Computer Security Symposium 2025
27-31 October 2025

How to Transform 5G-AKA into Stealth Key Exchange

辛　星漢[1,a)]

概要：For the 5G primary authentication, 5G-AKA (5G Authentication and Key Agreement) has been designed and standardized in the 3GPP consortium. In this paper, we discuss how to transform 5G-AKA into stealth key exchange while keeping compatibility with the current 3GPP standard.

キーワード：5G-AKA, stealth key exchange, forward secrecy, UE unlinkability

How to Transform 5G-AKA into Stealth Key Exchange

SEONGHAN SHIN[1,a)]

Abstract: For the 5G primary authentication, 5G-AKA (5G Authentication and Key Agreement) has been designed and standardized in the 3GPP consortium. In this paper, we discuss how to transform 5G-AKA into stealth key exchange while keeping compatibility with the current 3GPP standard.

Keywords: 5G-AKA, stealth key exchange, forward secrecy, UE unlinkability

1. Introduction

Currently, the fifth generation (5G) mobile network and telecommunication standard has been developed to meet the needs of enhanced mobile broadband, massive machine-type communications, and ultra-reliable and low-latency communications. Among several building blocks in this standard, 5G-AKA (5G Authentication and Key Agreement) [1] and EAP-AKA' (Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement) [1], developed by the 3GPP consortium, are of utmost importance, which allow a User Equipment (UE) and a Home Network (HN) to authenticate each other and establish key materials (i.e., anchor keys) for protecting subsequent 5G communications. The 5G-AKA protocol is a new version of the AKA variants used for 3G and 4G networks. A distinctive feature of the 5G-AKA protocol is that a SUPI (SUbscriber Permanent Identifier) of UE is sent to HN in the form of encrypted by using the ECIES (Elliptic Curve Integrated Encryption Scheme) KEM (Key Encapsulation Mechanism) with HN's public key.

In [2], Basin et al. provided a comprehensive formal model of the 5G-AKA protocol and evaluated the model with respect to the 5G security goals using the security protocol verification tool Tamarin [3]. Then, they found that some critical security goals for the 5G-AKA protocol were not met. In [4], Borgaonkar et al. showed a new privacy attack on subscriber privacy against all the AKA variants (including 5G-AKA) by exploiting a logical vulnerability in the protection mechanism of SQN (Sequence Number). Also, Koutsos [5] showed that all the known privacy attacks (except the IMSI-catcher attack) are possible in the 5G-AKA protocol and then proposed a modified 5G-AKA protocol that satisfies the unlinkability property and is proven in the Bana-Comon logic model [6]. After describing that the 5G-AKA protocol is still vulnerable to a series of active linkability attacks, Wang et al. [7] proposed a privacy-preserving 5G-AKA (called 5G-AKA') that is secure against active linkability attacks by encrypting a random challenge of

[1] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[a)] seonghan.shin@aist.go.jp

HN with an ECIES-KEM key, and is compatible with the SIM cards and Serving Networks (SNs). By using Tamarin [3], they also proved that the 5G-AKA' protocol achieves privacy, authentication, and secrecy. Recently, IETF EMU WG has been working on an EAP-AKA' (EAP-AKA' FS [8]) protocol that provides forward secrecy. The EAP-AKA' FS protocol is a simple combination of the EAP-AKA' [9] and Diffie-Hellman key exchange [10]. Very recently, You et al. [11] proposed an augmentation to 5G-AKA (5G-AKA-FS) to achieve forward security and thwart linkability attacks.

1.1 Motivation and Our Contributions

Recently, Fischlin [12] introduced a notion of stealth key exchange and showed how to embed a covert key exchange subprotocol into the regular TLS 1.3 execution [13], generating a stealth session key in addition to a regular session key. One prospective application of stealth key exchange is a sanitizable channel where a designated entity can partly access and modify payload data with a regular session key, while entities who share a stealth session key can covertly communicate in a channel protocol.

In this paper, we propose a stealth mode of 5G-AKA (for short, Stealth-5G-AKA) protocol that is compatible with the current 3GPP standard [1] with small modifications. In the Stealth-5G-AKA protocol, UE and SN share a stealth anchor key in addition to a (regular) anchor key. A main idea of the Stealth-5G-AKA protocol is that we set an embedded (elliptic curve) Diffie-Hellman public key as a random challenge, and an (elliptic curve) Diffie–Hellman key is used as a key material of a stealth anchor key for forward secrecy and unlinkability. Also, we discuss several security properties of the Stealth-5G-AKA protocol. Moreover, we compare the 5G-AKA [1] with relevant 3GPP-compatible protocols (5G-AKA' [7], 5G-AKA-FS [11] and Stealth-5G-AKA) with respect to security properties (stealthiness, security of stealth key, forward secrecy, UE anonymity/unlinkability and security of anchor key) and efficiency in terms of computation costs and communication costs.

