Identity Confidentiality in 5G Mobile Telephony Systems

Haibat Khan, Benjamin Dowling, and Keith M. Martin

Information Security Group, Royal Holloway, University of London
Haibat.Khan.2016@live.rhul.ac.uk {Benjamin.Dowling,Keith.Martin}@rhul.ac.uk

Abstract. The 3rd Generation Partnership Project (3GPP) recently proposed a standard for 5G telecommunications, containing an identity protection scheme meant to address the long-outstanding privacy problem of permanent subscriber-identity disclosure. The proposal is essentially two disjoint phases: an identification phase, followed by an establishment of security context between mobile subscribers and their service providers via symmetric-key based authenticated key agreement. Currently, 3GPP proposes to protect the identification phase with a public-key based solution, and while the current proposal is secure against a classical adversary, the same would not be true of a quantum adversary. 5G specifications target very long-term deployment scenarios (well beyond the year 2030), therefore it is imperative that quantum-secure alternatives be part of the current specification. In this paper, we present such an alternative scheme for the problem of private identification protection. Our solution is compatible with the current 5G specifications, depending mostly on cryptographic primitives already specified in 5G, adding minimal performance overhead and requiring minor changes in existing message structures. Finally, we provide a detailed formal security analysis of our solution in a novel security framework.

Keywords: 5G Security · Authentication · Privacy · Mobile Networks.

1 Introduction

While many mobile users may be comfortable with the fact that their service provider is able to identify them and track their geographical location ubiquitously, fewer are likely to be comfortable with an arbitrary third party having this capability. In hand of a third party, such capability could lead to undesirable breaches of end user privacy, opening the door to a range of potential consequences, such as harassment, stalking, employee monitoring, commercial profiling, etc. For these reasons, the global mobile telephony standardization body, the 3rd Generation Partnership Project (3GPP) has identified the following essential requirements related to user privacy [8]:

- **User Identity Confidentiality**[1]: The permanent identity of a user to whom a service is delivered cannot be eavesdropped on the radio access link.
- **User Location Confidentiality**: The presence or the arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link.
- **User Untraceability**: An intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link.

In mobile telephony systems, networks allocate to each SIM card a unique identifier,[2] known up to the Fourth Generation (4G) as an *International Mobile Subscriber Identity* (IMSI) and for the Fifth Generation (5G) as a *Subscription Permanent Identifier* (SUPI). As authentication between a user and its service provider is based on a shared symmetric key, it can only take place after user identification. However, if the IMSI/SUPI values are sent in plaintext over the radio access link, then users can be identified, located and tracked using these permanent identifiers. To avoid this privacy breach, the SIM card is assigned temporary identifiers (called *Temporary Mobile Subscriber Identity* (TMSI) until 3G systems and *Globally Unique Temporary User Equipment Identity* (GUTI) for 4G and 5G systems) by the visited network. These frequently-changing temporary identifiers are then used for identification purposes over the radio access link.

However, there are certain situations where authentication through the use of temporary identifiers is not possible. For instance, when a user registers with a network for the first time and is not yet assigned a temporary identifier. Another case is when the visited network is unable to resolve the IMSI/SUPI

[1] The official 3GPP documentation uses the term “Identity Confidentiality” to refer to the privacy of user identity. We follow the 3GPP naming convention.

[2] Users can also be identified through other unique identifiers, for instance *International Mobile Equipment Identity* (IMEI) which uniquely identifies the mobile equipment. However, it is only the IMSI/SUPI which is used for initial identification purposes.

from the presented TMSI/GUTI. An active man-in-the-middle adversary can intentionally simulate this scenario to force an unsuspecting user to reveal its long-term identity. These attacks are known as "IMSI-catching" attacks [17] and persist in today's mobile networks including the 4G LTE/LTE+ [20].

Defeating IMSI Catchers in 5G. IMSI-catching attacks have threatened all generations (2G/3G/4G) of mobile telecommunication for decades [5]. As a result of facilitating backwards compatibility for legacy reasons, this privacy problem appears to have persisted [6]. However, the 3GPP has now decided to address this issue, albeit at the cost of backward compatibility. In case of identification failure via a 5G-GUTI, unlike earlier generations, 5G security specifications do not allow plaintext transmissions of the SUPI over the radio interface [11]. Instead, an *Elliptic Curve Integrated Encryption Scheme* (ECIES)-based privacy-preserving identifier containing the concealed SUPI is transmitted [22]. We elaborate upon the details of this scheme further in Section 3.1.

Motivation. It is hoped that 5G specifications will be finalized in 2019. The first practical 5G deployments can then be expected a number of years later. However, it will almost certainly take a decade or so before all legacy systems are likely to be upgraded to 5G. It would thus seem that IMSI-catching attacks will remain an issue in the mid-term future, possibly even beyond 2030. By that time practical quantum computers will pose a much more immediate threat than they do today [3, 14], particularly with respect to cryptographic schemes such as ECIES, which are known to be vulnerable to quantum algorithms. The impact of quantum computers on mobile networks is already being discussed within the telephony industry [18], with a call to implement quantum-secure cryptography. It is thus imperative that 5G security specifications such as 3GPP TS 33.501 [11] (hereafter referred as TS 33.501) contain options for quantum-resistant schemes. Fortunately, 5G security has mostly relied upon symmetric cryptography for achieving its security objectives. However, the ECIES-based identification mechanism is an exception. In this paper, we propose a symmetric alternative to the ECIES mechanism, so that the "all symmetric" stature of 5G security can continue in a quantum future. Any proposal for an alternative user identification scheme for 5G systems should strive to satisfy the following requirements:

- Provision appropriate privacy guarantees such as anonymity and unlinkability against a quantum adversary.
- The performance overhead should be minimalistic.
- Offer appropriate deterrence against loss of synchronization between the user and its home network.
- Fulfill "Lawful Interception" requirements (details in Section 2.3) in mobile telecommunications.
- Ideally should adhere to the existing message structures as specified in current 5G specifications.

Our Contributions. The contributions of this paper are listed as below:

- We detail limitations of the ECIES-based identification scheme of TS 33.501.
- We present an alternate quantum resistant scheme which overcomes the limitations identified in the 3GPP scheme.
- We develop an appropriate model of security and formally prove the privacy guarantees offered by our proposal in this model.

Related Work. To our knowledge, this is the first work on 5G identity confidentiality since the publication of TS 33.501. Before a protection scheme was chosen, a study was conducted by 3GPP to evaluate a number of potential solutions. In total 24 proposals were considered, details of which can be found in the associated report 3GPP TR 33.899 (cf. Clause 5.7.4) [7]. Most (but not all) proposals were based on public-key cryptography, and the ECIES-based mechanism was selected as the final candidate. The few symmetric-key proposals all relied on utilizing pseudonyms for privacy purposes, and thus were susceptible to desynchronization attacks potentially causing permanent DoS attacks on the mobile users.

Various academic works have considered "IMSI-catching" attacks. The major thrust of these papers has been to devise a solution for 3G/4G without modifying the existing message structures out of concern for legacy devices and backwards-compatibility. Broek et al. [2] introduced a proposal based on changing pseudonyms, and required no modifications to the existing infrastructure. As a result of reliance on changing pseudonyms this solution was susceptible to desynchronization attacks. A similar proposal was that by Khan and Mitchell [15] which relied on using a set of IMSIs for a particular *USIM* to offer some degree of pseudonymity, however as in the case of [2], this solution could also get knocked out of the service permanently. Khan and Mitchell, based upon their previous work, subsequently presented an improved solution [16]. This solution relied on using a dynamic pseudo-IMSI for identification purposes, however identity desynchronization attacks still had the potential to cause permanent denial of service. Thus their solution is accompanied with an identity recovery mechanism (in case of desynchronization) which required no changes to the existing message structures. However, this solution fails to satisfy the LI requirements (see Section 2.3) without further changes to the existing message structures.

Paper Outline. The rest of the paper is organized as follows: Section 2 gives background on the 5G security architecture. Section 3 details the current identity confidentiality mechanism of 5G and its limitations. Section 4 introduces our proposal for 5G identity confidentiality, we define a security framework in Section 5 with which to assess our proposal, and in Section 6 presents its analysis. Finally, Section 7 concludes the article.

2 Essential Background on 5G Telephony

Here we explain the pertinent constituents of the 5G security ecosystem. We use a simpler terminology than that used in TS 33.501 in order to improve clarity.

2.1 Network Architecture

The mobile telephony network architecture consists of three major entities. *User Equipment* (UE), refers to a complete mobile phone and covers both the *Mobile Equipment* (ME) (the phone) and the *Universal Subscriber Identity Module* (USIM) (the SIM card). The USIM represents the relationship between the subscriber (end user) and its issuing *Home Network* (HN). During the USIM registration, the HN stores a unique SUPI, telephone number and other subscriber related data, including a secret key K and sequence number SQN, in the USIM. These subscriber related parameters also get stored by the corresponding HNs in their databases. These stored parameters later form the basis for security between the UEs and their HNs. Usually, a semi-trusted *Serving Network* (SN) provides the subscribers with access to the services of their HN. These services are provisioned after mutual authentication and establishment of a secure channel between the UE and SN with the help of the HN. When roaming, the serving network is referred to as the visited network. The communication medium between the UE and SN is wireless while that between the SN and HN is almost always a wired one.

