A Post-Quantum Secure Subscription Concealed Identifier for 6G

Vincent Ulitzsch
vincent@sect.tu-berlin.de
Technische Universität Berlin - SECT

Shinjo Park
pshinjo@sect.tu-berlin.de
Technische Universität Berlin - SECT

Soundes Marzougui
soundes.marzougui@tu-berlin.de
Technische Universität Berlin - SECT

Jean-Pierre Seifert
jean-pierre.seifert@tu-berlin.de
Technische Universität Berlin - SECT
Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany

ABSTRACT

5G saw the introduction of an encrypted user identifier, the Subscriber Concealed Identifier (SUCI), to provide confidentiality of the subscriber's whereabouts and identities. The SUCI protects the new generation of cellular networks against tracking devices, so-called IMSI-catchers, which have undermined users' confidentiality ever since the inception of cellular networks. However, the potential advent of large-scale quantum computers in the near future threatens to compromise the confidentiality provided by the SUCI yet again. The security of the public-key cryptography that underpins the SUCI relies on the hardness of the discrete logarithm problem. Using Shor's algorithm, a quantum adversary could break the SUCI's cryptography and once more gain the capability to track and identify users. Advancements in quantum computing are unpredictable, and a breakthrough might be only a decade away. Given the slow nature of standards and their implementation, it is thus necessary to already integrate *now* quantum-resistant cryptography into the current and also next-generation (6G) cellular networks. To contribute to this development, we propose a post-quantum secure scheme for the SUCI calculation, `KEMSUCI`. To this end, we first analyze the weak points in the current SUCI calculation scheme when considering quantum attacks. We then describe an alternative SUCI calculation scheme based on post-quantum secure key-encapsulation mechanisms (KEMs). Our proposed scheme can use any of the KEMs submitted to the NIST call for standardization of post-quantum secure cryptography (PQC) schemes. For the usage in `KEMSUCI`, the KEM should provide efficient execution on a SIM card and induce little network communication overhead. We evaluate all of the NIST PQC finalists under these aspects and identify Kyber and Saber as the best fit. Instantiated with these KEMs, `KEMSUCI` can be integrated into 5G and 6G. Compared to the existing SUPI protection schemes, `KEMSUCI` exhibits faster execution speed and only little communication overhead.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WiSec '22, May 16–19, 2022, San Antonio, TX, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9216-7/22/05…$15.00
https://doi.org/10.1145/3507657.3528540

CCS CONCEPTS

• **Security and privacy** → *Public key encryption*; **Mobile and wireless security**.

KEYWORDS

Key Encapsulation Mechanism, Post-quantum cryptography, SUCI, SUPI, 5G, 6G

ACM Reference Format:
Vincent Ulitzsch, Shinjo Park, Soundes Marzougui, and Jean-Pierre Seifert. 2022. A Post-Quantum Secure Subscription Concealed Identifier for 6G. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3507657.3528540

1 INTRODUCTION

Ever since cellular networks and smartphones evolved, they have become a part of our everyday lives, processing our voice and data communication and tracking our whereabouts. Tracking whereabouts of a cellular subscriber is frequently exploited by IMSI catchers [60]. These tracking devices collect the International Mobile Subscriber Identity (IMSI) – a number uniquely identifying a cellular subscriber – of victims in their vicinity from cellular signalling messages and correlate them into the person's track. This attack is possible even in 4G [55] since the IMSI is sometimes sent in cleartext over the radio interface when authenticating one-self to the base station. To mitigate the threat to the subscriber's privacy and restore confidentiality, 5G introduced the Subscription Permanent Identifier (SUPI) as a superset of the IMSI and also an encrypted subscriber identifier, the Subscription Concealed Identifier (SUCI), calculated by so-called protection schemes. The SUCI can be used in place of the IMSI, to hide the user's identity [4].

In order to calculate the SUCI, public-key cryptography was integrated into the SIM card: The subscriber's device encrypts the SUPI using a hybrid encryption scheme, based on an elliptic curve Diffie-Hellman key-exchange [6]. The complete calculation itself can be done directly in the SIM card or within the phone, using the public key stored in the SIM card.

The SUCI hides the identity and thwarts IMSI-catcher attacks because only the encrypted identifier is sent over the radio interface. However, quantum computers jeopardize the confidentially of the subscribers' identity and tracking data once again: The security of the SUCI relies on the hardness of the discrete logarithm problem.

In the future, large-scale quantum computers could be capable of computing the discrete logarithm and thus pose a looming threat to the SUCI's security.

As such, the potential advent of general-purpose quantum computers puts the guarantees provided by the current SUPI protection scheme and future use in 6G networks at risk. Although it is not clear when or even if quantum computers will ever reach a state that enables them to break discrete logarithm based cryptography, there are strong indications that a break-through will be reached in the succeeding decade [10, 18, 22, 29, 40, 50]. The quantum threat is further aggravated by the design of the SUPI protection schemes: The current SUPI protection schemes do not provide forward secrecy. As a result, breaking an operator's public key is sufficient to de-anonymize all SUPIs encrypted with that public key, without needing to solve additional discrete logarithm problems. Consequently, an adversary can invest large resources into recording SUCIs now and breaking just a small number of operator public keys (and thus de-anonymize all collected SUCIs) later on. Arguably, nation state actors have both the interest and the necessary financial resources to do so.

Standardization bodies like the European ETSI (Telecommunications Standards Institute) and NIST (National Institute of Standards and Technology) acknowledge the urgent need for post-quantum cryptography as well and have already started acting [20]. NIST called for proposals for post-quantum secure cryptographic schemes, i.e., schemes that remain secure even when attacked by a quantum computer, cf. [42]. The NIST call for proposals has recently entered round 3, with various candidates already being eliminated due to design insecurities. Moreover, telecommunication standard bodies designed the SUCI requirements to explicitly allow for additional post-quantum secure protection schemes later on [6] and the currently developed 6G standard aims to be post-quantum secure [31].

To prevent a large-scale deanonymization in the future with 6G networks, there is an urgent need for action. Establishing standards is a slow process. The idea of an encrypted identifier was first proposed in 2015 [62], after which it took three years to establish the concept as a standard, cf. Release 15.4.0 of 3GPP TS 23.003 [4]. Implementing those standards and rolling them out to billions of devices is even harder. For example, while 5G was standardized in 2018 [1], 5G smartphones took only 1% of the market in 2019 and 20% in 2020 [59]. As smartphone vendors are extending their support life-cycle, early 5G devices may remain active in the market for an extended amount of time. These devices may not support algorithms added to the standard later.

Given the slow nature of standards and their implementation, it is thus necessary to integrate quantum-resistant cryptography into the current and also next generation of cellular networks, i.e., 6G networks, now. The SUCI marks the first – and so far only – usage of public-key cryptography in the SIM card. As such, it is an excellent starting point when addressing post-quantum security in telecommunications protocols.

As a first step towards post-quantum secure cellular networks, we propose a post-quantum secure scheme for calculating the SUCI. To this end, we first investigate how the current SUCI calculation scheme succumbs to attacks by quantum computers. Then, we describe an alternative SUCI calculation scheme, `KEMSUCI` based on key-encapsulation mechanisms (KEMs). KEMs use public-key cryptography to securely transmit a shared secret (usually a key for a symmetric cipher) between two parties. Our proposed scheme can use any of the KEMs submitted to the NIST call for standardization.

Our contributions can be summarized as follows:

- We show that the SUCI protection schemes introduced in 5G are vulnerable to quantum attacks (Section 4).
- We propose a new, post-quantum secure, SUPI protection scheme coined `KEMSUCI`. `KEMSUCI` conceals the SUPI based on key-encapsulation mechanisms and provides security equivalent to the current protection schemes (Section 5).
- To choose an appropriate KEM for `KEMSUCI`, we derive requirements and metrics to optimize for when employing cryptography on the SIM card. We structure our results into an evaluation framework that guides us in selecting a suitable KEM. The framework contains concrete thresholds (for example, in terms of maximum memory usage) to assess whether a KEM's execution on a SIM card is at all possible (Section 6).
- We evaluate the NIST standardization "round 3 finalists" with the proposed framework and identify Kyber and Saber as the most suitable KEMs for usage in `KEMSUCI`. With these instantiations, `KEMSUCI` exhibits faster execution speed and only little communication overhead compared to the conventional SUPI protection schemes (Section 7).
- We show that `KEMSUCI` can be seamlessly integrated into the existing 5G standard as an alternative to the already existing protection schemes (Section 8).