2. Preliminaries

Table 1 shows abbreviations and notations to be used throughout this paper.

2.1 Elliptic Curve Integrated Encryption Scheme

The Elliptic Curve Integrated Encryption Scheme (ECIES) [14], [15], [16], [17] is a well-known hybrid encryption scheme consisting of a Key Encapsulation Mechanism (KEM) and a Data Encapsulation Mechanism (DEM) where messages of arbitrary length can be encrypted. This scheme is a key component of 5G-AKA.

The ECIES-KEM has the following three algorithms:

- **ECIES.KeyGen**(pp): On input of a public parameter pp, the algorithm outputs a private-public key pair (sk, PK) such that $PK = sk \cdot G$, where pp is an elliptic curve parameter standardized in [18], and $G \in pp$ is a base point.
- **ECIES.Encap**(PK): On input of a public key PK, the algorithm generates an ephemeral private-public key pair (r, R) such that $R = r \cdot G$, and then outputs a ciphertext $C_0 = R$ and a shared key $k_s = \mathsf{KDF}(r \cdot PK)$, where KDF is a key derivation function.
- **ECIES.Decap**(sk, C_0): On input of a private key sk and a ciphertext C_0, the algorithm outputs the shared key $k_s = \mathsf{KDF}(sk \cdot C_0)$.

The ECIES-DEM has the following two algorithms:

- **ECIES.SEnc**(k_s, M): On input of a key k_s and a message M, the algorithm first parses k_s as $k_1||k_2$, computes $C_1 = \mathsf{ENC}(k_1, M)$ and $C_2 = \mathsf{MAC}(k_2, C_1)$, and then outputs (C_1, C_2), where ENC is an encryption part of a symmetric encryption scheme and MAC is a message authentication code.
- **ECIES.SDec**$(k_s, (C_1, C_2))$: On input of a key k_s and a ciphertext (C_1, C_2), the algorithm first parses k_s as $k_1||k_2$. If $C_2 \neq \mathsf{MAC}(k_2, C_1)$, it outputs $\perp$. Otherwise, the algorithm outputs $M = \mathsf{DEC}(k_1, C_1)$, where DEC is a decryption part of a symmetric encryption scheme.

2.2 Stealth Key Exchange

Recently, Fischlin [12] introduced a notion of stealth key exchange and showed how to embed a covert key exchange subprotocol into the regular TLS 1.3 execution [13], generating a stealth session key in addition to a regular session key. The basic idea is to use exchanged nonces, which ensure that a session is unique, to transport other Diffie-Hellman public keys. One prospective application of stealth key exchange is a sanitizable channel where a designated entity[*1] can partly access and modify payload data with a regular session key, while entities who share a stealth session key can covertly communicate in a channel protocol (refer to [12] for the other applications).

A stealth key exchange protocol has the following security properties [12]:

Stealthiness: An outside attacker remains oblivious if a stealth mode has been used (i.e., a stealth session key has been generated) or not in the protocol execution.

Security of stealth key: A stealth session key should be se-

[*1] For example, it can be a sanitizer, who monitors incoming packets for malicious contents (e.g., malware) and locates suspicious ones in quarantine, in an intrusion detection system.

表 1　Abbreviations and notations

	Meanings
UE (ME, SIM)	User Equipment (Mobile Equipment, Subscriber Identity Modules)
HN	Home Network
SN	Serving Network
SUPI	SUbscriber Permanent Identifier
SUCI	SUbscriber Concealed Identifier
KEM	Key Encapsulation Mechanism
DEM	Data Encapsulation Mechanism
AMF	Access Management Function
AUSF	AUthentication Server Function
SEAF	SEcurity Anchor Function
k	A permanent key shared between UE and HN
K_{AUSF}	A master session key derived from Stealth-5G-AKA
K'_{AUSF}	A master stealth session key derived from Stealth-5G-AKA
K_{SEAF}	An anchor key derived from Stealth-5G-AKA
K'_{SEAF}	A stealth anchor key derived from Stealth-5G-AKA
k_{UE}	UE's shared key established by ECIES-KEM
k_{HN}	HN's shared key established by ECIES-KEM
(PK_{HN}, sk_{HN})	HN's ECIES public-private key pair where $PK_{HN} = sk_{HN} \cdot G$
ID_{SN}	A unique identifier of SN
ID_{HN}	A unique identifier of HN
SQN_{UE}	A sequence number of UE
SQN_{HN}	A sequence number of HN
$RAND$	A random challenge of HN

cure even if an attacker can control a regular session key derived in the same protocol execution.