2.2 Terminologies and Identities

A SUPI as defined in 3GPP TS 23.501 is usually a string of 15 decimal digits [12]. The first three digits represent the *Mobile Country Code* (MCC) while the next two or three form the *Mobile Network Code* (MNC) identifying the network operator. The length of the MNC field is a national affair. The remaining (nine or ten) digits are known as *Mobile Subscriber Identification Number* (MSIN) and represent the individual user of that particular operator. Each decimal digit of the SUPI is represented in binary by using the *Telephony Binary Coded Decimal* (TBCD) encoding [10]. The IMEI which uniquely identifies the ME, is a string of 15 digits. If the IMEI is sent in plaintext over the radio interface it could compromise user privacy; however from 4G onwards, the 3GPP specifications prohibit a UE from transmitting the IMEI until after establishment of a secure channel.

2.3 The 5G AKA

Security of communications between mobile subscribers and their service providers requires mutual authentication and key agreement. The 3GPP standard for 5G security specifies two *Authenticated Key Agreement* (AKA) protocols, *Extensible Authentication Protocol AKA'* (EAP-AKA') and 5G-AKA. EAP-AKA' is specified in RFC 5448 [1] while 5G-AKA is detailed in TS 33.501 (cf. sub-clause 6.1.3.2) [11]. The two protocols are mostly identical, we therefore consider only the 5G-AKA further in this paper. The 5G-AKA is instantiated with a set of seven unrelated symmetric key algorithms, denoted as $f_1, \ldots, f_5$, $f_1{}^*$ and $f_5{}^*$. Algorithms f_1, f_2 and $f_1{}^*$ act as message authentication functions, while f_3, f_4, f_5 and $f_5{}^*$ are used as key derivation functions[3]. Detail of how these cryptographic algorithms are used for calculation of various parameters and a pictorial representation of the 5G-AKA can be found in Table 1 and Figure 1 respectively. As already described in Section 2.1, the shared long-term secret key K, sequence number SQN and long-term identity $SUPI$ are stored in both UE and HN during USIM registration. The sequence numbers assure freshness in the 5G-AKA. All key derivation for 5G-AKA is performed using the Key Derivation Function (KDF) specified in 3GPP TS 33.220 [9]. The 5G-AKA protocol works as follows:

0. [4] To initiate authentication, the UE sends the SN either the 5G-GUTI in a "registration request" message or the SUCI as response to an "identifier request" message (See Section 3 for further details).
1. In case of a 5G-GUTI, the SN extracts the corresponding SUPI from its database and forwards it along with its serving network name (SN name) to the HN in an "authenticate request" message. Otherwise the SUCI is sent instead of the SUPI.

[3] The 3GPP documentation uses the term "key generating function" for these algorithms, while these are technically key derivation functions.

[4] This first Step is numbered 0 because its not an exclusive part of the AKA but rather the identification phase.

Table 1: Description of 5G-AKA parameters

Parameter	**Content/Description**
$RAND$	128 bit Random Challenge
SQN	48 bit Sequence Number
AMF	16 bit Authentication Management Field
$SNname$	Serving Network Name
AK	$f_5(K, RAND)$
CK	$f_3(K, RAND)$
IK	$f_4(K, RAND)$
RES	$f_2(K, RAND)$
MAC	$f_1(K, SQN \Vert RAND \Vert AMF)$
$AUTN$	$(SQN \oplus AK \Vert AMF \Vert MAC)$
$RES^*/XRES^*$	$KDF(CK \Vert IK, SNname \Vert RAND \Vert RES/XRES)$
$HXRES^*/HRES^*$	$SHA256(RAND \Vert XRES^*/RES^*)$
K_{AUSF}	$KDF(CK \Vert IK, SNname \Vert SQN \oplus AK)$
K_{SEAF}	$KDF(K_{AUSF}, SNname)$
AV	$(RAND \Vert AUTN \Vert HXRES^* \Vert K_{SEAF})$

2. If the SUCI is received in an authenticate request message by HN, it de-conceals (for details see Section 3.1) the SUPI from it. It further derives the expected response XRES* and generates the authentication vector AV. The AV consists of a random challenge RAND, an authentication token AUTN, a hash of expected response HXRES* and an anchor key K_{SEAF} which is cryptographically bound to the requesting SN.
3. The HN stores XRES*.
4. The HN forwards the 5G AV (RAND, AUTN, HXRES*, K_{SEAF}) in an "authenticate response" message to the SN.
5. The SN forwards RAND, AUTN to the UE in an Auth-Req message.
6. Upon receiving the RAND and AUTN, the UE verifies the freshness and authenticity as described in [8]. It then computes the response RES* and derives the anchor key K_{SEAF} to be used for establishment of the secure channel with the SN.
7. The UE returns RES* in an Auth-Resp message to the SN.
8. The SN then computes the hash of the response HRES* from the received RES* and compares HRES* with XHRES*. If they are equal, the SN considers the authentication successful.
9. The SN then sends RES*, as received from the UE, to the HN in an "authentication confirmation" message (containing the SUPI or SUCI and the serving network name).
10. When the HN receives a confirmation message, it compares RES* with the stored XRES*. If these two are equal, the HN considers the confirmation message as successfully verified.
11. Finally, the HN indicates to the SN in a "confirmation response" message whether the confirmation was successful or not. If the HN received a SUCI from the SN when authentication was initiated, and if the confirmation is successful, then the HN also includes the SUPI in this message.

Lawful Interception. Note that in Step 11 of the 5G-AKA, the HN provides the SUPI of the UE to the SN after successful authentication. This is due to the Lawful Interception (LI) requirements. The law enforcement agencies of almost all countries require that their local service providers should have the capability to locate and track any particular mobile user within the country. The SUPI is later used as an input to the session key derivation function between UE and SN. This ensures that the SUPI value provisioned by the HN is the one claimed by the UE, otherwise the communication breaks down.

3 Identity Confidentiality in 5G

In the 5G system, *Subscription Concealed Identifier* (SUCI) is a privacy preserving identifier containing the concealed SUPI. The UE generates a SUCI using a protection scheme (see Section 3.1) with the public key of the HN that was securely provisioned to the USIM during the USIM registration. Only the MSIN part of the SUPI gets concealed by the protection scheme while the home network identifier (MCC/MNC) gets transmitted in plaintext. The data fields constituting the SUCI are:

- **Protection Scheme Identifier.** This field represents the null scheme[5] or any other specified protection scheme.

[5] The null-scheme is used only if the UE is making an unauthenticated emergency session or if the HN has configured "null-scheme" to be used or if the HN has not provisioned the public key needed to generate SUCI.