2 BACKGROUND

2.1 Cryptography

The security of current asymmetric cryptography rests on the assumptions that integer factorization and (elliptic-curve) discrete logarithm problems are practically infeasible. When leveraging quantum computers, however, these assumptions no longer hold. In his seminal work, Peter Shor proposed polynomial-time quantum algorithms that solve (elliptic curve) discrete logarithm and integer factorization problems [48, 56]. Even so, it is subject to controversy whether Shor's algorithm ever translates into practical insecurity of existing public-key cryptography; there has been scepticism about the possibility of quantum computers being capable of breaking integer factorization and elliptic curve for practical relevant instances in the upcoming decades [22, 32, 51, 52].

Nevertheless, the National Institute of Standards and Technology (NIST) stated that

> "Regardless of whether we can estimate the exact time of the arrival of the quantum computing era, we must begin now to prepare our information security systems to be able to resist quantum computing. [43]"

and has initiated a standardization process for quantum-resistant cryptographic algorithms. The standardization process is expected to specify one or more additional digital signature, public-key encryption, and key-establishment algorithms, that are secure against quantum computer-equipped attackers [43].

In this paper, we leverage key-encapsulation mechanisms submitted to the NIST call for proposals. We introduce the concept in more detail in the following.

Key-Encapsulation Mechanism (KEM). KEMs build on asymmetric cryptography to securely transmit symmetric cryptographic key material. Usually, it is not practical to encrypt long messages with public-key cryptography. KEMs solve this problem by using public-key systems to encrypt short symmetric keys instead. The short keys are in turn used to encrypt long messages by symmetric ciphers. To this end, KEMs provide the means to securely establish a shared secret between two parties leveraging public key encryption; one party generates a shared secret and then encapsulates the secret using another party's public key. A KEM consists of three algorithms:

- $Kem.KeyGen$ is a probabilistic key-generation algorithm that generates a keypair (sk, pk).
- $Kem.Encapsulate$ is a probabilistic encapsulation mechanism that takes the public key pk as input. It proceeds to derive a shared secret k from some initial randomness, encrypts k using public-key cryptography, and then outputs the shared secret k and the corresponding ciphertext c. Note that each invocation of the $Kem.Encapsulate$ function produces a new random shared secret k. Most of the NIST KEMs output a 32-byte shared secret [8, 13, 16], although some allow for variable length-output [54].
- $Kem.Decapsulate$ is the decapsulation algorithm that takes as input a ciphertext c and a secret key sk and either returns a shared secret k or reports a failure.

We note that KEMs allow for the possibility of a decryption failure, in which case the encapsulation process has to be repeated. However, the failure probability is negligible for all NIST finalists and almost all practical purposes. For instance, the NIST finalist `Saber` has a failure probability of merely 2^{-120}.

We will make use of two security notions for KEMs: Indistinguishability under chosen-plaintext attacks (IND-CPA) and indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2). For formal definitions, we refer the reader to standard literature, cf. [17, 34] and only state an intuition here: If a KEM is IND-CCA2 -secure, it withstands attackers that are allowed to make decryption queries. An IND-CPA-secure KEM only needs to be secure against passive attackers that do not have access to a decryption oracle.

2.2 Cellular Network and SUPI/SUCI

In cellular networks up to 4G, each subscriber is identified through their IMSI (International Mobile Subscriber Identity) [4]. The IMSI serves as a permanent identity of a cellular subscriber. Upon registering itself to the network the user's device — also referred to as user equipment (UE) — sends the user's IMSI in-clear over the radio interface. To reduce the over-the-air transmission of the permanent identifier in cleartext, there is a temporary subscriber identity called TMSI (Temporary Mobile Subscriber Identity), which aims to conceal the subscriber's identity. While the TMSI can be interchangeably used with the IMSI in most cases, the TMSI is only issued after identifying the subscriber, which unfortunately requires the IMSI in cleartext. As a result, an external attacker can eavesdrop on these IMSIs [55, 60], identify and track users, or use the IMSIs for further attacks. Prior works have shown the practicality of these attacks with commodity hardware [44] and studied potential defensive mechanisms against commercial IMSI catchers [45].

To address this problem, van den Broek et al. [62] proposed a scheme encrypting the IMSI during initial subscriber identification. This idea was incorporated in 5G as SUPI (Subscription Permanent Identifier) and SUCI (Subscription Concealed Identifier) [4, 6]. The SUPI is a superset of the IMSI and also incorporates other types of cellular network users identifiers. 5G core network components are required to accept the SUCI [5, 6], which is an encrypted SUPI. The encryption procedure is specified through so-called protection schemes. A valid SUCI contains a SUPI type indicator, a home network identifier, a routing indicator, a protection scheme ID, a home network public key ID, and the protection scheme's output. The additional information is required to correctly identify the home network and the public key used to encrypt the SUPI. The home operator needs the information so that the core network can decrypt the SUCI and identify the subscriber successfully. The SUCI generation could be done on either the SIM card or the device itself. The operators' public keys are stored within the SIM card in either case. The introduction of SUPI/SUCI into the cellular network also entails the introduction of public-key cryptography in the cellular network's authentication. This is in contrast to the currently used authentication algorithm AKA (Authentication and Key Agreement) [2], which is based on symmetric key cryptography such as Milenage [3].

An additional defensive mechanism to the SUCI that has been proposed is the introduction of authenticated base-stations [26, 58]. Authenticated base-stations would mitigate active MITM attacks and downgrade attacks, that threaten to undermine the confidentiality provided by the SUCI on a protocol level [14].

3 RELATED WORK

Interest in post-quantum cryptography has been increasing remarkably, not only in academia but also in the industry, e.g., [28, 38, 61]. Post-quantum cryptography aims to substitute the current (quantum computer threatened) cryptography and will also be integrated into the small embedded devices surrounding us in our daily lives, e.g., building automation, or connected driving. To this end, researchers have excessively investigated the applicability of post-quantum cryptography on embedded systems for different use cases.

For example, in [37] the authors evaluated the applicability of the NIST post-quantum schemes to the most fundamental security use cases of embedded systems: secure boot and protection of intermediate keys. They analyzed the requirements stemming from this use case and identified the most fitting post-quantum candidates. Their analysis was followed by a proof-of-concept implementation on an ARM Cortex-R5 development board.

Similarly, Paul et al. focused on the applicability of post-quantum cryptography on the Trusted Platform Modules (TPMs), a chip designed to provide hardware-based and security-related functions in most IoT products. Paul et al. showed how TPMs could facilitate the migration toward post-quantum cryptography. They integrated post-quantum primitives and TPM functionality into the open-source TLS library MbedTLS [47]. They proved the feasibility of the

integration of post-quantum cryptography into mTLS. However, offloading critical computations onto more trustworthy hardware significantly decreases the device's performance.

In [12], Bürstinghaurs et al. were the first to combine post-quantum schemes and embedded systems for a performance evaluation of post-quantum TLS on embedded platforms. Their results were based on four different embedded platforms with three different ARM processors and an Xtensa LX6 processor. The authors showed that using a post-quantum KEM for TLS on embedded devices can result in faster execution compared to the established ECC based protocols [12].

In addition, the industry witnessed trials of post-quantum cryptography. For example, Paul and Scheibe integrated post-quantum primitives into the industrial protocol Open Platform Communications Unified Architecture (OPC UA) [46]. Their approach is compliant with the X.509 standard and provides mutual authentication between client and server. Their results show superior performance across all evaluated security levels in terms of handshake duration compared to conventional OPC UA but come at the expense of increased handshake messages sizes.

Multiple works have touched on the topic of post-quantum security of telecommunication networks [15, 39, 63]. These works provide an overview of the areas in telecommunication protocols that are affected by quantum computers and mention potential cryptographic primitives which can be used for their replacements. To the best of our knowledge though, this work is the first to propose a new protocol to make a specific telecommunication protocol, the SUCI, post-quantum secure.