3. A Stealth 5G-AKA Protocol

In this section, we propose a stealth mode of 5G-AKA (for short, Stealth-5G-AKA) protocol that is compatible with the current 3GPP standard [1] with small modifications. In the Stealth-5G-AKA protocol, UE and SN share a stealth anchor key K'_{SEAF} in addition to an anchor key K_{SEAF}. Also, the Stealth-5G-AKA protocol provides forward secrecy and unlinkability like [11]. Before executing the Stealth-5G-AKA protocol, UE holds $(k, PK_{HN}, SUPI, SQN_{UE})$ secretly and HN stores $(k, sk_{HN}, ID_{HN}, SQN_{HN})$ secretly. Also, SN stores ID_{SN}. Note that SN and HN are connected with a secure channel.

3.1 The Initiation Phase: Step 1

In this phase, UE sends its $SUPI$ as an encrypted form using ECIES [14], [15], [16], [17] with HN's public key PK_{HN}. Correspondingly, HN decrypts $SUCI$ with its private key sk_{HN}. After successful completion of Step 1, we proceed to Step 2 by selecting Stealth-5G-AKA among various authentication methods.

3.1.1 Step 1.1 (UE)

With HN's public key PK_{HN}, UE executes the followings:

(1) Generate an ephemeral ECIES private-public key pair (r, R) such that $R = r \cdot G$
(2) With PK_{HN}, compute a ciphertext $C_0 = R$ and a key $k_{UE} = \mathsf{KDF}(r \cdot PK_{HN})$
(3) Parse the shared key k_{UE} as $k_1 || k_2$
(4) Compute $C_1 = \mathsf{ENC}(k_1, SUPI)$ and $C_2 = \mathsf{MAC}(k_2, C_1)$
(5) Set $SUCI \leftarrow (C_0, C_1, C_2)$

UE sends $SUCI$ to SN.

3.1.2 Step 1.2 (SN)

SN receives $SUCI$ from UE and then SN sends $(SUCI, ID_{SN})$ to HN.

3.1.3 Step 1.3 (HN)

Upon receiving $(SUCI, ID_{SN})$ from SN, HN executes the followings:

(1) Compute a shared key $k_{HN} = \mathsf{KDF}(sk_{HN} \cdot C_0)$

(2) Parse the shared key k_{HN} as $k_1 || k_2$

If $C_2 \neq \mathsf{MAC}(k_2, C_1)$, HN outputs $\perp$. Otherwise, it outputs $SUPI = \mathsf{DEC}(k_1, C_1)$, and retrieves the corresponding k and SQN_{HN} from its database.

3.2 The Challenge-Response Phase: Step 2

In this phase, UE and HN authenticate each other via a challenge-response method and establish anchor keys (i.e., K_{SEAF} and K'_{SEAF}) together with SN. A main idea of the Stealth-5G-AKA protocol is that we set an embedded (elliptic curve) Diffie-Hellman public key Y as a random challenge $RAND$, and an (elliptic curve) Diffie–Hellman key DHK is used as a key material of a stealth anchor key for forward secrecy and unlinkability. This phase uses a series of HMAC-SHA-256 cryptographic key derivation functions $\mathsf{f}_1, \mathsf{f}_2, \mathsf{f}_3, \mathsf{f}_4, \mathsf{f}_5, \mathsf{f}_1^*$, and f_5^*, as specified by TS 33.501 [1]. Also, we denote by $\mathsf{H}_{\text{SHA-256}}$ the SHA-256 cryptographic hash function.