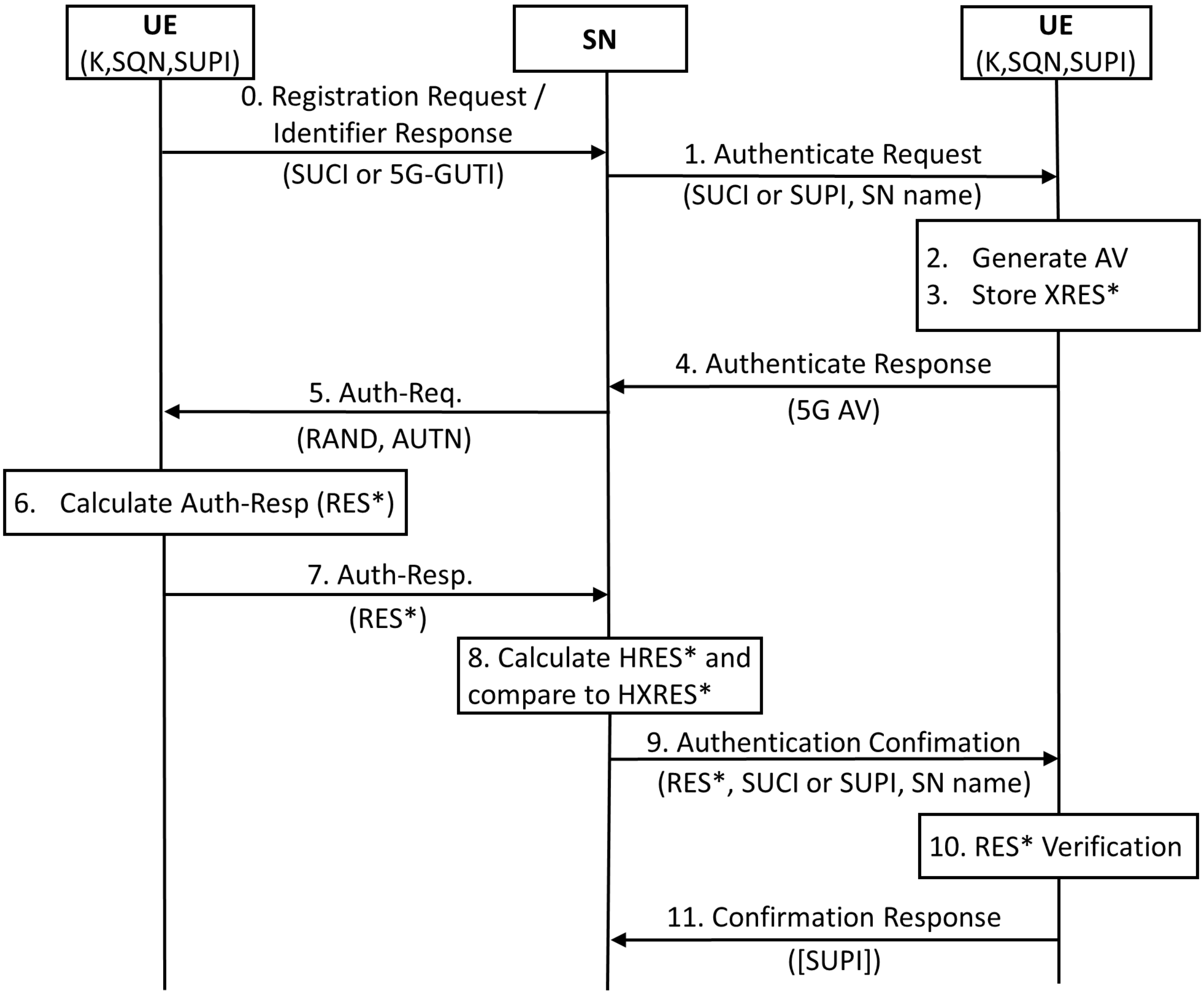


Fig. 1: Overview of the 5G-AKA Protocol

- **Home Network Public Key Identifier.** This represents the public key provisioned by the HN. In case of a null scheme, this field is set to null.
- **Home Network Identifier.** This contains the MCC and MNC part of the SUPI.
- **Protection Scheme Output.** This represents the output of the public key based protection scheme.

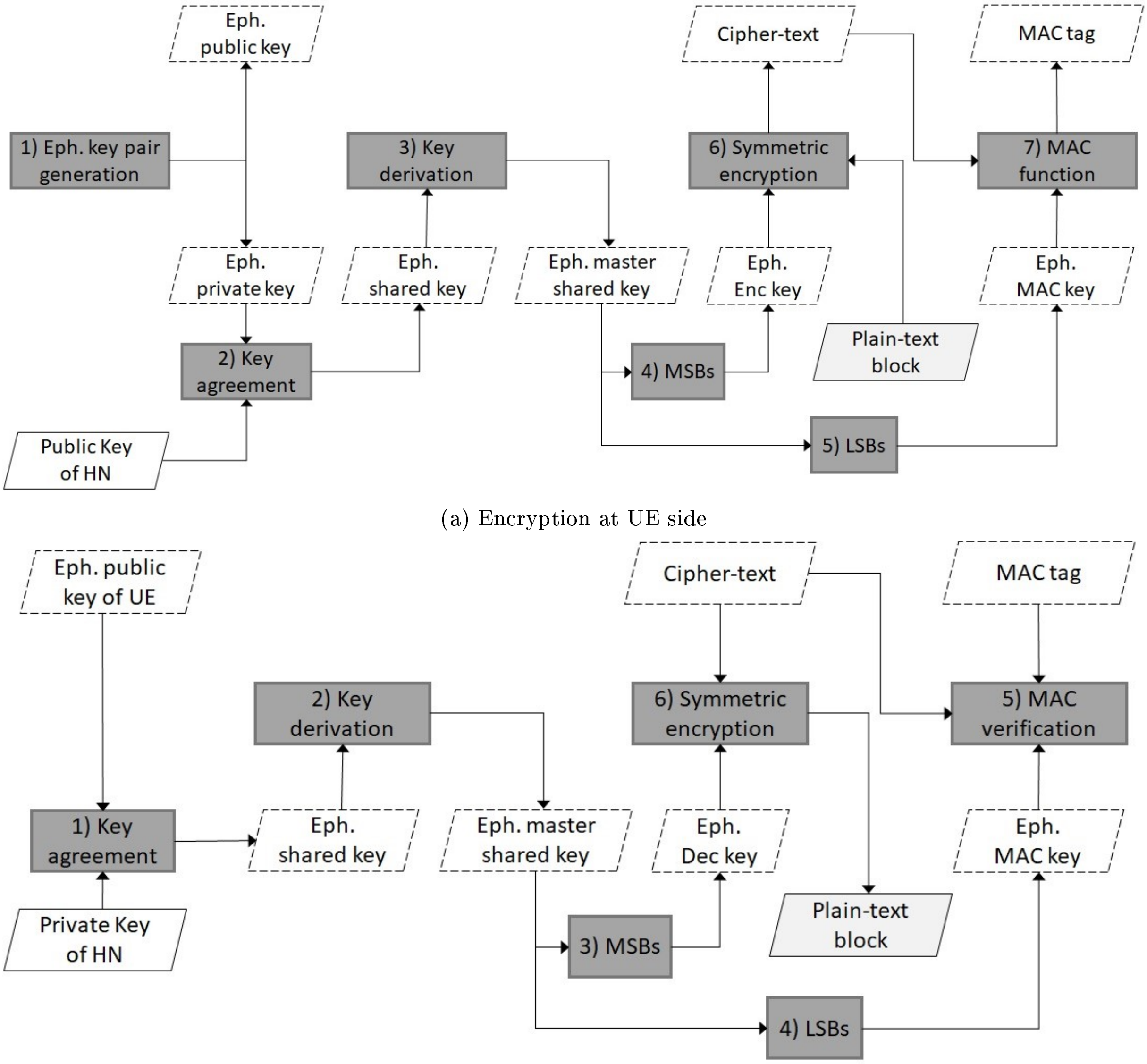
The subscriber identification mechanism allows the identification of a UE on the radio path by means of the SUCI. This mechanism is usually invoked by the SN by sending an **Identifier Request** to the UE, when the UE is not identifiable by means of a temporary identity. The UE then responds with the **Identifier Response**, containing the SUCI. Additionally, if the UE sends a **Registration Request** message of type "initial registration" to a mobile network for which it does not already have a 5G-GUTI, then the UE includes a SUCI to the **Registration Request**.

3.1 ECIES-based Protection Scheme

We now provide an overview of the ECIES-based protection scheme as described in TS 33.501 (cf. Annex C.3) [11]. To compute a fresh SUCI, the UE generates a fresh ECC (Elliptic Curve Cryptography) ephemeral public/private key pair utilizing the HN public key. Processing on the UE side is done according to the encryption operation defined in [19] and as further illustrated in Figure 2a. The final output of this protection scheme is the concatenation of the ECC ephemeral public key, the ciphertext value, the MAC tag value, and any other parameters, if applicable. The HN uses the received ECC ephemeral public key and its private key to deconceal the received SUCI. Processing on the HN side is illustrated in Figure 2b.

3.2 Limitations of the 3GPP Protection Scheme

Although the ECIES-based scheme is oblivious to loss of synchronization between the UE and HN and provisions robust key management, both of which lead to significant reduction in connection failures, there still are aspects which require further improvement [4].



(a) Encryption at UE side

(b) Decryption at HN side

Fig. 2: Detail of ECIES-based Protection Scheme

Post Quantum Vulnerability. As the ECIES-based scheme employs ECC to provision identity confidentiality, it relies on the hardness assumption of the Elliptic Curve Discrete Logarithm Problem (ECDLP). A quantum adversary capable of issuing quantum queries to an appropriate quantum computer can easily break this scheme employing Shor's quantum algorithm [21].

Chosen SUPI Attacks. Any arbitrary third party can always select a SUPI of his choosing and send the corresponding SUCI to the HN. Thereafter the adversary can look out for various responses from the HN, depending on whether the target user is present in that particular cell tower or not. Any noticeable variation in the perceived output would allow the adversary to confirm or deny the presence of the target in that particular cell. There is no mechanism in the ECIES-based scheme to prevent these kind of attacks.

Replay Attacks. Note that the ECIES-based scheme does not have any inherent mechanism to provide freshness guarantees to the HN and is thus susceptible to replay attacks. An adversary can always resend a previously encrypted SUPI to the HN and look out for various kinds of responses (an authentication challenge or a failure message). Based on the received response, a device whose SUPI is unknown to the attacker may be tracked with some confidence.

Bidding Down Attacks. An active adversary simulating a (false) base station can force the UE to use one of the previous generation (3G/4G) and then can get hold of the IMSI using an *identity request* message. Until all systems upgrade to 5G, nothing much can be done about these bidding down attacks. In the current 5G security specifications [11], the SUPI is derived directly from the IMSI, so these bidding down attacks also compromise the SUPI.

Update of HN Public Key. There could be situations which require the HN to have a robust way of quickly updating its public key to subscriber UEs. One such scenario could be a malware attack which tries to recover the home network's private key. Such situations enforce the need to have a quick way of updating the corresponding public keys.

4 Towards Quantum Resistant Identity Confidentiality

We now detail our proposal for an alternative protection scheme. Unlike the ECIES-based scheme, our proposal mostly requires the cryptographic primitives already provisioned by the current 5G specifications. We utilise the previously specified key derivation and message authentication functions of the 5G-AKA for our proposal. Specifically, we use function f_1 for message authentication and functions f_3, f_4, f_5 and ${f_5}^*$ for key derivation. As elaborated in 3GPP TS 33.102, no valuable information can be inferred from the values of any of these functions about other functions [8]. Table 2 gives a summary of notations used in the proposed scheme and Figure 3 provides an overview of the proposed scheme. Various phases of the scheme are explained further.

