**3GPP TSG-SA3 Meeting #124 S3-253844-r2**

**Wuhan, China, 13 – 17 October 2025**

**Source: Samsung**

**Title: Pseudo-CR on SUPI Concealment using Hybrid shared Key**

**Document for: Approval**

**Agenda item: 5.2.1**

**Spec: 3GPP TR 33.703**

**Version: v0.1.0**

**Work Item: FS\_CryptoPQC**

**Comments**

This pCR provides the proposed changes for hybrid shared key based solution combining ECC and PQC cryptography methods for SUPI protection key issue.

\* \* \* First Change \* \* \* \*

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[xx] 3GPP TR 33.938: "3GPP Cryptographic Inventory (Release 19)"

[yy] 3GPP TS 33.501: "Security architecture and procedures for 5G system (Release 19)".

[zz] 3GPP TS 23.003: "Numbering, addressing and identification".

[aa] NIST FIPS 203: “Module-Lattice-Based Key-Encapsulation Mechanism Standard”. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf>

[bb] NIST FIPS 204: “Module-Lattice-Based Digital Signature Standard”. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf>

[cc] NIST FIPS 205: “Stateless Hash-Based Digital Signature Standard”. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf>

[dd] 3GPP TS 23.003: "Numbering, addressing and identification".

[ee] Active Internet-Draft: Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3

<https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/>

[ff] Active Internet-Draft: Hybrid key exchange in TLS 1.3

<https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/>

[gg] SECG SEC 1: Recommended Elliptic Curve Cryptography, Version 2.0, 2009. Available <http://www.secg.org/sec1-v2.pdf>

[ff] SECG SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0, 2010. Available at <http://www.secg.org/sec2-v2.pdf>

[hh] IETF RFC 7748: "Elliptic Curves for Security".

[ii] RFC 9794: “Terminology for Post-Quantum Traditional Hybrid Schemes”.

\* \* \* Next Change \* \* \* \*

### 7.2 Solutions

Editor’s Note: This clause contains solutions to update 3GPP defined security protocols (for example SUCI calculation) to use the appropriate PQC algorithm, if those protocols are not expected to be updated by other SDOs to use PQC algorithms.

### 7.2.X Solutions to Protocol #1: SUCI calculations

Editor’s Note: If only SUCI calculation is considered, this subclause may be removed. If other protocol, e.g. MIKEY-SAKKE is studied, this subclause is used for each of such protocol identified.

#### 7.2.X.Y Solution #1 to Protocol #1: SUPI Concealment using Hybrid shared Key

Editor’s Note: Details on KDF inputs are FFS.

Editor's Note: The pros and cons (including security, complexity and efficiency) of combining traditional asymmetric cryptographic algorithms with post-quantum cryptographic algorithms for SUCI calculation is FFS.

Editor’s Note: Why to use an ad-hoc KEM combiner instead of adding a standard KEM combiner is FFS.

Editor’s Note: Detailed profiles needs to update later including other options.

##### 7.2.X.Y.1 Introduction

Replacing classical cryptography with PQC algorithms at an early stage carries an inherent risk as a first time widespread deployment and more rigorous testing of PQC algorithms may be needed. So it will be beneficial to have it integrated with classical asymmetric cryptography based security mechanisms via a hybrid approach, where both classical asymmetric algorithms and post-quantum algorithms coexist. The main objective of a hybrid shared key generation mechanism is to enable the creation of a secure shared secret that remains protected as long as at least one of its underlying key exchange components remains uncompromised. In case vulnerabilities are found in either type of algorithm, the presence of both classical and post-quantum algorithms in a hybrid setup reduces the impact of potential breaches, providing additional resilience to the overall cryptography.

##### 7.2.X.Y.2 Solution details

##### 7.2.X.Y.2.1 Processing on UE side

The Hybrid shared key generation scheme is implemented such that for computing a fresh SUCI, the UE uses the provisioned EC based public key of the home network, provisioned PQC-based public key of the home network, freshly generated ECC (elliptic curve cryptography) ephemeral public/private key pair and PQC-based key encapsulation mechanism (KEM) according to the parameters provisioned by home network. The processing on UE side is done as mentioned below.