3.2.1 Step 2.1 (HN)

Using (k, SQN_{HN}), HN generates an authentication vector $SE\text{-}AV$ as follows:

(1) Regular mode: Choose a random challenge $RAND \xleftarrow{\$} \{0,1\}^{256}$ [*2]

(2) Stealth mode: Generate an ephemeral Diffie-Hellman private-public key pair (y, Y) such that $Y = y \cdot G$

(3) Stealth mode: Set $RAND \leftarrow \mathsf{Embed}(Y)$ as a random challenge

(4) Compute $MAC \leftarrow \mathsf{f}_1(k, SQN_{HN}, AMF, RAND)$ and an anonymous key $AK \leftarrow \mathsf{f}_5(k, RAND)$

(5) Set $AUTN \leftarrow (AK \oplus SQN_{HN}, AMF, MAC)$

(6) Compute $CK \leftarrow \mathsf{f}_3(k, RAND)$ and $IK \leftarrow \mathsf{f}_4(k, RAND)$

(7) Compute expected responses $XRES \leftarrow \mathsf{f}_2(k, RAND)$, $XRES^* \leftarrow \mathsf{KDF}(CK||IK, ID_{SN}||RAND||XRES)$, and $HXRES^* \leftarrow \mathsf{LEFT}(128, \mathsf{H}_{\text{SHA-256}}(RAND||XRES^*))$

(8) Derive $K_{AUSF} \leftarrow \mathsf{KDF}(CK||IK, ID_{SN}||AK \oplus SQN_{HN})$ and $K_{SEAF} \leftarrow \mathsf{KDF}(K_{AUSF}, ID_{SN})$

(9) Increase SQN_{HN} by 1 (i.e., $SQN_{HN} \leftarrow SQN_{HN} + 1$)

(10)Set $HE\text{-}AV \leftarrow (RAND, AUTN, XRES^*, K_{AUSF})$ and $SE\text{-}AV \leftarrow (RAND, AUTN, HXRES^*)$

(11)Stealth mode: Compute a Diffie-Hellman key $DHK = y \cdot C_0$

(12)Stealth mode: Compute $CK' \leftarrow \mathsf{f}_3(k, RAND \oplus DHK)$ and $IK' \leftarrow \mathsf{f}_4(k, RAND \oplus DHK)$

(13)Stealth mode: Derive $K'_{AUSF} \leftarrow \mathsf{KDF}(CK'||IK', ID_{SN}||AK \oplus SQN_{HN})$ and $K'_{SEAF} \leftarrow \mathsf{KDF}(K'_{AUSF}, ID_{SN})$

[*2] Here, we use a 256-bit $RAND$ instead of a 128-bit $RAND$ in the 3GPP standard [1].

HN sends $SE\text{-}AV$ to SN.

3.2.2 Step 2.2 (SN)

SN receives $SE\text{-}AV$ from HN and stores $(RAND, HXRES^*)$. Then, SN sends $(RAND, AUTN)$ to UE.

3.2.3 Step 2.3 (UE)

Upon receiving $(RAND, AUTN)$ from SN, UE executes the followings:

(1) ME forwards $(RAND, AUTN)$ to SIM card.

(2) Run SIM card command `AUTHENTICATE`$(RAND, AUTN)$

(3) Compute $AK \leftarrow \mathsf{f}_5(k, RAND)$

(4) Parse $AUTN$ as $(CONC, AMF, MAC)$

(5) De-conceal $SQN_{HN} \leftarrow AK \oplus CONC$

(6) Check $\mathsf{f}_1(k, SQN_{HN}, AMF, RAND) = MAC$

- If this check does not pass, SIM card returns $\perp$ and then UE sends a failure message `Mac_Failure` to SN (see *Case i*).
- Otherwise, proceed to the next step

(7) Check $SQN_{UE} < SQN_{HN} < SQN_{UE} + \Delta$ [*3]

- If this check does not pass, SIM card computes $MAC^* \leftarrow \mathsf{f}_1^*(k, SQN_{UE}, AMF, RAND)$ and returns $AUTS \leftarrow (AK^* \oplus SQN_{UE}, AMF, MAC^*)$ where $AK^* \leftarrow \mathsf{f}_5^*(k, RAND)$. Then, UE re-synchronizes with HN by sending a failure message `Sync_Failure` and $AUTS$ to SN (see *Case ii*).
- Otherwise, proceed to the next step

(8) Set $SQN_{UE} \leftarrow SQN_{HN}$

(9) Compute $CK \leftarrow \mathsf{f}_3(k, RAND)$, $IK \leftarrow \mathsf{f}_4(k, RAND)$ and $RES \leftarrow \mathsf{f}_2(k, RAND)$

(10)Stealth mode: Set $Y \leftarrow \mathsf{Embed}^{-1}(RAND)$

(11)Stealth mode: Compute a Diffie-Hellman key $DHK = r \cdot Y$

(12)Stealth mode: Compute $CK' \leftarrow \mathsf{f}_3(k, RAND \oplus DHK)$ and $IK' \leftarrow \mathsf{f}_4(k, RAND \oplus DHK)$

(13)SIM card returns (CK, IK, RES, CK', IK') to ME.