4 QUANTUM ANALYSIS OF THE SUCI IN 5G

4.1 SUCI Calculation in 5G

5G introduced so-called protection schemes that leverage asymmetric cryptography to encrypt the SUPI into the SUCI, in an attempt to restore confidentiality of the subscriber's identity and location data. Upon registration, the UE picks a protection scheme, encrypts the SUPI accordingly, and then sends the scheme's output together with a protection scheme identifier to the operator. The SUPI protection schemes have withstood classical cryptanalysis so far but they are not post-quantum secure. This section describes the existing protection schemes and shows how they can be attacked with a quantum computer.

The 5G standard currently defines two protection schemes to encrypt the SUPI into the SUCI, in addition to the null cipher [6]. Both schemes rely on the Elliptic Curve Integrated Encryption Scheme (ECIES), a hybrid encryption scheme, to encrypt the SUPI. The schemes only differ in the curve used for ECIES.

The ECIES scheme works as follows: The subscriber's device and the home network first establish a shared secret through Diffie-Hellman. Using a Key Derivation Function (KDF), they derive an ephemeral encryption key from this shared secret and use this key to encrypt the SUPI by a symmetric cipher. Only the encrypted result is sent over the network. Figures 1 and 2 depict the SUPI encryption and decryption procedure, as performed by the subscriber and home operator, respectively. Figure 3 provides pseudocode for the scheme.

In detail, the procedure works as follows:

(1) The operator generates one or more long-term elliptic curve public/private key pairs. The subscriber's Universal Subscriber Identity Module (USIM) stores the operator's public keys. This step is done once.
(2) Upon identifying to the network, the UE generates an ephemeral elliptic curve public/private key-pair. Using Diffie-Hellman, the UE combines the generated private key with the home operator's public key to obtain a shared secret with the home operator.
(3) The UE uses a KDF, evaluated on the shared secret, to generate an ephemeral encryption key for the symmetric cipher, an initial counter block (ICB), and a key for the MAC function.
(4) The UE uses the derived ephemeral encryption key and the ICB to encrypt the SUPI using a symmetric cipher (potentially in counter mode). The MAC ensures the integrity of the ciphertext.
(5) The UE sends, among other information, its ephemeral public key, the resulting encrypted SUPI, and the MAC-tag over the air.
(6) Using the ephemeral public key and its own private key, the home operator also derives the shared secret. The operator uses the KDF to derive the necessary symmetric key material, verifies the integrity of the ciphertext and finally decrypts the SUCI into the SUPI.

Note that the resulting SUCI is different on each invocation of ECIES, since the UE generate a new ephemeral key pair each time.

The two protection schemes use Curve25519 and secp256r1, respectively. For both curves, the best known attack takes around 2^{128} classical operations. Other than the curves, the two schemes share the instantiations of the cryptographic primitives, namely:

Symmetric cipher Both schemes use AES-128 in counter mode as the symmetric cipher.

Key Derivation Function The keys are derived by the ANSI-X9.63 key derivation function, which builds on top of SHA-1 or SHA-2 [9]. The ANSI-X9.63 KDF allows deriving variable-length output from a shared secret value.

MAC The MAC is in both cases specified as HMAC–SHA-256 with a key length of 256 bits.

4.2 Post-Quantum Security of the SUCI Calculation

The current SUCI protection schemes do not withstand attacks with quantum computers. Shor's algorithm renders "all" cryptography based on the hardness of (elliptic curve) discrete logarithm problems, such as Diffie-Hellman, insecure. Moreover, the current protection schemes derive the shared secret k using a long-term public key instead of using an ephemeral public key also on the operator's side. As a result, the protocol does not provide forward secrecy [24], and one invocation of Shor's algorithm is sufficient to de-anonymize many (recorded) SUCIs.

Such a large-scale de-anonymization attack would proceed as follows:

(1) Given an operator's public key $H_A = d_A \cdot P$ and the curve's base point P, an adversary equipped with a quantum computer can simply solve the discrete logarithm problem to

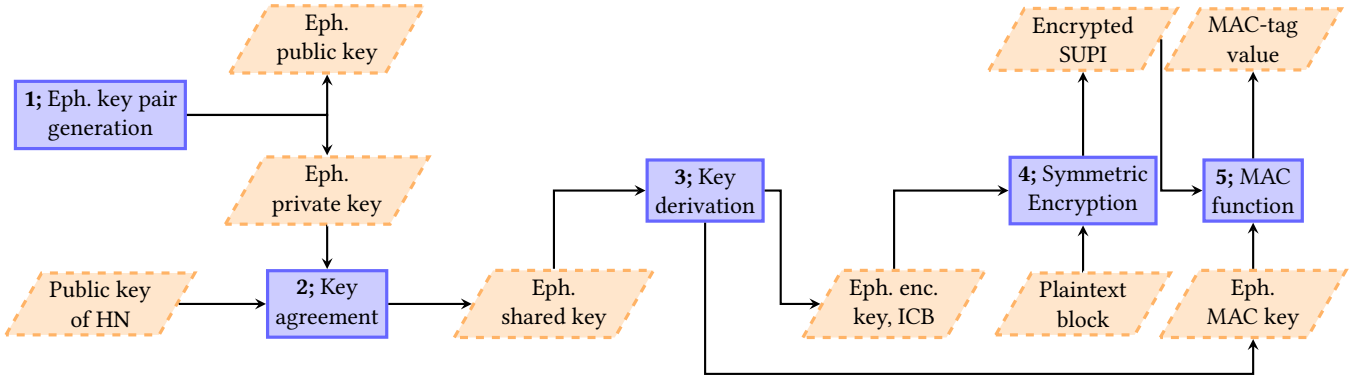


Figure 1: ECIES-based SUCI calculation at the UE [6]. Step 1 and 2 are realized via a Diffie-Hellman Key Agreement.

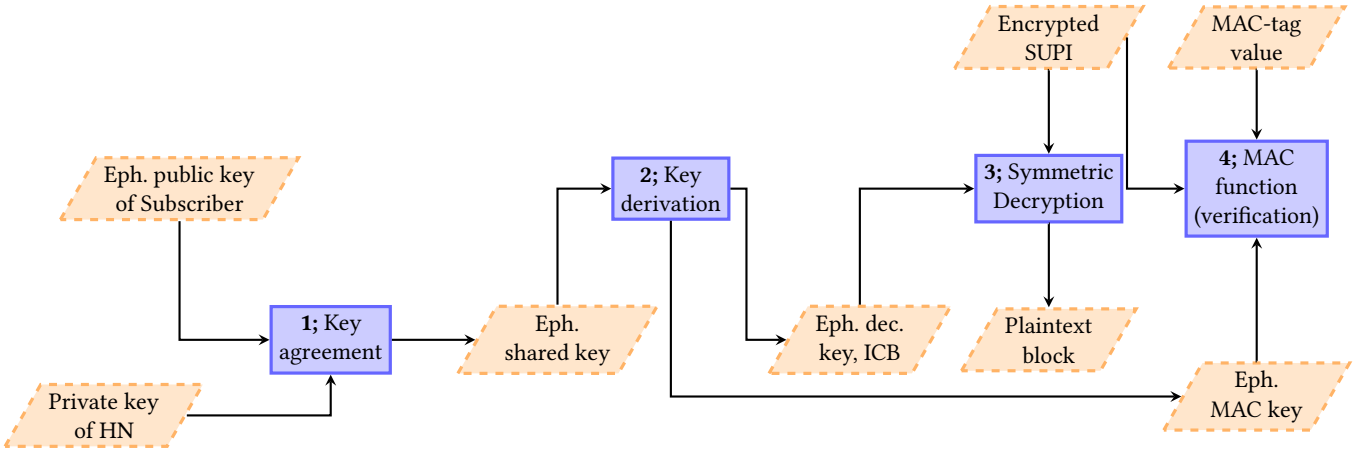


Figure 2: ECIES-based SUCI decryption at home network [6]

UE (pk, SUPI)	Home Network (d_B, $d_A \cdot P$, SUCI, t)
$d_A, d_A \cdot P \leftarrow DH.Gen()$	
$k \leftarrow DH.Shared(d_A, d_B \cdot P)$	$k \leftarrow DH.Shared(d_B, d_A \cdot P)$
$(K_{ENC}, ICB, K_{MAC}) \leftarrow KDF(k)$	$(K_{ENC}, ICB, K_{MAC}) \leftarrow KDF(k)$
$SUCI \leftarrow Encrpt(K_{ENC}, ICB, \texttt{SUPI})$	if $Mac.Ver(K_{MAC}, SUCI, t) = \text{True}$
$t \leftarrow MAC.Sign(K_{MAC}, SUCI)$	$SUPI \leftarrow Decrpt(K_{ENC}, ICB, \texttt{SUCI})$
return $(d_A \cdot P, SUCI, t)$	

Figure 3: The SUCI calculation scheme as described in [5]. The functions `MAC.Sign`, `MAC.Ver` perform MAC tag creation and verification and the functions `Encrypt` and `Decrypt` perform symmetric encryption and decryption. `DH.Gen()` and `DH.Shared()` describe the standard Diffie-Hellman functions with respect to some base point P, that generate a public-private keypair and return a shared secret respectively.

obtain the operator's private key d_A. Note that this only needs to be done once.