Table 2: Notation used in the proposed scheme

Notation	Description
A and B	Identification parameters generated by HN
$SQNID$	Counter used for replay prevention
K_{HN}	Long term secret key of HN
K_N	Randomly generated ephemeral parameter
$RANDID$	Freshly generated random number
$CKID$	Confidentiality key
$AKID$	Anonymity key
$MACID$	MAC Tag
f_1	Message Authentication Function
f_3, f_4, f_5, ${f_5}^*$	Key Derivation Functions
AE.Enc	Authenticated Encryption Function
AE.Dec	Authenticated Decryption Function
$f(K, X)$	Execution of keyed-function f upon input X with key K

4.1 System Setup Phase

The HN generates a long-term secret key K_{HN} for the calculation of identification parameters for its subscribers. HN stores this value internally in some secure way, allowing no other entity access. HN randomly chooses K_N during the UE registration and computes the (data) confidentiality key $CKID = f_4(K_{HN}, K_N)$ for the protection scheme as well as identification parameters $A = SUPI \oplus CKID$ and $B = K_{HN} \oplus K_N$. In addition to the $SUPI$, the AKA sequence number SQN and the shared key K (which are all from the original 5G-AKA), the UE also stores identification parameters A and B along with an additional 48 bit identification sequence number $SQNID_{UE}$ with initial value set to 1. HN initialises a corresponding identification sequence number $SQNID_{HN}$[6] with initial value of 0 and stores $SQNID_{HN}$ in its database. An algorithmic description of the computations of HN during this phase can be found below:

```
1. $K_{HN} \xleftarrow{\$} \{0,1\}^\lambda$
2. For each $UE$:
   $K_N \xleftarrow{\$} \{0,1\}^\lambda$
   $SQNID_{UE} \leftarrow 1$
   $SQNID_{HN} \leftarrow 0$
   $CKID \leftarrow f_4(K_{HN}, K_N)$
3. $A \leftarrow SUPI \oplus CKID$
4. $B \leftarrow K_{HN} \oplus K_N$
5. $K \xleftarrow{\$} \{0,1\}^\lambda$
6. $UE \leftarrow (SUPI, K, A, B)$
```

[6] Note that HN will maintain a separate distinct value of $SQNID_{HN}$ for each UE in its database.

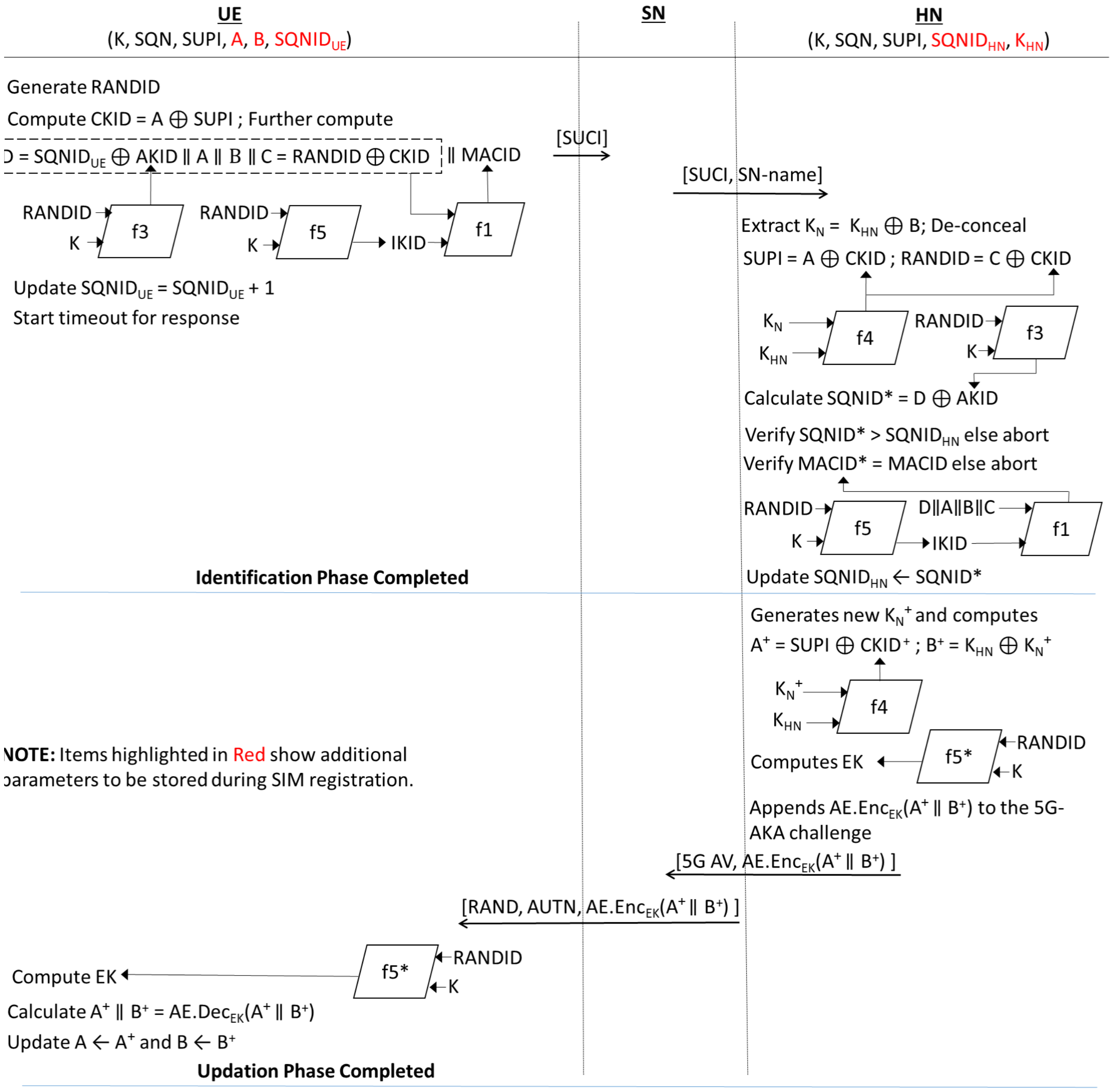

UE
(K, SQN, SUPI, A, B, SQNIDUE)
SN
HN
(K, SQN, SUPI, SQNIDHN, KHN)
Generate RANDID
Compute CKID = A ⊕ SUPI ; Further compute
D = SQNIDUE ⊕ AKID || A || B || C = RANDID ⊕ CKID || MACID
RANDID
K
f3
RANDID
K
f5
IKID
f1
Update SQNIDUE = SQNIDUE + 1
Start timeout for response
[SUCI]
[SUCI, SN-name]
Extract KN = KHN ⊕ B; De-conceal
SUPI = A ⊕ CKID ; RANDID = C ⊕ CKID
KN
KHN
f4
RANDID
K
f3
Calculate SQNID* = D ⊕ AKID
Verify SQNID* > SQNIDHN else abort
Verify MACID* = MACID else abort
RANDID
K
f5
D||A||B||C
IKID
f1
Update SQNIDHN ← SQNID*
Identification Phase Completed
Generates new KN+ and computes
A+ = SUPI ⊕ CKID+ ; B+ = KHN ⊕ KN+
KN+
KHN
f4
NOTE: Items highlighted in Red show additional
parameters to be stored during SIM registration.
Computes EK
f5*
RANDID
K
Appends AE.EncEK(A+ || B+) to the 5G-AKA challenge
[5G AV, AE.EncEK(A+ || B+)]
[RAND, AUTN, AE.EncEK(A+ || B+)]
f5*
RANDID
K
Compute EK
Calculate A+ || B+ = AE.DecEK(A+ || B+)
Update A ← A+ and B ← B+
Updation Phase Completed

Fig. 3: Our Proposed Protection Scheme PQID

4.2 Identification Phase

An algorithmic description of the operations of UE and HN during this phase are presented below. Note that the output of $f_3(K, RANDID)$ is truncated to get a 48 bit $AKID$. The UE prepares the $SUCI = (\texttt{label}_{\texttt{ps}}, \emptyset, \texttt{label}_{HN}, (D\|A\|B\|C\|MACID))$ using various data fields[7] as explained in Section 3 and forwards $SUCI$ to SN. The SN appends its “SN-name” (cf. Clause 6.1.1.4 of [11]) to the received SUCI and forwards the resulting message to HN. Upon successful MAC verification, HN accepts the extracted $SUPI$ as valid for subsequent processing.

