* denotes the generation and verification of a symmetric key encryption and decryption process. *F* denotes the generation and verification of a key function, a key derivation function, a MAC function, or a hash function, which are grouped into one category because they require the same amount of calculation [27]. *XOR* denotes the generation and verification of an XOR value.

The storage overhead is composed of three parts: Public parameters, identity information, and keys [34]. Hence, the storage overhead in Table 3 represents the total storage overhead of the public parameters, identity information, and keys on the UE, SN, and HN. For the 5G AKA protocol, the total storage overhead = 1408 + 512 + 1408 = 3328 bits. For the protocol in [23], the total storage overhead = 1920 + 2688 + 3072 = 7680 bits. For the protocol in [24], the total storage overhead = 2432 + 512 + 256 = 5504 bits. For the protocol in [26], the total storage overhead = 1408 + 576 + 1408 = 3392 bits. For the protocol in [28], the total storage overhead = 1408 + 512 + 1408 = 3328 bits. For our proposed 5GAKA-LCCO protocol, the total storage overhead = 1408 + 512 + 1472 = 3392 bits.

From Table 3, the communication overhead of the 5GAKA-LCCO protocol is considerably less than that of the 5G AKA protocol and the recently improved 5G AKA protocols [23,24,26,28]. According to [26], $PED > ECDH > ED > F > XOR$ in computation overhead, so the computation overhead of the 5GAKA-LCCO protocol is also lower than that of the 5G AKA protocol and the recently improved 5G AKA protocols [23,24,26,28]. Hence, our proposed 5GAKA-LCCO protocol has less communication and computation overhead than the 5G AKA protocol and the recently improved 5G AKA protocols. In addition, the storage overhead of the 5GAKA-LCCO protocol is slightly more than that of the 5G AKA protocol and the improved 5G AKA protocol in [28] and is equivalent to that of the improved 5G AKA protocol in [26]. However, the storage overhead of the 5GAKA-LCCO protocol is considerably less than that of the two improved 5G AKA protocols in [23,24].

7. Conclusions

In this paper, we provide overviews of both the latest version of the 5G AKA protocol and the 5G-IPAKA protocol, where the 5G-IPAKA protocol is a recently improved 5G AKA protocol. Then, we point out that one of the shortcomings of the 5G AKA protocol has not been completely overcome in the 5G-IPAKA protocol, leading to DoS attacks against the SN and HN. As a result, much of the energy of both the SN and HN is consumed. Additionally, the 5G AKA protocol has large communication and computation overhead. Thus, whether the authentication is successful or failed, this will lead to a great deal of energy consumption, while the 5G-IPAKA protocol has an even larger communication and computation overhead.

To solve these problems, we propose a 5GAKA-LCCO protocol. Compared with the latest version of the 5G AKA protocol, the main improvements of the 5GAKA-LCCO protocol include the fact that the key derivation process of the pre-shared key is modified, the challenge-response mechanism for the SN is added, the challenge-response mechanism for the HN is added, the timestamp mechanism for the SN and HN is added, the synchronization failure procedure is removed, the MAC failure procedure is replaced with a timeout mechanism on the HN, and the communication and computation overhead of the authentication process is reduced.

Finally, we use both the strand space model and the Scyther tool to formally analyze the security of the 5GAKA-LCCO protocol. As a result, mutual authentication and injection among the UE, SN, and HN are established. Therefore, the 5GAKA-LCCO protocol is secure in both the strand space model and the Scyther tool. Based on further discussion and comparative analysis, the 5GAKA-LCCO protocol can completely overcome the above shortcomings of the latest version of the 5G AKA protocol and is better than the recently improved 5G AKA protocols in overcoming these shortcomings. In addition, the 5GAKA-LCCO protocol has less communication and computation overhead than the 5G AKA protocol and the recently improved 5G AKA protocols.

In the above protocols, the public key cryptography mechanism, which has large computation and storage overhead, is always used. To further reduce the computation overhead, we will further improve these protocols in the future so that they do not use the public key cryptography mechanism.

Author Contributions: Methodology, Y.X.; formal analysis, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No.61741216, 61402367), Shaanxi Science and Technology Co-ordination and Innovation Project (No.2016KTTSGY01-03), National Key Research and Development Program (No. 2018YFC08242-04), and New Star Team Project of Xi'an University of Posts and Telecommunications.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Notations

$ABBA$	the ABBA parameter
AK, AK^*	two anonymous keys
$AKMA$	the AKMA indication and routing indicator
AMF	the authentication management field
AMF_0	a dummy value of all zeros
$AUTN$	an authentication token of the HN
$AUTH_{SEAF}$	an authentication token of the SEAF
$AUTS$	a resynchronization parameter
BK	a base key derived from K
CK	a cipher key
$ECDH$	the generation and verification of an ECDH exchange
ED	the generation and verification of a symmetric key encryption and decryption process
EK	an encryption key
$f_1()$, $f_1^*()$, $f_2()$	three message authentication functions
$f_3()$, $f_4()$, $f_5()$, $f_5^*()$	four key generating functions
F	the generation and verification of a key function or a key derivation function or a MAC function or a hash function