(2) Then, the adversary needs to capture a packet containing a subscriber's ephemeral public key $H_B = d_B \cdot P$ and the SUPI encrypted using the broken public key H_A. Such a packet is transferred over radio when registering to the network, for example. This packet is sufficient to de-anonymize the subscriber: Given the captured packet, the adversary can follow the decryption steps at the home network side to obtain a user's SUPI. The only confidential information required to decrypt the SUCI is the respective home operator's private key, which the attacker obtained in step (1).

The quantum threat is particularly severe for the SUCI calculation for two reasons: First, as shown above, the SUCI calculation protocol does not provide forward secrecy: As a result, breaking an operator's public key is sufficient to de-anonymize all SUPIs encrypted with that public key, without needing to solve additional discrete logarithm problems. This reduces the cost and requirements of a large-scale de-anonymization considerably, since an attacker only needs to break a small amount of public keys with a quantum computer. Adversaries can record SUCIs and their corresponding locations now in the hope of de-anonymization through quantum computers at a later point in time. Second, solving the elliptic curve discrete logarithm on a quantum computer requires a much lower number of qubits than solving the integer factorization problem. To calculate the discrete logarithm of a point on Curve25519, as little as 2330 qubits are sufficient [52], while very recent and additional quantum bit savings could be applied as well [19].

5 A POST-QUANTUM SECURE SUCI CALCULATION METHOD

As shown in Section 4.2, the current SUPI protection schemes are not quantum resistant. To secure cellular networks against quantum adversaries, we propose a modification of the existing SUPI protection schemes. The main idea is to replace the Diffie-Hellman algorithm of the protection scheme described in Section 4 with a post-quantum secure KEM. As the symmetric cryptography parts of the current SUCI calculation schemes do not succumb to quantum adversaries, replacing the Diffie-Hellman step with such a KEM should warrant post-quantum security of the SUCI. There are various possible choices for post-quantum secure KEMs, since the NIST standardization process encompasses a call for post-quantum secure KEMs. All "round 3" finalists were already subject to heavy scrutiny by the cryptographic research community, which provides reasonable assurance in the security of their design.

Our post-quantum secure protection scheme, which we call `KEMSUCI`, works as follows:

(1) Each operator generates a public/private key-pair using $Kem.KeyGen$ once. The resulting public key pk is stored on the subscriber's USIM.
(2) To encrypt its SUPI, the UE uses the function $Kem.Encapsulate(pk)$ to generate some initial randomness, derive the shared secret k from this randomness and compute a ciphertext c, representing the encapsulated shared key.
(3) After that, the UE follows the current SUCI calculation scheme, as described in Section 4: The UE expands the shared secret k using a KDF, to obtain a secret key for the symmetric cipher, an ICB, and and a key for the MAC. After that, the UE encrypts the SUPI using the derived symmetric key. A MAC-tag verifies the integrity of the encrypted SUPI. The subscriber's device then shares the KEM output c, amounting to the encrypted shared secret k, the encrypted SUPI, and the respective MAC tag with the home operator.
(4) The home operator uses the $Kem.Decapsulate(c, sk)$ function to retrieve the shared secret k from the KEM output c. In case of a decapsulation failure, the UE can easily retry the encapsulation process. A decapsulation failure does not induce more performance overhead than a transmission failure and is mainly of theoretical interest since the probability of a decryption failure is negligible for all NIST finalists, as pointed out in Section 2.1.
(5) After obtaining the shared secret, the home operator follows the same procedure as the subscriber to obtain the decrypted SUPI.

Figure 4 depicts the calculation of the SUCI on the subscriber side and Figure 5 depicts the steps taken by the home operator to retrieve the SUPI from the SUCI. Figure 6 provides pseudocode for `KEMSUCI`.

Clearly, replacing the Diffie-Hellman step with a KEM does not reduce the security of the original protocol, as long as the KEM is secure. The calculated SUCI is different on each invocation, as is the case for ECIES based protection schemes. To provide 128-bit quantum security, we recommend the following instantiation:

KEM We strongly conjecture that an IND-CPA-secure KEM is sufficient for `KEMSUCI` to be secure. However, for the sake of brevity, we skip a formal proof here and require an IND-CCA2-secure KEM instead. This does not limit our choices, though: All NIST Round 3 finalists are IND-CCA2-secure already. Additionally, the KEM should preserve the same level of security in a quantum setting that the ECIES protection scheme provides in a classical setting. To this end, `KEMSUCI` requires that it should take at least 2^{128} (potentially quantum) operations to retrieve the encapsulated secret given the KEM ciphertext and KEM public key.

Symmetric Cipher The symmetric cipher should provide at least 128 bits of security. For a key-space of size N, Grover's algorithm entails a quantum attack that can retrieve the secret key in $O(\sqrt{N})$ operations, cf. [23, 30]. Thus, to achieve 128 bits of post-quantum security, AES-256 should be used instead of AES-128.

KDF To expand the shared secret k, we can use the ANSI-X9.63 key derivation function that is also used in ECIES.

MAC The MAC should also provide 128 bits of security. This is already the case for HMAC-SHA256, the HMAC used in ECIES.

6 KEM EVALUATION

All NIST Round 3 finalists claim to provide the required IND-CCA2 security. However, besides this security guarantee, the KEM for `KEMSUCI` also needs to meet the requirements that follow from executing algorithms in resource-constrained environments, such as smart cards. Moreover, cellular network protocols should be optimized for low latency, as bandwidth on the radio layer is only available in limited capacity. These two goals guide the selection process for KEMs. To identify KEMs suited for usage in `KEMSUCI`, we structure the above requirements and optimization priorities in an evaluation framework.

6.1 Evaluation Framework

To evaluate which KEM is suitable for usage in the `KEMSUCI` protocol, we analyze each scheme with respect to speed, RAM, embedded flash storage, and energy consumption requirements. In doing so, we mainly optimize for two things: That the KEM can encapsulate efficiently on a standard SIM card and that the communication overhead induced by the KEM is minimal. We capture these requirements in an evaluation framework consisting of a) a categorization of the different performance metrics into four different priorities and b) thresholds for certain metrics that if exceeded will prevent a KEM from being considered at all. The different categories are:

- ▲ **Essential**: A metric is considered essential if it is critical for the successful operation of the entire `KEMSUCI` protocol.
- ■ **Important**: A metric is important if it ensures good performance on the SIM card.
- ● **Nice-To-Have**: A good performance here is not a requirement for the KEM to be considered but could tip the balance in favor of the specific KEM.
- ○ **Negligible**: The metric is neglected for the use-case at hand.