Description of UE's operations:

1. $RANDID \xleftarrow{\$} \{0,1\}^{\lambda}$
2. $CKID \leftarrow A \oplus SUPI$
3. $AKID \leftarrow f_3(K, RANDID)$
4. $IKID \leftarrow f_5(K, RANDID)$
5. $C \leftarrow RANDID \oplus CKID$
6. $D \leftarrow SQNID_{UE} \oplus AKID$
7. $MACID \leftarrow f_1(IKID, D\|A\|B\|C)$
8. $SUCI \leftarrow (\texttt{label}_{\texttt{ps}}, \epsilon, \texttt{label}_{HN},$ $D\|A\|B\|C\|MACID)$
9. $SQNID_{UE} \leftarrow SQNID_{UE} + 1$

[7] Note that $\texttt{label}_{\texttt{ps}}$ is a constant value indicating the protection scheme, and $\texttt{label}_{HN}$ is a constant value identifying the HN.

Description of HN's operations:

```
1. $K_N \leftarrow K_{HN} \oplus B$
2. $CKID \leftarrow f_4(K_{HN}, K_N)$
3. $SUPI \leftarrow A \oplus CKID$
4. $RANDID \leftarrow C \oplus CKID$
5. $AKID \leftarrow f_3(K, RANDID)$
6. $IKID \leftarrow f_5(K, RANDID)$
7. $SQNID^* \leftarrow D \oplus AKID$
8. if $SQNID^* \leq SQNID_{HN}$ abort
9. $MACID^* \leftarrow f_1(IKID, D \| A \| B \| C)$
10. if $MACID \neq MACID^*$ abort
11. $SQNID_{HN} \leftarrow SQNID^*$
```

4.3 Update Phase

An algorithmic description of the operations of *UE* and *HN* during this phase can be found below. The output of the encryption scheme $\mathsf{AE.Enc}_{EK}(A^+ \| B^+)$ gets appended to the 5G-AKA authentication vector AV and is forwarded to the *SN* as part of the *authenticate response* message (Step 4 in Figure 1) of the 5G-AKA. The *SN* upon receipt of the response message undertakes the required steps necessary for 5G-AKA and forwards the encrypted identification parameters to the *UE* along with the 5G-AKA authentication challenge parameters *RAND* (note that *RAND* is unrelated to *RANDID*) and *AUTN* (Step 5 in Figure 1).

Algorithmic description of *HN*'s operations in the Update Phase:

```
1. ${K_N}^+ \leftarrow \{0,1\}^\lambda$
2. $CKID^+ \leftarrow f_4(K_{HN}, {K_N}^+)$
3. $A^+ \leftarrow SUPI \oplus CKID^+$
4. $B^+ \leftarrow K_{HN} \oplus {K_N}^+$
5. $EK \leftarrow {f_5}^*(K, RANDID)$
6. $EKID \leftarrow \mathsf{AE.Enc}(EK, A^+ \| B^+)$
```

Algorithmic description of *UE*'s operations in the Update Phase:

```
1. $EK \leftarrow {f_5}^*(K, RANDID)$
2. $A^+ \| B^+ \leftarrow \mathsf{AE.Dec}(EK, EKID)$
3. $A,\ B \leftarrow A^+,\ B^+$
```

5 Security Framework

In this section, we introduce our *Symmetric Updatable Private Authentication* (SUPA) experiment, that follows in the long tradition of standard Bellare-Rogaway key-indistinguishability games. Essentially, a SUPA protocol is a protocol that authenticates an end-user to a central node via a shared symmetric key in a private way. In comparison to similar BR-styled mutual authentication games, our SUPA framework diverges by considering *identity confidentiality.* In particular, the SUPA security experiment asks the adversary to decide which of two parties attempted to authenticate itself to a centralised home network. In addition, SUPA distinguishes itself by considering a *multi-stage* authentication protocol - i.e. subsequent authentication attempts between the UE and the HN (after the first successful authentication) are not independent, but instead dependent on values derived from previous stages. This allows us to capture both *User Identity Confidentiality* and *User Untraceability* from the 3GPP requirements of user privacy. We can now turn to formally defining a SUPA protocol.

Definition 1 (Symmetric Updatable Private Authentication). *A Symmetric Updatable Private Authentication (SUPA) protocol is a tuple of algorithms* $\{\mathsf{SetupHN}, \mathsf{SetupUE}, \mathsf{Identify}, \mathsf{Update}\}$.

- $\mathsf{SetupHN}(\lambda) \rightarrow K_{HN}$*:* SetupHN *takes as input some security parameter* λ *and outputs a long-term symmetric key* K_{HN}*.*
- $\mathsf{SetupUE}(\lambda, K_{HN}) \rightarrow K, st$*:* SetupHN *takes as input some security parameter* λ *and a long-term symmetric key* K_{HN}*, and outputs some shared (between the UE and the HN) secret state st and a shared symmetric key* K*.*
- $\mathsf{Identify}(role, m, st, K_{HN}) \rightarrow (id, m', st')$*:* Identify *takes as input the role of the party in the protocol execution, a (potentially empty) message* m*, the internal state of the party st and (if* $role = HN$*) the long-term HN key* K_{HN}*, and outputs an identifier id, a new (potentially empty) message* m'*, and an updated state* st'*. Note that the identifier id doubles as a failure flag if the* Identify *algorithm is forced to abort.*
- $\mathsf{Update}(role, m, st, K_{HN}) \rightarrow (m', st')$*:* Update *takes as input the role of the party in the protocol execution, a (potentially empty) message* m*, the internal state of the party st and (if* $role = HN$*) the long-term HN key* K_{HN}*, and outputs a new (potentially empty) message* m'*, an updated state* st'*. As in* Identify*, the output message* m' *doubles as a failure flag if the* Update *algorithm is forced to abort.*

5.1 Execution Environment

Here we describe the execution environment of the SUPA security experiment. The experiment $\mathsf{Exp}^{\mathsf{SUPA}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The challenger $\mathcal{C}$ maintains a single *HN*, running a number of instances of the SUPA protocol Π, and a set of (up to) n_N users $UE_1, \ldots, UE_{n_N}$ (representing nodes communicating with the home network HN), each potentially running a single session executing (up to) n_S consecutive stages of Π. The protocol Π is represented as a tuple of algorithms $\mathsf{SUPA} = \{\mathsf{SetupHN}, \mathsf{SetupUE}, \mathsf{Identify}, \mathsf{Update}\}$. We abuse notation and use π_i^s to refer to both the identifier of the s-th stage of Π being run by node UE_i and the collection of per-session variables maintained for this stage. Each session maintains the following set of per-session variables:

- $i \in \{1, \ldots, n_N\}$ - the index of the party UE_i.
- $ltk \in \{0,1\}^\lambda$ - the long-term symmetric secret of UE_i, shared with HN.
- $id \in \{0,1\}^*$ - the identifier of party UE_i.
- $m_s \in \{0,1\}^* \cup \{\perp\}$ - the concatenation of messages sent by the session, initialised by $\perp$.
- $m_r \in \{0,1\}^* \cup \{\perp\}$ - the concatenation of messages received by the session, initialised by $\perp$.
- $st \in \{0,1\}^* \cup \{\perp\}$ - the per-stage secret state of the session, initialised by $\perp$.
- $s \in \{1, \ldots, n_S\}$ - the index of the most recently completed authentication stage, initialised by 1 and increased monotonically.
- $\alpha \in \{\mathtt{active}, \mathtt{accept}, \perp\}$ - the current status of the session, initialised by $\perp$.

Our experiment begins with the challenger $\mathcal{C}$ sampling the random test bit $b \xleftarrow{\$} \{0,1\}$. The challenger generates the long-term symmetric key of the HN K_{HN} and initialises its corruption registers (which maintain the list of secrets $\mathcal{A}$ has leaked). At this point, $\mathcal{A}$ now gains access to the queries listed in Section 5.2 and eventually terminates and outputs a single guess bit b'. If $\mathcal{A}$ has caused the challenger to either execute $\mathsf{Identify}(HN, m, HN.st, K_{HN}) \to (id, m', st')$ such that there exists some session $\pi_i^s.id = id$, but $m \not\subset \pi_i^s.m_s$[8] or; execute $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon) \to (m', st')$ such that $m' \neq \perp$ but there was no execution of $\mathsf{Update}(role^*, m^*, st^*, K_{HN}) \to (m, st'')$. If either of these are true and $\mathsf{fresh}(i,s) = \mathtt{true}$ then $\mathcal{C}$ returns 1. Otherwise, if $\mathcal{A}$ issued a **Test**(i^*, s^*) query, then $\mathcal{C}$ computes $\mathsf{fresh}(i^*, s^*)$. If $\mathsf{fresh}(i^*, s^*)$ is $\mathtt{true}$, then the challenger returns $(b = b')$, otherwise the challenger returns $b^* \xleftarrow{\$} \{0,1\}$.

5.2 Adversary Queries

Here we describe the intuition behind each query that $\mathcal{A}$ has access to during the SUPA experiment. For full details on each of these queries, see Figure 4.