(14)ME computes $RES^* \leftarrow \mathsf{KDF}(CK||IK, ID_{SN}||RAND||RES)$, and derives $K_{AUSF} \leftarrow \mathsf{KDF}(CK||IK, ID_{SN}||CONC)$ and $K_{SEAF} \leftarrow \mathsf{KDF}(K_{AUSF}, ID_{SN})$.

(15)Stealth mode: ME derives $K'_{AUSF} \leftarrow \mathsf{KDF}(CK'||IK', ID_{SN}||CONC)$ and $K'_{SEAF} \leftarrow \mathsf{KDF}(K'_{AUSF}, ID_{SN})$.

(16)ME returns $(K_{SEAF}, K'_{SEAF}, RES^*)$.

UE stores (K_{SEAF}, K'_{SEAF}) and sends RES^* to SN (see *Case*

[*3] The first condition $SQN_{UE} < SQN_{HN}$ ensures the freshness of $(RAND, AUTN)$. Also, the second condition $SQN_{HN} < SQN_{UE} + \Delta$, which is optional in the non-normative Annex C of TS 33.102 [19], prevents a wrap-around of SQN_{UE}. For example, if Δ is too small (i.e., $\Delta = 2$), an attacker can make a synchronization failure by sending $SUCI$ computed by the attacker with a fake $SUPI$. After this attack, the honest UE and HN can no longer authenticate each other. In TS 33.102 [19], a recommended value of Δ is 2^{28} so as to decrease the synchronization failure rate.

iii).

Case i (**SIM card returns** $\perp$)**:** The UE sends a failure message `Mac_Failure` to the SN.

Case ii (**SIM card returns** $AUTS$)**:** The UE re-synchronizes with the HN by sending a failure message `Sync_Failure` and $AUTS$ to the SN. Upon receiving (`Sync_Failure`, $AUTS$), SN sends (`Sync_Failure`,$AUTS$,$RAND$,$SUCI$) to the HN. Then, HN parses $AUTS$ as $(AK^* \oplus SQN_{UE}, AMF, MAC^*)$, and de-conceals $SQN_{UE} \leftarrow AK^* \oplus SQN_{UE} \oplus \mathsf{f}_5^*(k, RAND)$. Next, HN checks its authenticity by comparing $MAC^* = \mathsf{f}_1^*(k, SQN_{UE}, AMF, RAND)$. If the check holds, HN re-sets SQN_{HN} by $SQN_{UE}+1$ (i.e., $SQN_{HN} \leftarrow SQN_{UE}+1$).

Case iii (**ME returns** $(K_{SEAF}, K'_{SEAF}, RES^*)$)**:** The UE stores (K_{SEAF}, K'_{SEAF}) and sends RES^* to the SN. Upon receiving RES^*, SN computes a hashed value $HRES^* \leftarrow \mathsf{LEFT}(128, \mathsf{H}_{\text{SHA-256}}(RAND||RES^*))$ and checks its validity by comparing $HRES^*$ with $HXRES^*$. If $HRES^* = HXRES^*$, SN forwards RES^* to the HN. Next, HN authenticates the UE by comparing RES^* with $XRES^*$. If $RES^* = XRES^*$, HN sends its result and $(SUPI, K_{SEAF}, K'_{SEAF})$ to the SN. The SN continues the protocol only if both checks hold, and aborts the protocol otherwise. When all checks pass, UE and SN communicate with session keys derived from anchor keys (i.e., (K_{SEAF}, K'_{SEAF})) in the subsequent 5G procedures. According to TS 33.501 [1], 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}.