As a minimum requirement, a standard SIM card must be able to run the encapsulation algorithm. This entails thresholds for certain metrics, e.g., the KEM's memory usage. As a point of reference, we will use the SLC 14MCO256 controller, a 32-bit SIM card controller

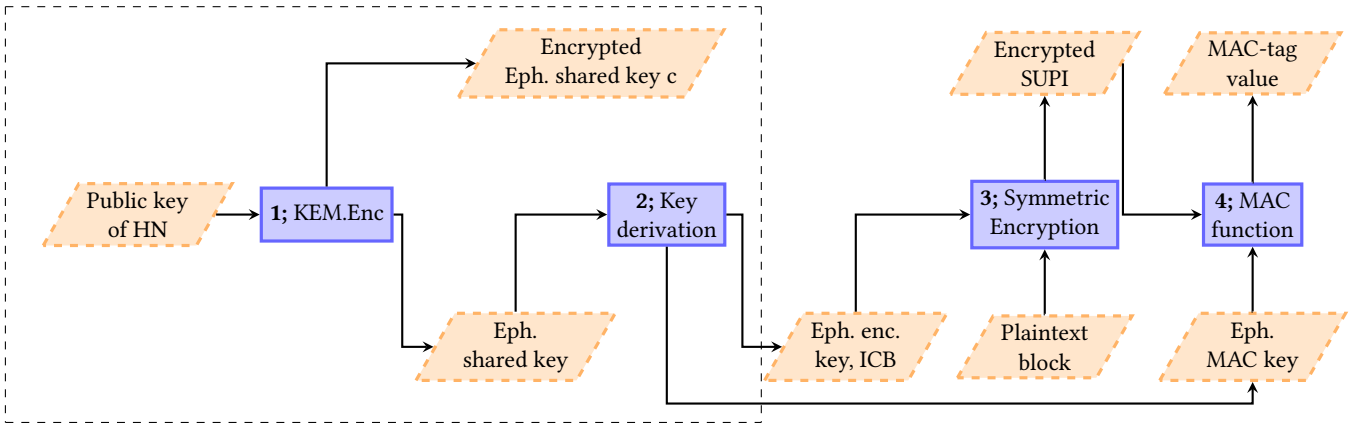


Figure 4: SUCI calculation at UE for KEMSUCI. The dashed box highlights changes to the ECIES protocol.

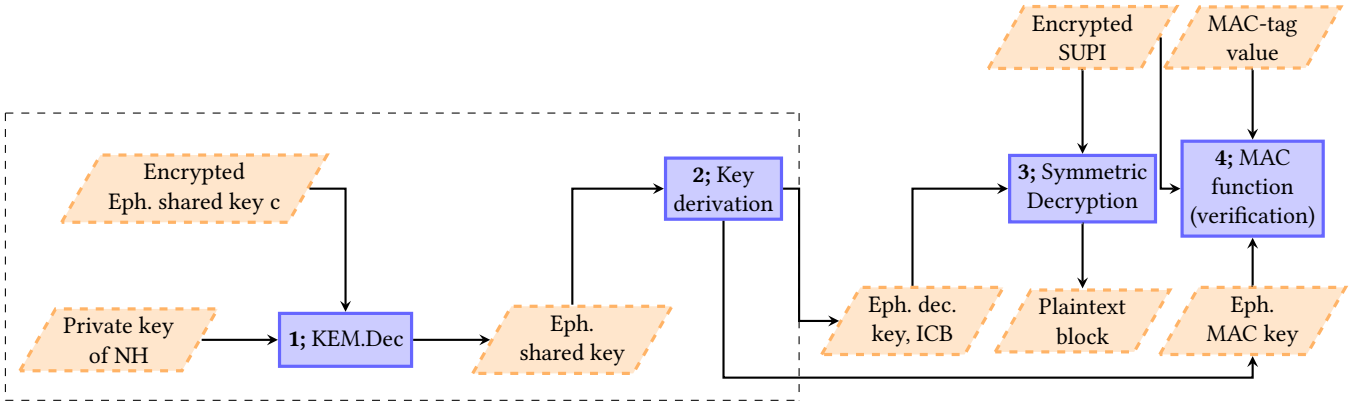


Figure 5: SUCI calculation at home network. The dashed box highlights changes to the ECIES protocol.

UE (pk, SUPI)	Home Network (sk, c, SUCI, t)
$(k, c) \leftarrow Kem.Enc(pk)$	$k \leftarrow Kem.Dec(sk, c)$
$(K_{ENC}, ICB, K_{MAC}) \leftarrow KDF(k)$	$(K_{ENC}, ICB, K_{MAC}) \leftarrow KDF(k)$
$SUCI \leftarrow Encrpt(K_{ENC}, ICB, \texttt{SUPI})$	if $Mac.Ver(K_{MAC}, SUCI, t) =$ True
$t \leftarrow MAC.Sign(K_{MAC}, SUCI)$	$SUPI \leftarrow Decrpt(K_{ENC}, ICB, \texttt{SUCI})$
return $(c, SUCI, t)$	

Figure 6: The `KEMSUCI` protocol. The functions `MAC.Sign`, `MAC.Ver` perform MAC tag creation and verification and the functions `Encrypt` and `Decrypt` perform symmetric encryption and decryption.

produced by Infineon. The SLC 14MCO256 controller has 256 kByte embedded flash storage and 10 kByte of RAM available [27]. Thus, we do not consider any scheme that requires more than 256 kByte flash storage and more than 10 kByte RAM for usage in `KEMSUCI`.

Assuming that the security provided by the schemes indeed matches their claimed security level, the most apparent differences between the schemes lie in the following metrics, which we categorize into the above-mentioned priorities:

Low ciphertext size (Essential): The ciphertext which encapsulates the shared secret k has different sizes depending on the KEM. We consider a small ciphertext size integral, since the ciphertext needs to be sent over-the-air once per connection attempt and cellular network protocols should minimize latency. 3GPP sets the upper limit of the protection scheme's length to at most 3000 bytes, citing the future usage of quantum-resistant protection scheme as a reason [6].

Low storage size of the code (Negligible): Smart cards commonly use Read-Only Memory (ROM) to store executable code. This metric can safely be neglected as ROM is available in a large capacity.

Small size of the public key (Nice-To-Have): In contrast to the code itself, smart cards typically store public keys in embedded flash memory [1]. We prefer a small public key as standard smart cards have only limited embedded flash sizes. Nevertheless, state-of-the-art SIM cards offer a generous amount of embedded flash, reducing the importance of this metric. The public key still must not exceed 256 kByte, the amount of embedded flash memory available on the SLC 14MCO256.

[1] It would also be possible to store the public key in ROM. However, this would require changes in the supply chain structure of the SIM card manufacturers — the public key would need to be embedded within the ROM already during the silicon production process. Moreover, compromised private keys would entail a large recall action instead of a simple over-the-air update.

Small size of the private key (**Negligible**): The private key is stored on the operator's infrastructure side (AuC/HLR), where we can assume sufficient storage to be available.

Few cycles required for key generation (**Negligible**): The key generation is only done once per home operator, and then the public key is given to all SIM cards. We thus do not need to optimize for a fast key generation.

Few cycles required for encapsulation of the KEM (**Important**): The KEM encapsulation is performed on the phone or the SIM card, which has limited computing power. Good user experience necessitates a fast encapsulation function.

Few cycles required for decapsulation of the KEM (**Nice-to-Have**): The decapsulation is done on the network side, where we can assume much more computing power than on the SIM card. However, for scalability reasons, the computing time should still be limited.

Low memory usage of key generation (**Negligible**): This metric is negligible since the key generation is only done once per operator.

Low memory usage of encapsulation (**Important**): Random access memory (RAM) is very limited on smart cards, and the KEM encapsulation is executed on the SIM card, once per connection attempt. A low memory usage when encapsulating is thus essential to allow the schemes to run on a SIM card. The Infineon SLC 14MCO can only execute KEM schemes that use a maximum of 10 kByte of RAM during encapsulation.

Low memory usage of decapsulation (**Negligible**): The decapsulation is done on the network side, where we can assume sufficient memory to be available.

Energy consumption of encapsulation (**Nice-To-Have**): The encapsulation is done in the phone or the SIM card, which has limited battery capacity. However, as encapsulation is not expected to be run frequently on the device, we assume that the impact of extra power consumption compared to the whole energy consumption of the smartphone is not significant.

Table 1 and 2 summarize the maximum allowed thresholds and the metric priorities. Our evaluation metrics and their assigned priorities suggest picking a scheme that fulfills:

(1) Small ciphertext size — at most 3000 bytes.
(2) High speed and low memory usage during encapsulation. The memory usage must not exceed 10 kByte.
(3) Small public key size. The public key size must not exceed a maximum of 256 kBytes.