$HMAC()$	a hash function for computing MAC
H_1	one message that are not inspected by the SN
HN	the HN
$HRES^*$	a hashing response from RES^*
$HXRES^*$	a hashing expected response from $XRES^*$
ICB	an initial counter block
IK	an integrity key
K	a long-term key between the UE and the HN
K_{AMF}	a key between the UE and the access and mobility management function
K_{AUSF}	a key derived from CK and IK
$\mathcal{K}_P$	the key set of the penetrator
K_{SEAF}	a key derived from K_{AUSF}
$K_{SN,HN}$	the session key between the SN and the HN
$KDF()$	a key derivation function
MAC	a MAC of the HN
MAC_{ARPF}	a MAC of the ARPF
$MACf$	the "MAC failure" indication
MAC_{SEAF}	a MAC of the SEAF
MAC_{SN}	a MAC of the SN
MAC_{UE}	a MAC of the UE
$MAC_{UE,2}$	another MAC of the UE
MK	a MAC key
$ngKSI$	identifying the K_{AMF} and the partial native security context
PED	the generation and verification of a public key encryption and decryption process
$RAND$	an unpredictable challenge of the HN
$RAND_{SN}$, $RAND_{SEAF}$	two unpredictable challenges of the SEAF
$RAND_{UE}$	an unpredictable challenge of the UE
$RAND'_{UE}$	a challenge calculated based on $RAND_{UE}$ and $RAND_{SEAF}$
RES	a response
RES^*	a response from RES
$Result$	the authentication result
$SEAF_{ID}$	the identity of the SEAF
$SHA256()$	a hash function
SN	the SN
SNN	the serving network name of the SN
SQN	a fresh sequence number generated by the HN
SQN_{UE}	the highest sequence number the USIM has accepted
$SUCI$	a SUCI of the UE
$SUPI$	a SUPI of the UE
$Syncf$	the "Synchronization failure" indication
T_{AUSF}	the current time of the AUSF
T_{SN}	a timestamp generated by the SEAF
T_{UDM}	the current time of the UDM
UE	the UE
x	an ephemeral private key of the UE for Diffie–Hellman exchange
$x \cdot G$	an ephemeral public key of the UE for Diffie–Hellman exchange
XOR	the generation and verification of an XOR value
$XMAC$	a MAC locally computed by the UE
$XRES$	an expected response
$XRES^*$	an expected response from $XRES$
y	an ephemeral private key of the HN for Diffie–Hellman exchange
$y \cdot G$	an ephemeral public key of the HN for Diffie–Hellman exchange
Δt_{AUSF}	the time difference set by the AUSF
Δt_{UDM}	the time difference set by the UDM
$\|\|$	a concatenation