Remark1 In the Stealth-5G-AKA protocol, we use the elliptic curve `Curve25519` [20] for Diffie-Hellman key exchange parts instead of the recommended elliptic curves (e.g., `secp256k1` and `secp256r1`) in [18]. For Embed, we use the embedding technique Elligator 2 [21], which allows us to efficiently create elliptic curve points that are statistically indistinguishable from uniform random bit strings (see the Elligator webpage [22] for more details). Note that Elligator 2 is more general and works well with `Curve25519` while Elligator 1 based on [23] works for some elliptic curves. In [21], Bernstein et al. discussed an issue that the NIST elliptic curve `P-256` may not easily yield almost uniform random bit strings since the order of `P-256` is not sufficiently close to a power of 2. Also, the other option for Embed can be found in [12], where the random mapping Embed maps a large portion of elliptic curve points to strings that are statistically close to uniform bit strings.

4. Security of Stealth-5G-AKA

Due to the lack of space, we informally discuss several security properties of the Stealth-5G-AKA protocol.

4.1 Stealthiness

Here, we show that the Stealth-5G-AKA protocol provides stealthiness by giving a reduction to the security of Embed. Let $\text{transcript}_{\text{Regular}}$ (resp., $\text{transcript}_{\text{Stealth}}$) be a communication transcript $(SUCI, (RAND, AUTN), RES^*)$ of the regular (resp., stealth) mode of Stealth-5G-AKA. Also, let $\mathcal{B}$ be an attacker who can distinguish an embedded elliptic curve point from a uniform random bit string. The only difference between $\text{transcript}_{\text{Regular}}$ and $\text{transcript}_{\text{Stealth}}$ is that $RAND \xleftarrow{\$} \{0,1\}^{256}$ is used in the regular mode and $RAND \leftarrow \mathsf{Embed}(Y)$ is used in the stealth mode. We assume that there exists an outside attacker $\mathcal{A}$ who can distinguish the regular mode from the stealth mode in the Stealth-5G-AKA protocol execution. By using $\mathcal{A}$, we can trivially construct an attacker $\mathcal{B}$ who breaks the security of Embed. In other words, the outside attacker remains oblivious if the stealth mode has been used or not in the Stealth-5G-AKA protocol execution.

4.2 Security of Stealth Key

Next, we show that a stealth anchor key is secure even if an attacker can control a (regular) anchor key derived in the same Stealth-5G-AKA protocol execution. We assume that there exists an attacker who can control the anchor key K_{SEAF} derived in the Stealth-5G-AKA protocol. Thanks to the security of KDF and the master stealth session key K'_{AUSF}, it is clear that the stealth anchor key K'_{SEAF} is secure against the attacker.

4.3 Forward Secrecy

In the Stealth-5G-AKA protocol, forward secrecy should be discussed in two flavors: an anchor key and a stealth anchor key. Regarding the stealth anchor key K'_{SEAF}, the Stealth-5G-AKA protocol guarantees forward secrecy since the ephemeral (elliptic curve) Diffie-Hellman key exchange [10] is inherent in its construction. Specifically, the ephemeral Diffie-Hellman public key on the UE (resp., HN) side is $R = r \cdot G$ in **Step 1.1** (resp., $Y = y \cdot G$ in **Step 2.1**) and the ephemeral Diffie–Hellman key $DHK = y \cdot C_0$ in **Step 2.1** (resp., $DHK = r \cdot Y$ in **Step 2.3**) is used as a key material of the stealth anchor key K'_{SEAF} on the HN (resp., UE) side. Note that KEM can be generally understood as the Diffie-Hellman key exchange.

Regarding the anchor key K_{SEAF}, the Stealth-5G-AKA protocol does not provide forward secrecy because the computation of K_{SEAF} is the same as the 5G-AKA protocol. This problem arises from the fact that the 5G-AKA protocol itself has no forward secrecy [11]. A fix to this problem is to use $RAND \oplus DHK$ instead of $RAND$ in the computations of $\mathsf{f}_1, \mathsf{f}_2, \mathsf{f}_3, \mathsf{f}_4, \mathsf{f}_5, \mathsf{f}_1^*$, and f_5^*. However, this fix requires modifications to the 3GPP standard [1].

4.4 UE Unlinkability

Similar to Section 4.3, UE unlinkability in the Stealth-5G-AKA protocol should be discussed in two flavors: an anchor key and a stealth anchor key. Regarding the stealth anchor key K'_{SEAF}, the Stealth-5G-AKA protocol guarantees UE unlinkability since the ephemeral (elliptic curve) ECIES public key $R = r \cdot G$ in **Step 1.1** on the UE side is used in the ephemeral Diffie–Hellman key $DHK = y \cdot R$ in **Step 2.1** which is subsequently used as a key material of the stealth anchor key K'_{SEAF} on the HN side.