Metrics	Thresholds
Low ciphertext size	3000 bytes
Small size of public key	256 kBytes
Low memory usage of encryption	10 kBytes

Table 1: Metrics Thresholds

6.2 Evaluation of the NIST Round 3 Finalists

We evaluate all NIST Round 3 finalist KEMs for usage in KEMSUCI. The NIST Round 3 finalists are: The code-based KEM Classic McEliece, and three lattice-based KEMS: CRYSTALS-KYBER, NTRU, and SABER.

The KEM can be configured with different parameter choices, depending on the desired security level. We will compare all schemes with respect to the configuration that provides NIST security level 1. Schemes at NIST security level 1 should yield post-quantum security equivalent to the classical security of the ECIES based protocol: If a KEM is configured in NIST Security Level 1, breaking the KEM on a (potentially quantum) computer should be at least as hard as a key search for AES-128 on a classical computer [42]. Higher security levels correspond to worse performance and are not required for our use-case.

Out of the four NIST Round 3 finalists, we eliminate McEliece immediately. McEliece has a public key size of 256 kByte, which would take up all the embedded flash memory available on the SLC 14MCO256. This would leave no space for other applications/data, rendering McEliece unfit for KEMSUCI. We thus choose to compare only the three remaining candidates, Kyber, Saber, and NTRU according to the metrics described above. Table 3 provides an overview of the three schemes and their performance according to the above evaluation metrics. The encapsulation speed, memory usage, and code size are based on benchmarks performed on an ARM Cortex M4 chip executing the reference implementations [33].For energy consumption estimates, we rely on numbers reported by [53], as measured on an STM32 Nucleo-64 development board, which uses an ARM Cortex M4.

For the decapsulation operations, we list the cycles required on an Intel Haswell, with AVX2 instructions enabled, since the decapsulation is performed on the operator's side. We refer to the numbers stated within the submission documents of the respective KEMs [13, 16, 54]. Note that the NTRU submission document only states the number of decapsulation cycles for a Haswell without AVX2 instructions [13].

The evaluation shows that the SLC 14MCO25 microcontroller can only execute LightSaber and Kyber-512, since the NRTU KEM uses more than 10 kB memory. The scheme which strikes the overall best balance seems to be LightSaber. It has a relatively small ciphertext size (736 bytes), and the fastest encryption. The memory usage is still within an acceptable range and does not exceed the defined threshold (Table 1). Alternatively, Kyber-512 provides similar performance. NRTU has the smallest ciphertext size but a prohibitively large memory usage. NRTU optimizations tailored towards low memory usage might allow for execution on a SIM card — we leave a final evaluation for future work.

In summary, according to our metrics and priorities, Saber and Kyber are well suited for KEMSUCI. Implementations tailored to the present use case could reveal further optimizations favoring our prioritized metrics. These optimizations could tip the balance towards either Kyber or Saber. For example, KEMSUCI could also use an IND-CPA version of Saber, Dagger (developed in [25]), which offers reduced public key and ciphertext sizes. On the other hand, the target device could also influence the decision: For example, Kyber's lower energy usage might be more critical on IoT devices. We recommend integrating both schemes into the 5G and 6G standard for now.

Metrics	Negligible	Nice-To-Have	Important	Essential
Low ciphertext size				▲
Low storage size of code		●		
Small size of the public key		●		
Small size of the private key	○			
Few cycles for encapsulation			■	
Few cycles for decapsulation		●		
Low memory usage of key generation	○			
Low memory usage of encapsulation			■	
Low memory usage of decapsulation	○			
Low energy consumption of encapsulation		●		

Table 2: Evaluation metrics with their assigned priorities

NIST KEM Candidates	Parameter Configuration	Implementation	Ciphertext size (Bytes) ▲	Speed of *Kem.Enc* (Cycles) ■	Memory usage of *Kem.Enc* (Bytes) ■	Storage size of the code (Bytes) ●	Public key size (Bytes) ●	Energy usage *Kem.Enc* (μJ) ●	Speed of *Kem.Dec* on Haswell (Cycles) ●
Kyber	Kyber-512	m4	768	551,681	2,300	10,700	800	533.953	34,572
Saber	LightSaber	m4fspeed	736	481,006	6,284	18,900	672	559.310	61,000
NTRU	ntruhps2048509	m4f	699	563,397	14,068	191,656	699	395.641	1,940,870

Table 3: Overview of different KEM schemes and their performance.

Side-Channel Resistance. Smart cards, such as the SIM card, are generally subject to fault injection and (power) side-channel attacks [35]. Hence KEMSUCI and the KEM implementations need to be evaluated under this aspect. When attacking the KEMSUCI protocol while executed on the SIM card, an attacker aims at targeted de-anonymization. This is because the SUPI is the only long-term secret processed when executing KEMSUCI on the UE. We highlight the difference to other cryptographic algorithms run on the SIM card: Milenage implementations susceptible to power side-channel attacks would allow for SIM card cloning [36]. De-anonymization can be achieved through timing and power side-channel or fault injection attacks:

Time-based side-channel information, gathered while the SIM card executes KEMSUCI, could reveal the SUPI. The KEMSUCI implementation should therefore be constant-time. The implementations evaluated in the present study do claim (so far, undisputed) resistance against timing attacks and can thus be used as-is [16, 54].

The attacker could also mount side power side-channel or fault injection attacks. We distinguish two scenarios: Attacks that require immediate physical access to an unlocked SIM card and attacks that can be executed from close proximity, such as attacks based on electromagnetic (EM) emanations [7]. Protection against the former is not required: With physical access, the SIM card's SUPI can simply be read out by sending AT commands to the SIM with a smart card reader. Attacks from close proximity constitute a more serious threat to the SUPI's confidentiality. Masking — a dedicated countermeasure against power side-channel attacks [49] — can mitigate this threat, but KEM implementations that integrate masking suffer from a performance impact. For instance, a first order masked version of Kyber768 takes 2,978,441 cycles for an encapsulation operation on M4, while the reference implementation only takes 876,197 cycles. The reference implementation of both Kyber and Saber lack those dedicated countermeasures and consequentially succumbed to power side-channel attacks [41, 57]. It remains to be evaluated whether side-channel attacks can also be mounted against SIM cards in close proximity and to what extent masking is required.

7 COMPARISON WITH THE EXISTING SCHEMES

We compare KEMSUCI, instantiated with either Kyber or Saber, to the already existing protection schemes in terms of output size of the scheme, public key size, and cycle count needed until the shared secret is computed on the SIM card (which is after encapsulation or a full Diffie-Hellman operation, respectively).

Table 4 summarizes the comparison. The output size refers to the size of the package send to the network when communicating the SUCI. The output size of KEMSUCI is:
`Encapsulating Ciphertext Size + 64-bit MAC + length of the SUPI`
where the SUPI is usually 15 bytes long. The output size of KEMSUCI-LightSaber is therefore: 736 + 64 + 15 = 815
The output size when using `Kyber-512` is: 768 + 64 + 15 = 847.

While the output size of KEMSUCI is considerably larger compared to the ECIES based schemes, KEMSUCI does not introduce additional messages to communicate the SUCI. We thus expect the impact on the network latency to be minimal. Comparing the KEMSUCI to the ECIES based schemes in other dimensions is non-trivial: Speed, memory usage, and energy consumption of both the elliptic curve operations as well as the encapsulation operations depend highly on the employed optimizations in the implementation, hardware support for underlying mathematical operations, and the target platform. Nevertheless, they are strong indications that KEMSUCI, instantiated with Saber or Kyber, is faster than the established SUPI protection schemes: Reported cycle counts on the M4, measured for both optimized x25519 as well as optimized secp256r1 implementations, are much higher than the cycles required for encapsulation in Kyber or Saber [21, 53]. For instance, the authors of [21] report that a full x25519 operation needs 894,391

cycles, while Kyber-512 only needs 551,681 cycles. This is line with other research; for example, the authors of [12] have found the Kyber-KEM to be faster than its Diffie-Hellman counterparts in their application as well. Surprisingly, the energy usage of Kyber and Saber was measured to be higher than elliptic-curve based algorithms [53]. Further research and experiments conducted on a SIM card are needed for a final comparison of Diffie-Hellman and KEM-based approaches on the SIM card.