- **Create**(i): Allows $\mathcal{A}$ to initialise a new UE party with shared symmetric state and shared symmetric key with HN.
- **Send**$(i, s, m) \to m'$: Sends a message m to session π_i^s, which updates the per-session variables, returning a (potentially empty) message m'.
- **Corrupt**$(i) \to \pi_i.ltk$: Reveals to $\mathcal{A}$ the long-term symmetric key of UE_i
- **Test**$(i, s, i', s') \to m$: Uses the random bit b sampled by $\mathcal{C}$ to begin a new $\mathsf{Identify}$ phase with either π_i^s (if $b = 0$) or $\pi_{i'}^{s'}$ (if $b = 1$). For ease of notation, we refer to the "test" session as π_b (and the other session as π_{1-b}. Note that $\mathcal{A}$ cannot issue this query if there exists some stage s such that either $\pi_i^s.\alpha = \mathtt{active}$ or $\pi_{i'}^{s'}.\alpha = \mathtt{active}$, nor can $\mathcal{A}$ issue **Send** queries to π_i^s or $\pi_{i'}^{s'}$ until π_b has either accepted or rejected the protocol execution.
- **SendTest**$(m) \to m'$: Allows $\mathcal{A}$ to send a message m to the test session π_b after $\mathcal{A}$ has issued a **Test** query. After $\pi_b.\alpha \neq \mathtt{active}$, then the challenger responds to **SendTest** queries with $\perp$.
- **StateReveal**$(i, s) \to \pi_i^s$: Reveals to $\mathcal{A}$ the full internal state of π_i^s

5.3 Security Definitions

Here we define the security of a Symmetric Updatable Private Authentication Protocols, and additionally show that the PQID protocol described in Figure 3 executes correctly in the presence of a passive adversary.

Definition 2 (Private Authentication Security). *Let Π be a SUPA protocol, and n_N, $n_S \in \mathbb{N}$. For a given cleanness predicate clean, and a PPT algorithm $\mathcal{A}$, we define the advantage of $\mathcal{A}$ in the SUPA game to be:*

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{SUPA},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|.$$

We say that Π is SUPA-secure if, for all $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ is negligible in the security parameter λ.

We also need to define *Identification Correctness* as well as *Update Correctness*, to ensure that we only capture protocols that are actually useful.

[8] Note that here we are using $\subset$ to indicate substrings

$\mathsf{Exp}^{\mathsf{SUPA},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$:

```
$b \xleftarrow{\$} \{0,1\}$
$K_{HN} \xleftarrow{\$} \mathsf{SetupHN}(\lambda)$
$\mathsf{LSKflag}_i, \ldots, \mathsf{LSKflag}_{n_N} \leftarrow \mathtt{clean}$
$\mathsf{PSSflag}_1^1, \ldots, \mathsf{PSSflag}_{n_S}^{n_N} \leftarrow \mathtt{clean}$
$ctr \leftarrow 0$
$b' \xleftarrow{\$} \mathcal{A}^{\mathsf{Send}\star,\mathsf{Create},\mathsf{Corrupt},\mathsf{StateReveal}}(\lambda)$
if $\exists\ (i^*, s^*)$ s.t. $((\mathsf{Identify}(HN, m, HN.st, K_{HN}) \to (id, m', st)$ s.t $\pi_{i^*}^{s^*}.id = id$, $m \neq \pi_{i^*}^{s^*}.m_r) \wedge (\mathsf{clean}(\pi_{i^*}^{s^*}))) \vee ((\mathsf{Update}(UE, m, \pi_{i^*}^{s^*}.st, \epsilon) \to (m', st')$ s.t. $m' \neq \bot$, $\nexists \mathsf{Update}(HN, m^*, HN.st, K_{HN}) \to (m, HNst'))) \wedge (\mathsf{clean}(\pi_{i^*}^{s^*})))$ then
    return $1$
end if
if $\mathsf{clean}(\pi_b) \wedge \mathsf{clean}(\pi_{1-b})$ then
    return $(b' = b)$
else
    return $b' \xleftarrow{\$} \{0,1\}$
end if
```

Test$((i,s),(i',s'))$:

```
if $(\pi_i^s.\alpha = \mathtt{active}) \vee (\pi_{i'}^{s'}.\alpha = \mathtt{active})$ then
    return $\bot$
end if
if $(b = 0)$ then
    $\pi_b \leftarrow \pi_i^s$
    $\pi_{b-1} \leftarrow \pi_{i'}^{s'}$
else
    $\pi_b \leftarrow \pi_{i'}^{s'}$
    $\pi_{b-1} \leftarrow \pi_i^s$
end if
$m \leftarrow \Pi.\mathsf{Identify}(UE, \bot, \pi_b.st, \bot)$
return $m$
```

StateReveal(i,s):

```
if $\pi_i^s.st = \bot$ then
    return $\bot$
end if
$\mathsf{PSSflag}_s^i \leftarrow \mathtt{corrupt}$
return $\pi_i^s.st$
```

Create(λ):

```
$ctr \leftarrow ctr + 1$
$\pi.s \leftarrow 1$
$\pi.ltk, \pi.st \leftarrow \Pi.\mathsf{SetupUE}(\lambda, K_{HN})$
$\pi.i \leftarrow ctr$
return $\pi.i$
```

SendTest(m):

```
$\mathsf{Send}(\pi_b, m) \to m'$
return $m'$
```

Send$(role, i, m)$:

```
if $role = HN$ then
    $(HN.st', m') \leftarrow \Pi.F(\lambda, HN, m)$
end if
let $s = max\{s : \pi_i^s.\alpha \neq \bot\}$
if $\pi_i^s.\alpha \neq \mathtt{active}$ then
    return $\bot$
end if
$\pi_i^s.m_r \leftarrow \pi_i^s.m_r \| m$
$(\pi_i^s, m') \leftarrow \Pi.F(\lambda, \pi_i^s, m)$
$\pi_i^s.m_s \leftarrow \pi_i^s.m_s \| m'$
return $m'$
```

Corrupt(i):

```
$\mathsf{LSKflag}_i \leftarrow \mathtt{corrupt}$
return $\pi_i.ltk$
```

Fig. 4: An algorithmic description of the SUPA security experiment. We assume the existence of a function F that is capable of taking as input a message m and the current internal state $\pi_i^s.st$ of the protocol execution and forwarding the inputs to either Update or Identify as appropriate. We refer to the "test" session in the description of the SUPA experiment as π_b (and the other session as π_{1-b}).

Definition 3 (Identification Correctness). *Let Π be a* SUPA *protocol. We say that Π has* identification correctness *if after an execution of* $\mathsf{Identify}(HN, m', HN.st, K_{HN}) \rightarrow (id', m^*, st')$ *in the presence of a passive adversary $\mathcal{A}$ such that for some session $\pi_i^s.m_s = m'$, then $\pi_i^s.id = id'$.*

It is fairly straightforward to see that the proposed protocol in Figure 3 has identification correctness: The fields $A = SUPI \oplus CKID$ and $B = K_{HN} \oplus K_N$ sent by the UE contains all the information necessary to recompute the identifier $SUPI$ of the UE. HN first computes $K_N = B \oplus K_{HN}$ and then $CKID = f_4(K_{HN}, K_N)$. Retrieving $SUPI$ is then simply a matter of $SUPI \leftarrow A \oplus CKID$.

Update correctness is a little different to identification correctness. We only require that the session executing an Update using output from HN simply updates their state without aborting the protocol execution, instead of having to agree to some shared updated state. This is to capture stateless HN sessions that simply regenerate per-session state when required, usually by processing client-maintained tokens. In this sense, the A and B values sent by the UE during our PQID protocol are tokens that allow HN to recover per-session state.

Definition 4 (Update Correctness). *Let Π be a* SUPA *protocol. We say that Π has* update correctness *if after an execution of* $\mathsf{Update}(UE, m', \pi_i^s.st, \epsilon) \rightarrow (m^*, \pi_i^s.st')$ *in the presence of a passive adversary $\mathcal{A}$ such that for some execution of* $\mathsf{Update}(HN, m, HN.st, K_{HN}) \rightarrow (m', HN.st')$*, then $m^* \neq \perp$ and $\pi_i^s.st' \neq \pi_i^s.st$.*

Similarly to identification correctness, it is straightforward to see that the proposed protocol in Figure 3 has update correctness: The fields $A^+ = SUPI \oplus CKID^+$ and $B^+ = K_{HN} \oplus {K_N}^+$ encrypted under $EK = f_{5^*}(K, RANDID)$ sent by the HN contains all the information necessary to update the values A, B and $CKID$. UE computes $EK = f_{5^*}(K, RANDID)$ (where $RANDID$ was sampled initially by UE and K is the long-term symmetric key shared by UE and HN, so both are known to UE), and decrypts A^+ and B^+. Afterwards, UE updates $A \leftarrow A^+$, $B \leftarrow B^+$, $CKID \leftarrow A^+ \oplus SUPI$.

Finally, we require a cleanness predicate, in order to disallow combinations of **Corrupt** and **StateReveal** queries that allow an adversary to trivially break $SUPA$ security. We do not capture notions of forward secrecy, so our cleanness predicate is very simple: $\mathcal{A}$ is not allowed to break sessions that it has issued either a **Corrupt** or a **StateReveal** query to.

Definition 5 (SUPA-clean). *A session π_i^s in the* SUPA *experiment defined in Figure 4 is* clean *if* $\mathsf{LSKflag}_i \neq \mathtt{corrupt}$ *and* $\mathsf{PSSflag}_i^s \neq \mathtt{corrupt}$.

6 Analysis of the Proposed Protection Scheme

In this section we discuss and analyse our proposed 5G identification scheme within the SUPA security framework, and show that it achieves the notion of *Symmetric Updatable Private Authentication* protocols.