References

1. Xu, S.; Gan, Z. Review and trends of 5G security technology. *Radio Commun. Technol.* **2020**, *46*, 133–138.
2. 3GPP TS 33.102: 3G Security. Security Architecture. Available online: https://www.3gpp.org/DynaReport/33102.htm (accessed on 26 January 2022).
3. 3GPP TS 33.401: 3GPP System Architecture Evolution (SAE). Security Architecture. Available online: https://www.3gpp.org/DynaReport/33401.htm (accessed on 26 January 2022).
4. 3GPP TS 33.501: 3GPP System Architecture Evolution (SAE). Security Architecture. Available online: https://www.3gpp.org/DynaReport/33501.htm (accessed on 26 January 2022).
5. Ferrag, M.A.; Maglaras, L.; Argyriou, A.; Kosmano, D.; Janicke, H. Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes. *J. Netw. Comput. Appl.* **2018**, *101*, 55–82. [CrossRef]
6. Jover, R.P.; Marojevic, V. Security and protocol exploit analysis of the 5G specifications. *IEEE Access* **2019**, *7*, 24956–24963. [CrossRef]
7. Ahmad, I.; Shahabuddin, S.; Kumar, T.; Okwuibe, J.; Ylianttila, M. Security for 5G and beyond. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3682–3722. [CrossRef]
8. Khan, R.; Kumar, P.; Jayakody, D.N.K.; Liyanage, M. A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 196–248. [CrossRef]
9. Hussain, S.R.; Echeverria, M.; Karim, I.; Chowdhury, O.; Berino, E. 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, London, UK, 11–15 November 2019; pp. 669–684.
10. Hussain, S.R.; Echeverria, M.; Chowdhury, O.; Li, N.; Bertino, E. Privacy attacks to the 4G and 5G cellular paging protocols using side channel information. In Proceedings of the 26th Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 24–27 February 2019; pp. 1–15.
11. Khan, H.; Martin, K.M. A survey of subscription privacy on the 5G radio interface-the past, present and future. *J. Inf. Secur. Appl.* **2020**, *53*, 102537. [CrossRef]
12. Dehnel-Wild, M.; Cremers, C. *Security Vulnerability in 5G-AKA Draft*; Department of Computer Science, University of Oxford: Oxford, UK, 2018.
13. Meier, S.; Schmidt, B.; Cremers, C.; Basin, D. The Tamarin prover for the symbolic analysis of security protocols. In Proceedings of the 25th International Conference on Computer Aided Verification, Saint Petersburg, Russia, 13–19 July 2013; pp. 696–701.
14. Basin, D.; Dreier, J.; Hirschi, L.; Radomirovic, S.; Sasse, R.; Stettler, V. A formal analysis of 5G authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1383–1396.
15. Liu, F.; Peng, J.; Zuo, M. Toward a secure access to 5G network. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, New York, NY, USA, 1–3 August 2018; pp. 1121–1128.
16. Borgaonkar, R.; Hirschi, L.; Park, S.; Shaik, A. New privacy threat on 3G, 4G, and upcoming 5G AKA Protocols. *Proc. Priv. Enhancing Technol.* **2019**, *3*, 108–127. [CrossRef]
17. Cremers, C.; Dehnel-Wild, M. Component-based formal analysis of 5G-AKA: Channel assumptions and session confusion. In Proceedings of the 26th Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 24–27 February 2019; pp. 1–15.
18. Koutsos, A. The 5G-AKA authentication protocol privacy. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 17–19 June 2019; pp. 464–479.
19. Bana, G.; Comon-Lundh, H. Towards unconditional soundness: Computationally complete symbolic attacker. In Proceedings of the First international conference on Principles of Security and Trust (ETAPS), Tallinn, Estonia, 24 March–1 April 2012; pp. 189–208.
20. Bana, G.; Comon-Lundh, H. A computationally complete symbolic attacker for equivalence properties. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 609–620.
21. Braeken, A.; Liyanage, M.; Kumar, P.; Murphy, J. Novel 5G authentication protocol to improve the resistance against active attacks and malicious serving networks. *IEEE Access* **2019**, *7*, 64040–64052.
22. Gharsallah, I.; Smaoui, S.; Zarai, F. A secure efficient and lightweight authentication protocol for 5G cellular networks: SEL-AKA. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1311–1316.
23. Hu, X.; Liu, C.; Liu, S.; Cheng, X. A security enhanced 5G authentication scheme for insecure channel. *Trans. Inf. Syst.* **2020**, *103*, 711–713. [CrossRef]
24. Hu, X.; Liu, C.; Liu, S.; Li, J.; Cheng, X. A vulnerability in 5G authentication protocols and its Countermeasure. *IEICE Trans. Inf. Syst.* **2020**, *103*, 1806–1809. [CrossRef]
25. Edris, E.K.K.; Aiash, M.; Loo, J.K. Formal verification and analysis of primary authentication based on 5G-AKA protocol. In Proceedings of the 2020 7th International Conference on Software Defined Systems (SDS), Paris, France, 20–23 April 2020; pp. 256–261.
26. Ouaissa, M.; Ouaissa, M. An improved privacy authentication protocol for 5G mobile networks. In Proceedings of the 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), Dehradun, India, 21–22 August 2020; pp. 136–143.

27. Parne, B.L.; Gupta, S.; Gandhi, K.; Meena, S. PPSE: Privacy preservation and security efficient AKA protocol for 5G communication networks. In Proceedings of the 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), New Delhi, India, 14–17 December 2020; pp. 1–6.
28. Xiao, Y.; Wu, Y. 5G-IPAKA: An Improved Primary Authentication and Key Agreement Protocol for 5G Networks. *Information* **2022**, *13*, 125. [CrossRef]
29. Fábrega, F.J.T.; Herzog, J.C.; Guttman, J.D. Mixed strand spaces. In Proceedings of the 12th IEEE Computer Security Foundations Workshop, Mordano, Italy, 30 June 1999; pp. 72–82.
30. Fábrega, F.J.T.; Herzog, J.C.; Guttman, J.D. Strand space: Proving security protocols correct. *J. Comput. Secur.* **1999**, *7*, 191–230. [CrossRef]
31. Herzog, J.C. The Diffie-Hellman key-agreement scheme in the strand-space model. In Proceedings of the 16th IEEE Computer Security Foundation Workshop, Pacific Grove, CA, USA, 30 June–2 July 2003; pp. 234–247.
32. The Scyther Tool. Available online: http://www.cs.ox.ac.uk/people/cas.cremers/scyther (accessed on 7 April 2022).
33. Cremers, C.J.F. Scyther—Semantics and Verification of Security Protocols. Ph.D. Dissertation, Institute for Programming Research Algorithmics, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.
34. Sun, Y.; Cao, J.; Ma, M.; Zhang, Y.; Li, H.; Niu, B. EAP-DDBA: Efficient anonymity proximity device discovery and batch authentication mechanism for massive D2D communication devices in 3GPP 5G HetNet. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 370–387. [CrossRef]