Protection Schemes	Scheme Output Size (Bytes)	Public Key Size (Bytes)	Cycle Count Until Secret Establishment
KEMSUCI-LightSaber [33]	815	672	481,006
KEMSUCI-Kyber-512 [33]	847	800	551,681
ECIES-Curve25519 [21]	111	32	894391
ECIES-secp256r1 [53]	112	33	11,630,000

Table 4: Comparison of the new protection schemes against the old protection schemes in terms of public key size, ciphertext size and cycles required to establish a shared secret on the SIM card. The table lists the references that state cycle counts for implementations on an ARM Cortex M4.

8 INTRODUCING THE POST-QUANTUM SECURE SUPI PROTECTION SCHEME IN 5G

Given that the inception and roll-out of 6G might be more than a decade away, post-quantum security should be implemented in 5G already, whenever possible. In the case of the SUCI, the ETSI specification for 5G security allows for this via the protection scheme identifier [6]. Each method to calculate the SUCI is assigned a unique protection scheme identifier so that an entity receiving a SUCI can choose the correct scheme to process it. Right now, there are three potential protection schemes: The null-scheme, which does not perform any encryption, and the ECIES schemes using either Curve25519 or secp256r1. The standard reserves further identifiers for future standardized protection schemes. When communicating the SUCI over the radio network, the payload also contains the protection scheme identifier, indicating the protection scheme used for the SUCI calculation. We propose to expand the current 5G standard by two additional protection schemes. Both schemes should follow the SUCI calculation procedure as described in Section 5, where one scheme uses `LightSaber` as the KEM (`KEMSUCI`-LightSaber), and the other scheme uses `Kyber-512` (`KEMSUCI`-Kyber512). Note that `KEMSUCI` re-uses most of the cryptographic components that are already implemented in the current SUPI protection schemes. Thus, integrating `KEMSUCI` into SIM cards with SUCI support is mostly a matter of extending code that implements the already existing SUPI protection schemes to accommodate KEMs.

9 CONCLUSION & OUTLOOK

Quantum computers threaten to break the security promises of the 5th generation of cellular networks. One specific instance is the confidentiality of the user's identity and location, which the SUCI aims to protect. A few invocation of Shor's algorithm could be sufficient to de-anonymize all subscribers of a given operator. Given the slow nature of standard bodies and the fact that new changes need to be rolled to billions of devices, often with limited update infrastructure, the migration to post-quantum secure alternatives needs to commence now. Our paper is a first step in this direction. We design and describe a post-quantum secure alternative to the so-far singular use of public-key cryptography in SIM cards. Our proposed scheme, `KEMSUCI`, hinges on a KEM with efficient encapsulation on a smart card and small ciphertext size. We show that Kyber-512 and Lightsaber fulfill the requirements and provide a structured approach to address the selection of post-quantum secure algorithms for a given use case. We believe our evaluation framework can also support the selection of post-quantum cryptographic algorithms for different applications.

By keeping our changes to the original SUCI calculation minimal, we can seamlessly integrate the `KEMSUCI` as a new protection scheme in the already existing 5G standard. This facilitates a timely roll-out, which we recommend to happen soon.

Outlook. Integration of a quantum-resistant method to calculate the SUCI would already be a significant step towards post-quantum security of radio communication in cellular networks, but by no means mitigate the threat posed by quantum computers completely. Although the SUCI is the only use of public-key cryptography in the SIM card, the SIM card relies heavily on symmetric-key cryptography to authenticate and authorize itself to the network. Ensuring that the symmetric cryptography is post-quantum secure might turn out to be non-trivial: Recent work on quantum cryptanalysis disproves the common belief that doubling the key size is sufficient to make symmetric-key cryptography post-quantum secure, cf. [11].

Quantum cryptanalysis of the symmetric key cryptography used in cellular networks is thus necessary to secure the next generation of telecommunication networks against quantum computers. Moreover, new generations of cellular networks witnessed the introduction of IP-based communications protocols (e.g., in VoWifi), or communication between different entities of the telecommunications network. Some of these protocols rely heavily on public-key cryptography. Thus, they need to be replaced by post-quantum secure alternatives as well. For instance, in 5G, most of the communication in the core network is now secured via TLS as supported by HTTP/2, which is not post-quantum secure [6]. Given the unpredictability of further advancements in quantum computing and the tremendous amount of work to be done, we see the urgency of a) investigating the security of telecommunications protocols against quantum attacks and b) designing new post-quantum secure alternatives.

ACKNOWLEDGMENTS

The work described in this paper has been supported by the German Federal Ministry of Education and Research (BMBF) under the project Full Lifecycle Post-Quantum PKI - FLOQI (ID 16KIS1074) and by the Einstein Research Unit "Perspectives of a quantum digital transformation: Near-term quantum computational devices and quantum processors" of the Berlin University Alliance. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-RIC, project identification number: 16KISK030.

REFERENCES

[1] 3GPP. 2019. *Release description; Release 15.* Technical Report (TR) 21.915. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/21915.htm Version 15.0.0.

[2] 3GPP. 2020. *3G security; Security architecture.* Technical Specification (TS) 33.102. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/33102.htm Version 16.0.0.

[3] 3GPP. 2020. *3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm specification.* Technical Specification (TS) 35.206. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/35206.htm Version 16.0.0.

[4] 3GPP. 2021. *Numbering, addressing and identification.* TS 23.003. 3rd Generation Partnership Project. http://www.3gpp.org/dynareport/23003.htm Version 17.4.0.

[5] 3GPP. 2021. *System architecture for the 5G System (5GS).* Technical Specification (TS) 23.501. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/23501.htm Version 17.3.0.

[6] 3GPP. 2022. *Security architecture and procedures for 5G System.* Technical Specification (TS) 33.501. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/33501.htm Version 17.4.2.

[7] Dakshi Agrawal, Bruce Archambeault, Josyula R Rao, and Pankaj Rohatgi. 2002. The EM side—channel (s). In *International workshop on cryptographic hardware and embedded systems*. Springer, 29–45.

[8] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. 2020. *Classic McEliece.* Technical Report. National Institute of Standards and Technology. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[9] X9 ANSI. 1998. 63: Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography. *American National Standards Institute* (1998).

[10] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[11] Xavier Bonnetain, Gaëtan Leurent, María Naya-Plasencia, and André Schrottenloher. 2021. Quantum linearization attacks. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 422–452.

[12] Kevin Bürstinghaus-Steinbach, Christoph Krauß, Ruben Niederhagen, and Michael Schneider. 2020. Post-Quantum TLS on Embedded Systems: Integrating and Evaluating Kyber and SPHINCS+ with Mbed TLS. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (Taipei, Taiwan) *(ASIA CCS '20)*. Association for Computing Machinery, New York, NY, USA, 841–852. https://doi.org/10.1145/3320269.3384725

[13] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. 2020. *NTRU.* Technical Report. National Institute of Standards and Technology. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[14] Merlin Chlosta, David Rupprecht, Christina Pöpper, and Thorsten Holz. 2021. 5G SUCI-catchers: still catching them all?. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 359–364.

[15] T Charles Clancy, Robert W McGwier, and Lidong Chen. 2019. TUTORIAL: Post-Quantum Cryptography and 5G Security.. In *WiSec'19: ACM Conference on Security and Privacy in Wireless and Mobile Networks*.

[16] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. 2020. *SABER.* Technical Report. National Institute of Standards and Technology. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[17] Alexander W Dent. 2003. A designer's guide to KEMs. In *IMA International Conference on Cryptography and Coding*. Springer, 133–151.

[18] Oliver Dial, Jerry Chow, and Jay Gambetta. 2021. IBM quantum breaks the 100-qubit processor barrier. https://research.ibm.com/blog/127-qubit-quantum-processor-eagle

[19] Martin Ekerå. 2021. Quantum algorithms for computing general discrete logarithms and orders with tradeoffs. *Journal of Mathematical Cryptology* 15, 1 (2021), 359–407. https://doi.org/doi:10.1515/jmc-2020-0006

[20] ETSI. 2020. ETSI releases migration strategies and recommendations for Quantum-Safe schemes. https://www.etsi.org/newsroom/press-releases/1805-2020-08-etsi-releases-migration-strategies-and-recommendations-for-quantum-safe-schemes

[21] Hayato Fujii and Diego F Aranha. 2017. Curve25519 for the Cortex-M4 and beyond. In *International Conference on Cryptology and Information Security in Latin America*. Springer, 109–127.