Regarding the anchor key K_{SEAF}, the Stealth-5G-AKA protocol does not provide UE unlinkability because the computation of K_{SEAF} is the same as the 5G-AKA protocol. This problem arises from the fact that the 5G-AKA protocol itself is susceptible to UE linkability attacks (e.g., [7]). Basically, this problem can be solved by tightly relating $SUCI$ to the subsequent computations on the HN side. As in Section 4.3, a fix to this problem is to use $RAND \oplus DHK$ instead of $RAND$ in the computations of $\mathsf{f}_1, \mathsf{f}_2, \mathsf{f}_3, \mathsf{f}_4, \mathsf{f}_5, \mathsf{f}_1^*$, and f_5^*. However, this fix also requires modifications to the 3GPP standard [1].

5. Comparison

In this section, we compare the 5G-AKA [1] with relevant 3GPP-compatible protocols (5G-AKA' [7], 5G-AKA-FS [11] and Stealth-5G-AKA) with respect to several security properties (stealthiness, security of stealth key, forward secrecy, UE anonymity/unlinkability and security of anchor key) and efficiency in terms of computation costs and communication costs. We summarize comparative results in Table 2 and Table 3.

From Table 2, it is clear that the only Stealth-5G-AKA protocol provides stealthiness and security of stealth anchor key. In Table 3, $RAND$ in the 5G-AKA [1] protocol is a 128-bit random challenge while $RAND$ in the 5G-AKA-FS [11] and Stealth-5G-AKA protocols is a 256-bit random challenge. For Embed, we use the embedding technique Elligator 2 [21], which needs two trials to find a suitable elliptic curve point on average and an inversion for the embedding. Note that two elliptic curve modular multiplications (i.e., R and k_{UE}) on the UE side, and one elliptic curve modular multiplication (i.e., Y) and Embed on the HN side can be pre-computed in the Stealth-5G-AKA protocol. From Table 2 and Table 3, one can see that there is a trade-off of security and efficiency between the 5G-AKA-FS [11] and Stealth-5G-AKA protocols. Also, we stress that the Stealth-5G-AKA protocol is fully compatible with the 3GPP standard [1] with small modifications to achieve stealthiness in the 5G-AKA protocol.

6. Conclusion

In this paper, we proposed a stealth mode of 5G-AKA (for short, Stealth-5G-AKA) protocol that is compatible with the current 3GPP standard [1] with small modifications. In the Stealth-5G-AKA protocol, UE and SN share a stealth anchor key in addition to a (regular) anchor key. Also, we discussed several security properties of the Stealth-5G-AKA protocol. Moreover, we compared the 5G-AKA [1] with relevant 3GPP-compatible protocols (5G-AKA' [7], 5G-AKA-FS [11] and Stealth-5G-AKA) with respect to security properties (stealthiness, security of stealth key, forward secrecy, UE anonymity/unlinkability and security of anchor key) and efficiency in terms of computation costs and communication costs. As in [12], we expect that the Stealth-5G-AKA protocol can be applied to a sanitizable channel (e.g., for private 5G) where a designated entity can partly access and modify payload data with a (regular) anchor key, while entities who share a stealth anchor key can covertly communicate in a channel protocol.

Future works of the Stealth-5G-AKA protocol include formal security analysis using security verification tools (e.g., Tamarin or ProVerif) and experimental performance evaluation on a realistic 5G testbed or a simulation framework for deployment feasibility in practical settings.

参考文献

[1] 3GPP TS 33.501, Security architecture and procedures for 5G system (Release 18.2), `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169`, June 2023

[2] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, V. Stettler, A Formal Analysis of 5G Authentication, CCS 2018, pp. 1383–1396, ACM, 2018

[3] S. Meier, B. Schmidt, C. Cremers, D. Basin, The TAMARIN Prover for the Symbolic Analysis of Security Protocols, Computer Aided Verification (CAV) 2013, pp. 696–701, Springer, 2013

[4] R. Borgaonkar, L. Hirschi, S. Park, A. Shaik, New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols, Proceedings on Privacy Enhancing Technologies (PoPETs), Vol. 2019, No. 3, pp. 108–127, 2019

[5] A. Koutsos, The 5G-AKA Authentication Protocol Privacy, EuroS&P 2019, pp. 464–479, IEEE, 2019