6.1 Formal Analysis

Theorem 1. *The SUPA protocol* PQID *given in Figure 3 is* SUPA*-secure under cleanness predicate* clean *and assuming all hash functions are random oracles. For any PPT algorithm $\mathcal{A}$ against the* SUPA *experiment,* $\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean}}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible under the* $\mathsf{AuthEnc}$, kdf *and* eufcma *security assumptions of the* AE, KDF *and* MAC *schemes, respectively.*

Proof. Before we begin our analysis in earnest, we show that an adversary $\mathcal{A}$ is unable to recover the long-term symmetric-key of the home network K_{HN} (with non-negligible probability) even if $\mathcal{A}$ reveals all long-term secrets K of all nodes and all per-stage secret states st assuming underlying hash functions are random oracles. In our proof we work within the random oracle model, and $\mathcal{A}$ cannot learn anything about K_{HN} from hash outputs $\mathsf{H}(K_{HN}, X)$ (where X is any concatenation of arbitrary values). We turn to $\mathcal{A}$ attempting to learn K_{HN} that has been "blinded" through exclusive-or (XOR) operations, which are only sent in the following values: $B = K_{HN} \oplus K_N$ and $B^+ = K_{HN} \oplus {K_N}^+$. K_N and ${K_N}^+$ are acting as one-time-pads encrypting the long-term symmetric key of the home network HN, and each $K_N/{K_N}^+$ is a value internal to the home network that *cannot* be compromised via $\mathcal{A}$ issuing a **Corrupt** or **StateReveal** query. $\mathcal{A}$ therefore cannot recover K_{HN} in this way, but can attempt to guess and verify the guess by first querying **StateReveal** to any UE party, recovering $CKID$ and B, and querying the random oracle with $({K_{HN}}', B \oplus {K_{HN}}')$ and comparing the output of the random oracle with $CKID$. The probability of $\mathcal{A}$'s success in this strategy is ${}^{q_r}/{}_{2^{\lambda-1}}$. (where q_r is the number of queries that $\mathcal{A}$ makes to the random oracle and λ is the bit-length of K_{HN}). During our analysis then, we assume that in each stage of a protocol execution K_{HN} is indistinguishable from a uniformly-random value ${K_{HN}}^*$ from the same distribution.

In our analysis, we split our proof into three cases:

1. $\mathcal{A}$ has caused a session π_i^s to reach a status $\texttt{accept}$ when calling $\mathsf{Update}(UE,\ m, \pi_i^s.st, \epsilon)$ such that m is not the output of HN and $\mathsf{clean}(\pi_i^s) = \texttt{true}$.
2. $\mathcal{A}$ has caused HN to call $\mathsf{Identify}(HN, m, HN.st, K_{HN}) \rightarrow (id', m', HN.st')$ such that $\exists \pi_i^s.id = id'$, but m was not the output of some $\mathsf{Identify}(UE, \epsilon, \pi_i^s.st, \epsilon)$ and $\mathsf{clean}(\pi_i^s) = \texttt{true}$.
3. $\mathcal{A}$ has output a guessed bit b' after issuing a $\mathbf{Test}(i, s, i', s')$ query

We show that $\mathcal{A}$ has negligible advantage in causing the first two cases to occur, and thus $\mathcal{A}$ also has negligible advantage in winning the SUPA experiment in the third case. Due to space constraints we instead provide a proof sketch.

Case 1. We begin by guessing the session π_i^s such that π_i^s has reached a status $\texttt{accept}$ when calling $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ and m is not an output of the home network HN. Next, we replace the keys $AKID$, $IKID$ and EK computed in the session π_i^s with uniformly-random values $AKID^*$, $IKID^*$ and EK^* from $\{0,1\}^{|\mathsf{KDF}|}$ where $|\mathsf{KDF}|$ represents the output length of KDF, by interacting with a challenger implementing a KDF security game. Finally, we define an abort event $\mathsf{abort}_{\mathsf{dec}}$ that occurs when π_i^s sets $\pi_i^s.\alpha \leftarrow \texttt{accept}$ during a call to $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ and m is not the output of the home network HN. We do this by constructing a simulator $\mathcal{B}$ that interacts with an AE challenger, computing $\mathsf{AE.Enc}(EK^*, A^+\|B^+)$ by querying $(A^+\|B^+,\ A^+\|B^+)$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle instead of computing it honestly. Thus, if $\mathsf{abort}_{\mathsf{dec}}$ occurs, then m is a ciphertext that decrypts correctly by the AE challenger, but was not the output of the query $(A^+\|B^+,\ A^+\|B^+)$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle. Thus, when $\mathsf{abort}_{\mathsf{dec}}$ occurs, $\mathcal{B}$ has broken the ae security of the AE challenger, and we have bound the advantage of $\mathcal{A}$ in causing π_i^s to accept an Update when the received message was not the honest output of the HN.

Case 2. Similarly to Case 1 we begin by guessing the index of the session π_i^s and replacing the keys $AKID$, $IKID$ and EK computed in *any* stage s of the session π_i^s and the HN with uniformly-random values $AKID^*$, $IKID^*$ and EK^*. However, in Case 2 we interact with n_S challengers implementing n_S KDF security games. Finally, we define an abort event $\mathsf{abort}_{\mathsf{mac}}$ that occurs when HN outputs $\pi_i^s.id = id'$ during a call to $\mathsf{Identify}(HN, m, HN.st, K_{HN})$ and m is not the output of some stage s of the sessions owned by UE_i. We do this by constructing a simulator $\mathcal{B}$ that interacts with an MAC challenger, computing $\mathsf{MAC}(IKID^*, D\|A\|B\|C)$ by querying $(D\|A\|B\|C)$ to the MAC challenger instead of computing it honestly within HN or any session owned by UE_i. Thus, if $\mathsf{abort}_{\mathsf{mac}}$ occurs, then $\mathcal{A}$ has managed to produce a MAC tag under a key $IKID^*$ that verifies correctly, but was not the output of a query to the MAC challenger and has broken the eufcma security of the MAC challenger. Thus we have bound the advantage of $\mathcal{A}$ in causing HN to accept an $\mathsf{Identify}$ phase when the received message was not the honest output of the π_i^s.

Case 3. In this case we show that the advantage that $\mathcal{A}$ has in guessing the test bit b is negligible. We begin by guessing the session π_i^s such that $\mathcal{A}$ issues a $\mathbf{Test}(i^*, s^*, i', s')$ query and $\pi_i^s = \pi_b$. Next, we replace the key K_{HN} used in the test session π_i^s with a uniformly random values $K_{HN}{}^*$ from the same distribution $\{0,1\}^\lambda$, following the argument at the beginning of Section 6, incurring a loss of ${}^{q_r}/_{2^{\lambda-1}}$. Following this, we replace the value $CKID^+$ computed in the *previous stage* of the test session π_i^{s-1} with a uniformly-random value $CKID^{+*}$ from $\{0,1\}^{|\mathsf{KDF}|}$ as in the previous cases. Similarly, we replace that the keys $AKID$, $IKID$ and EK computed in the *previous stage* of the test session π_i^{s-1} with uniformly-random values $AKID^*$, $IKID^*$ and EK^*. We now interact with an AE challenger, computing $\mathsf{AE.Enc}(EK^*, A^+\|B^+)$ by querying $(A^+\|B^+,\ A^{+*}\|B^{+*})$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle instead of computing it honestly. At this point, the A^{+*}, B^{+*} values sent in the ciphertext in the previous stage are *independent* of the A^+, B^+ used in the test session. Similarly we replace the value $CKID$, and keys $AKID$, $IKID$ and EK computed in the *test stage* the test session with uniformly-random values via two KDF assumptions. Next, we hide the $SQNID_{UE}$ value with uniformly random value from the same distribution, to prevent linking it with values used in previous stages of the test session. Since $SQNID_{UE}$ is sent as the first field $D = SQNID_{HN} \oplus AKID$ in the $SUCI$ message during the Identification Phase, and by previous replacement of $AKID$ as a uniformly random and independent value, we argue that $AKID$ acts as a one-time-pad perfectly hiding the replaced $SQNID_{UE}$ value. Finally, we interact with an AE challenger, computing $\mathsf{AE.Enc}(EK, A^+\|B^+)$ (sent by the HN to the test session in the *test stage*) by querying $(A^+\|B^+,\ A^{+*}\|B^{+*})$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle. At this point, the A^{+^*}, B^{+*} values sent in the ciphertext are *independent* of the A^+, B^+ used in the *next stage* of the test session. We conclude that all values sent in the tested session π_i^s are independent of any value sent in previous and future sessions, and thus the adversary has negligible advantage in distinguishing between test sessions. Thus we have:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{SUPA,clean}}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) \leq n_N n_S &\cdot \left(\mathsf{Adv}^{\mathsf{KDF}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{AuthEnc}}_{\mathsf{AE},\mathcal{A}}(\lambda)\right) \\ &+ n_N n_S \cdot \left(\mathsf{Adv}^{\mathsf{KDF}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{eufcma}}_{\mathsf{MAC},\mathcal{A}}(\lambda)\right) \\ &+ n_N n_S \cdot \left({}^{q_r}\!/_{2^{\lambda-1}} + 4 \cdot \mathsf{Adv}^{\mathsf{KDF}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{AuthEnc}}_{\mathsf{AE},\mathcal{A}}(\lambda)\right)\end{aligned}$$

Now we discuss how our proposal prevents certain attacks and motivate our proposals to change aspects of the 3GPP specification.