[22] Roger A Grimes. 2019. *Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto.* John Wiley & Sons.

[23] Lov K Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 212–219.

[24] Christoph G Günther. 1989. An identity-based key-exchange protocol. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 29–37.

[25] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Ralf Zimmermann. 2020. Post-quantum WireGuard. *IACR Cryptol. ePrint Arch.* 2020 (2020), 379.

[26] Syed Rafiul Hussain, Mitziu Echeverria, Ankush Singla, Omar Chowdhury, and Elisa Bertino. 2019. Insecure connection bootstrapping in cellular networks: the root of all evil. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 1–11.

[27] Infineon. 2022. Product Brief: SLC 14 – 65nm Innovation for SIM Cards. https://www.infineon.com/dgdl/SLC+14+Product+Brief+-+65nm+Innovation+for+SIM+Cards+(2013).pdf?fileId=5546d46149b40f650149d256d791045c

[28] Infineon. 2022. World's first post-quantum cryptography on a contactless security chip. https://www.infineon.com/cms/en/product/promopages/post-quantum-cryptography/

[29] Intel Corporation. 2019. Intel introduces 'horse ridge' to enable commercially viable quantum computers. https://newsroom.intel.com/news/intel-introduces-horse-ridge-enable-commercially-viable-quantum-computers/#gs.ngaylt

[30] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. 2020. Implementing Grover oracles for quantum key search on AES and LowMC. *Advances in Cryptology–EUROCRYPT 2020* 12106 (2020), 280.

[31] DongHyun Je. 2021. Towards 6G Security: Technology Trends, Threats, and Solutions. https://research.samsung.com/blog/Towards-6G-Security-Technology-Trends-Threats-and-Solutions

[32] Gil Kalai. 2020. The Argument against Quantum Computers, the Quantum Laws of Nature, and Google's Supremacy Claims. *arXiv preprint arXiv:2008.05188* (2020).

[33] Matthias J Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. 2019. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. (2019).

[34] Jonathan Katz and Yehuda Lindell. 2020. *Introduction to modern cryptography*. CRC press.

[35] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Annual international cryptology conference*. Springer, 388–397.

[36] Junrong Liu, Yu Yu, François-Xavier Standaert, Zheng Guo, Dawu Gu, Wei Sun, Yijie Ge, and Xinjun Xie. 2015. Small tweaks do not help: Differential power analysis of milenage implementations in 3G/4G USIM cards. In *European Symposium on Research in Computer Security*. Springer, 468–480.

[37] Soundes Marzougui and Juliane Krämer. 2019. Post-Quantum Cryptography in Embedded Systems. In *Proceedings of the 14th International Conference on Availability, Reliability and Security* (Canterbury, CA, United Kingdom) *(ARES '19)*. Association for Computing Machinery, New York, NY, USA, Article 48, 7 pages. https://doi.org/10.1145/3339252.3341475

[38] Microsoft. 2022. Cryptography in the era of quantum computers. https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/

[39] Chris J Mitchell. 2020. The impact of quantum computing on real-world security: A 5G case study. *Computers & Security* 93 (2020), 101825.

[40] Michele Mosca. 2018. Cybersecurity in an Era with Quantum Computers: Will We Be Ready? *IEEE Security & Privacy* 16 (09 2018), 38–41. https://doi.org/10.1109/MSP.2018.3761723

[41] Kalle Ngo, Elena Dubrova, and Thomas Johansson. 2021. *Breaking Masked and Shuffled CCA Secure Saber KEM by Power Analysis.* Association for Computing Machinery, New York, NY, USA, 51–61. https://doi.org/10.1145/3474376.3487277

[42] NIST. 2017. *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process.* Technical Report. National Institute of Standards and Technology (NIST), Washington, D.C. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

[43] NIST. 2021. Post-Quantum Cryptography - CSRC, NIST. https://csrc.nist.gov/projects/post-quantum-cryptography

[44] Ivan Palamà, Francesco Gringoli, Giuseppe Bianchi, and Nicola Blefari-Melazzi. 2021. IMSI catchers in the wild: A real world 4G/5G assessment. *Computer Networks* 194 (2021), 108137.

[45] Shinjo Park, Altaf Shaik, Ravishankar Borgaonkar, and Jean-Pierre Seifert. 2019. Anatomy of commercial IMSI catchers and detectors. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*. 74–86.

[46] Sebastian Paul and Patrik Scheible. 2020. Towards Post-Quantum Security for Cyber-Physical Systems: Integrating PQC into Industrial M2M Communication. In *Computer Security – ESORICS 2020*, Liqun Chen, Ninghui Li, Kaitai Liang, and Steve Schneider (Eds.). Springer International Publishing, Cham, 295–316.

[47] Sebastian Paul, Felix Schick, and Jan Seedorf. 2021. TPM-Based Post-Quantum Cryptography: A Case Study on Quantum-Resistant and Mutually Authenticated TLS for IoT Environments. In *The 16th International Conference on Availability,*

Reliability and Security (Vienna, Austria) *(ARES 2021)*. Association for Computing Machinery, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/3465481.3465747

[48] John Proos and Christof Zalka. 2004. Shor's discrete logarithm quantum algorithm for elliptic curves. arXiv:quant-ph/0301141 [quant-ph]

[49] Emmanuel Prouff and Matthieu Rivain. 2013. Masking against side-channel attacks: A formal security proof. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 142–159.

[50] Rigetti Computing. 2021. Rigetti computing announces next-generation 40Q and 80Q Quantum Systems. https://www.globenewswire.com/news-release/2021/12/15/2352647/0/en/Rigetti-Computing-Announces-Next-Generation-40Q-and-80Q-Quantum-Systems.html

[51] Yosef Rinott, Tomer Shoham, and Gil Kalai. 2020. Statistical aspects of the quantum supremacy demonstration. *arXiv preprint arXiv:2008.05177* (2020).

[52] Martin Roetteler, Michael Naehrig, Krysta M Svore, and Kristin Lauter. 2017. Quantum resource estimates for computing elliptic curve discrete logarithms. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 241–270.

[53] Markku-Juhani O Saarinen. 2020. Mobile energy requirements of the upcoming NIST post-quantum cryptography standards. In *2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 23–30.

[54] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. 2020. *CRYSTALS-KYBER*. Technical Report. National Institute of Standards and Technology. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[55] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. 2016. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. In *23rd Annual Network and Distributed System Security Symposium, NDSS San Diego, California, USA, February 21-24, 2016*.

[56] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (Oct 1997), 1484–1509. https://doi.org/10.1137/s0097539795293172

[57] Bo-Yeon Sim, Aesun Park, and Dong-Guk Han. 2021. Chosen-ciphertext Clustering Attack on CRYSTALS-KYBER using the Side-channel Leakage of Barrett Reduction. Cryptology ePrint Archive, Report 2021/874. https://ia.cr/2021/874.

[58] Ankush Singla, Rouzbeh Behnia, Syed Rafiul Hussain, Attila Yavuz, and Elisa Bertino. 2021. Look before you leap: Secure connection bootstrapping for 5g networks to defend against fake base-stations. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. 501–515.

[59] Statista. 2021. Forecast 5G-enabled smartphone shipments as share of total smartphone shipments worldwide from 2019 to 2023.

[60] Daehyun Strobel. 2007. *IMSI-Catcher*. Technical Report. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.397.8140&rep=rep1&type=pdf

[61] TÜVit. 2022. Post-Quantum Cryptography: IT Security in the Era of Quantum Technology. https://www.tuvit.de/en/innovations/post-quantum-cryptography/

[62] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. 2015. Defeating IMSI Catchers. *CCS '15* (2015), 340–351. https://doi.org/10.1145/2810103.2813615

[63] Jing Yang and Thomas Johansson. 2020. An overview of cryptographic primitives for possible use in 5G and beyond. *Science China Information Sciences* 63, 12 (2020), 1–22.