[6] G. Bana, H. C.-Lundh, A Computationally Complete Symbolic Attacker for Equivalence Properties, CCS 2014, pp. 609–620, ACM, 2014

[7] Y. Wang, Z. Zhang, Y. Xie, Privacy-Preserving and Standard-Compatible AKA Protocol for 5G, USENIX Security Symposium 2021, pp. 3595–3612, USENIX Association, 2021

表 2 Security comparison of the 5G-AKA and relevant 3GPP-compatible protocols

Protocols	Stealthiness	Security of Stealth Key	Forward Secrecy	UE Anonymity / Unlinkability	Security of Anchor Key
5G-AKA [1]	No	No	No	Yes / No	Yes
5G-AKA' [7]	No	No	No	Yes / Yes	Yes
5G-AKA-FS [11]	No	No	Yes	Yes / Yes	Yes
Stealth-5G-AKA	Yes	Yes	Yes	Yes / Yes	Yes

表 3 Efficiency comparison of the 5G-AKA and relevant 3GPP-compatible protocols where Mul is an elliptic curve modular multiplication and $|l|$ indicates a bit-length of l

Protocols	Computation costs of		Communication costs between UE and SN
	UE	HN	
5G-AKA [1]	2Mul	1Mul	$\lvert SUCI\rvert + \lvert RAND\rvert + \lvert AUTN\rvert + \lvert RES^*\rvert$
5G-AKA' [7]	2Mul	1Mul	$\lvert SUCI\rvert + \lvert \mathsf{ENC}\rvert + \lvert AUTN\rvert + \lvert RES^*\rvert$
5G-AKA-FS [11]	3Mul	3Mul	$\lvert SUCI\rvert + \lvert RAND\rvert + \lvert AUTN\rvert + \lvert RES^*\rvert$
Stealth-5G-AKA	3Mul+ Embed^{-1}	3Mul+ Embed	$\lvert SUCI\rvert + \lvert RAND\rvert + \lvert AUTN\rvert + \lvert RES^*\rvert$

[8] IETF Internet-Draft, Forward Secrecy for the Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS), `https://www.ietf.org/archive/id/draft-ietf-emu-aka-pfs-12.html`, February 2024

[9] IETF RFC 9048, Improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA'), `https://www.rfc-editor.org/rfc/rfc9048.html`, October 2021

[10] W. Diffie, M. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644–654, IEEE, 1976

[11] I. You, G. Kim, S. Shin, H. Kwon, J. Kim, J. Baek, 5G-AKA-FS: A 5G Authentication and Key Agreement Protocol for Forward Secrecy, Sensors, Vol. 24(1), No. 159, MDPI, 2024

[12] M. Fischlin, Stealth Key Exchange and Confined Access to the Record Protocol Data in TLS 1.3, CCS 2023, PP. 2901–2914, ACM, `https://eprint.iacr.org/2023/651`, 2023

[13] IETF RFC 8446, The Transport Layer Security (TLS) Protocol Version 1.3, `https://www.rfc-editor.org/rfc/rfc8446.html`, August 2018

[14] V. Shoup, A Proposal for an ISO Standard for Public Key Encryption (version 2.1), `https://www.shoup.net/papers/iso-2_1.pdf`, December 2001

[15] SECG SEC 1, Elliptic Curve Cryptography, `https://www.secg.org/sec1-v2.pdf`, May 2009

[16] ISO/IEC 18033-2:2006, Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers, `https://www.iso.org/standard/37971.html`, May 2006

[17] BSI TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths, `https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html`, January 2024

[18] SECG SEC 2, Recommended Elliptic Curve Domain Parameters, `https://www.secg.org/sec2-v2.pdf`, January 2010

[19] 3GPP TS 33.102, 3G security; Security architecture (Release 16), `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2262`, July 2020

[20] D. J. Bernstein, Curve25519: New Diffie-Hellman Speed Records, PKC 2006, PP. 207–228, Springer, 2006

[21] D. J. Bernstein, M. Hamburg, A. Krasnova, T. Lange, Elligator: elliptic-curve points indistinguishable from uniform random strings, CCS 2013, pp. 967–980, ACM, 2013

[22] Elligator: Hiding cryptographic key exchange as random noise, `https://elligator.org/`

[23] P.-A. Fouque, A. Joux, M. Tibouchi, Injective Encodings to Elliptic Curves, ACISP 2013, pp. 203–218, Springer, 2013