Update of Long-Term Secret Parameters. As elaborated in Section 3.2, it may be required for *HN* to update its long-term secret key. In the current ECIES-based mechanism this is a difficult proposition as it requires a suitable mechanism to transport the updated public key of the *HN* to all of its subscribers and also an update-confirmation mechanism used by the subscribers. With our proposal, no such mechanism is required as the secret key is internal to *HN*. However, updating the K_{HN} will require an interim period during which the *HN* has to operate with both the new and old key, but this would be handled within domains of the Identification Scheme itself.

Migration to Authenticated Encryption in 5G. Our proposal uses authenticated encryption to update identification parameters. Currently, the 3GPP specifications do not list authenticated encryption algorithms, but instead separate encryption and integrity algorithms, ascribed to historical reasons. Previous generations of mobile telephony used to avoid integrity protection of user traffic (voice/data) because of the substantial errors during the radio channel propagation. Only the signalling traffic used to be integrity protected. But as the quality of radio traffic improved, provisions to encrypt user traffic were also created. Though we could have achieved the requisite security guarantees in our scheme using the currently specified primitives by following the “Encrypt-then-authenticate” paradigm, we stress that our approach is clearer and suggest that the 3GPP specifications should introduce such primitives.

Replay Prevention. We include and authenticate sequence numbers *SQNID* in our protection scheme to prevent replay attacks. Moreover, they also provide appropriate resilience to desynchronization between the *UE* and *HN* as now an arbitrary third party cannot initiate an identification request without access to the shared secret key K.

Chosen SUPI Attacks. Our scheme is resilient to *chosen SUPI attacks* (Section 3.2), due to inclusion of the shared secret key K as the keying input for the computation of the MAC tag *MACID*.

Multiple Identification Parameters. In the case of an unexpected interruption, the *UE* will re-attempt identification using the same parameters A and B. Although this does not violate the session unlinkability criterion (as it is effectively the same session), one could imagine the *UE* storing multiple pairs of identification parameters in these cases.

7 Conclusion

In this work we introduced a new private identification scheme for the 5G specification, a quantum-secure alternative to the current public-key based solution. We describe the limitations of the existing solution, and discuss how our proposal mitigates these drawbacks. We introduce a security model for *Symmetric Updatable Private Authentication* protocols, and prove the security of our proposal. We require minimal changes to the current 5G messages, mostly utilise the same underlying cryptographic primitives, with minimal computational overhead. As Release 15 (Phase 1 of 5G specifications planned to be finalized by September, 2018) is not closed yet, now is the correct time to incorporate and evaluate such additions. However, if such inclusions seem difficult in Release 15 then appropriate provisions need to be created in this Release to facilitate such changes in the future Release 16 (Phase 2 of 5G). Such actions would ensure that subsequent migration to quantum-resistant alternatives will be smooth after the 5G infrastructure gets deployed.

For 5G, the most cryptographically relevant quantum algorithms are Grover’s searching algorithm [13] (quadratically faster than any classical brute force searching scheme) and Shor’s factoring algorithm [21] (exponentially faster than the best known classical factoring algorithm - the number field sieve). It is worth noting, however, that if a quantum-resistant alternative was suggested that utilises the symmetric-key primitives offered by the current 3GPP specification (and their associated parameter sizes), then this may not achieve post-quantum security. For example, the output of the MAC algorithm (referred to as f_1, see Table 2) is 64 bits. For such a proposal to realize resilience against quantum algorithms [3], the standard technique to achieve this would be to increase the length of the classical-secure key-size, preferably to 256 bits. As regards the effects of bidding down attacks, in the current 3GPP specifications, the *SUPI* gets derived directly from the IMSI and thus is susceptible to bidding down attacks (Section 3.2) by an active adversary. To thwart such attacks, it is suggested to 3GPP that the derivation of *SUPI* should be independent of the previous generations’ IMSI.

For future work, we suggest a security analysis of the combined 5G-AKA protocol and our proposal in an Authenticated Key Agreement security model. Another interesting direction may be to augment our SUPA security experiment to capture quantum adversaries, to show post-quantum security of our scheme.

References

1. Arkko, J., Lehtovirta, V., Eronen, P.: Improved extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka'). RFC **5448**, 1–29 (2009). https://doi.org/10.17487/RFC5448, https://doi.org/10.17487/RFC5448
2. van den Broek, F., Verdult, R., de Ruiter, J.: Defeating IMSI catchers. In: Ray, I., Li, N., Kruegel, C. (eds.) Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015. pp. 340–351. ACM (2015). https://doi.org/10.1145/2810103.2813615, http://doi.acm.org/10.1145/2810103.2813615
3. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography. US Department of Commerce, National Institute of Standards and Technology (2016)
4. ETSI-SAGE: First response on ECIES for concealing IMSI or SUPI (Oct 2017), https://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=832160
5. Fox, D.: Der imsi-catcher. Datenschutz und Datensicherheit **26**(4) (2002)
6. 3$^{\text{rd}}$ Generation Partnership Project: Rationale and track of security decisions in Long Term Evolution (LTE) RAN / 3GPP System Architecture Evolution (SAE) (3GPP TR 33.821 Version 9.0.0 Release 9) (Jun 2009), http://www.3gpp.org/DynaReport/33821.htm
7. 3$^{\text{rd}}$ Generation Partnership Project: Study on the security aspects of the next generation system (3GPP TR 33.899 Version 1.3.0 Release 14) (Aug 2017), http://www.3gpp.org/DynaReport/33899.htm
8. 3$^{\text{rd}}$ Generation Partnership Project: 3G Security; Security Architecture (3GPP TS 33.102 Version 15.0.0 Release 15) (Jun 2018), http://www.3gpp.org/DynaReport/33102.htm
9. 3$^{\text{rd}}$ Generation Partnership Project: Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)(3GPP TS 33.220 Version 15.2.0 Release 15) (June 2018), http://www.3gpp.org/DynaReport/33220.htm
10. 3$^{\text{rd}}$ Generation Partnership Project: Mobile Application Part (MAP)Specification (3GPP TS 29.002 Version 15.3.0 Release 15) (Mar 2018), http://www.3gpp.org/DynaReport/29002.htm
11. 3$^{\text{rd}}$ Generation Partnership Project: Security Architecture and Procedures for 5G Systems (3GPP TS 33.501 Version 15.0.0 Release 15) (Mar 2018), http://www.3gpp.org/DynaReport/33501.htm
12. 3$^{\text{rd}}$ Generation Partnership Project: System Architecture for the 5G System (3GPP TS 23.501 Version 15.1.0 Release 15) (Mar 2018), http://www.3gpp.org/DynaReport/23501.htm
13. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996). https://doi.org/10.1145/237814.237866, http://doi.acm.org/10.1145/237814.237866
14. Kelly, J.: A Preview of Bristlecone, Google's New Quantum Processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html, [Online; accessed 08-June-2018]
15. Khan, M.S.A., Mitchell, C.J.: Improving air interface user privacy in mobile telephony. In: Chen, L., Matsuo, S. (eds.) Security Standardisation Research - Second International Conference, SSR 2015, Tokyo, Japan, December 15-16, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9497, pp. 165–184. Springer (2015). https://doi.org/10.1007/978-3-319-27152-1_9, https://doi.org/10.1007/978-3-319-27152-1_9
16. Khan, M.S.A., Mitchell, C.J.: Trashing IMSI catchers in mobile networks. In: Noubir, G., Conti, M., Kasera, S.K. (eds.) Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017. pp. 207–218. ACM (2017). https://doi.org/10.1145/3098243.3098248, http://doi.acm.org/10.1145/3098243.3098248
17. Lilly, A.: IMSI catchers: hacking mobile communications. Network Security **2017**(2), 5–7 (2017). https://doi.org/10.1016/S1353-4858(17)30014-4, https://doi.org/10.1016/S1353-4858(17)30014-4
18. Mattsson, J.: Post-quantum cryptography in mobile networks (2017), https://www.ericsson.com/research-blog/post-quantum-cryptography-mobile-networks/
19. SECG SEC 1: Recommended Elliptic Curve Cryptography, Version 2.0. http://www.secg.org/sec1-v2.pdf (2009)
20. Shaik, A., Seifert, J., Borgaonkar, R., Asokan, N., Niemi, V.: Practical attacks against privacy and availability in 4g/lte mobile communication systems. In: 23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016. The Internet Society (2016), http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/practical-attacks-against-privacy-availability-4g-lte-mobile-communication-systems.pdf
21. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 124–134. IEEE Computer Society (1994). https://doi.org/10.1109/SFCS.1994.365700, https://doi.org/10.1109/SFCS.1994.365700
22. Shoup, V.: A proposal for an ISO standard for public key encryption. IACR Cryptology ePrint Archive **2001**, 112 (2001), http://eprint.iacr.org/2001/112