1. UE generates an ephemeral EC public key and an ephemeral EC private key at UE with Elliptical Curve (EC) key generation function.
2. UE generates a first ephemeral shared key (s1) based on the ephemeral EC private key of UE and an EC based home network public key.
3. UE generates a second ephemeral PQC shared key (s2) and an encrypted PQC shared key based on a PQC-based public key associated with the home network using ML-KEM [aa].
4. UE generates an ephemeral hybrid shared key based on the first ephemeral shared key and the second ephemeral shared key using methods like concatenation.
5. UE generates ephemeral symmetric encryption key and ephemeral MAC key using a KDF function and ephemeral hybrid shared key.
6. UE protects the plaintext block (i.e. *SUPI or* UE ID*)*, using the encryption key and the MAC key. The final output is the concatenation of the ECC ephemeral public key, the encrypted PQC shared key, the ciphertext value, the MAC tag value.

Figure 7.2.X.Y.2-1 illustrates the UE's steps.



Figure 7.2.X.Y.2-1: Encryption based on Hybrid shared key generation at UE

Finally, the proposed solution comprises transmitting the encrypted PQC shared key along with the ephemeral public key of UE, the encrypted PQC shared key, the cipher-text value, and the MAC-tag value associated with the subscriber by the UE to a network entity for authenticating the subscriber. The scheme output as defined in TS 23.003 [zz] to be updated to scheme output shown in Figure 7.2.X.Y.2-2.



Figure 7.2.X.Y.2-2: Scheme output based on Hybrid PQC-based SUPI concealment

Note: Ciphertext output from PQC key encapsulation is referred to as encrypted PQC shared key as there is another ciphertext value from step 3 of symmetric encryption, to avoid confusion.

##### 7.2.X.Y.2.2 Processing on home network side

The Hybrid shared key generation scheme is implemented such that for deconcealing a SUCI, the home network uses the received ECC ephemeral public key of the UE, encrypted PQC shared key, EC based private key of the home network and the PQC-based private key of the home network.

1. Home network (HN) generates a first ephemeral shared key (s1) based on the ephemeral EC public key, received from UE, and an EC based home network private key.
2. HN decapsulates the encrypted PQC shared key, received from UE, to derive the second ephemeral shared key (s2) using ML-KEM [aa].
3. HN generates an ephemeral hybrid shared key based on the first ephemeral shared key (s1) and the second ephemeral shared key (s2) using methods like concatenation.
4. HN generates ephemeral symmetric encryption key and ephemeral MAC key using a KDF function and ephemeral hybrid shared key.
5. HN verifies the MAC and decrypts the ciphertext to derive the plaintext block (i.e. SUPI or UE ID), using the MAC key and encryption key respectively.

Figure 7.2.X.Y.2-3 illustrates the home network's steps.



Figure 7.2.X.Y.2-3: Decryption based on Hybrid shared key generation at home network

Note: Ciphertext input to PQC key decapsulation is referred to as encrypted PQC shared key as there is another ciphertext value to step 3 of symmetric decryption, to avoid confusion.

## 7.2.X.Y.2.3 Sample Profiles for SUCI calculation

Profile C uses Post-Quantum Traditional (PQ/T) hybrid scheme as defined in RFC 9794 [ii] which is a multi-algorithm scheme where at least one component algorithm is a post-quantum algorithm and at least one is a traditional algorithm. The traditional algorithm component uses its own standardized processing for key generation (section 6 of RFC 7748 [hh]) and shared secret calculation (section 5 of RFC 7748 [hh]). The Diffie-Hellman primitive X25519 (section 5 of RFC 7748 [hh]) takes two random octet strings as input, decodes them as scalar and coordinate, performs multiplication, and encodes the result as an octet string. The shared secret output octet string from X25519 is used as the input Z in the ECIES KDF (section 3.6.1 of [gg]). The post-quantum algorithm component of PQ/T scheme uses ML-KEM as defined in [aa]. Final shared secret key Z1 is derived from combining Z and shared secret generated from ML-KEM [aa]. Use the key derivation function KDF to generate keying data K of length *enckeylen + icblen + mackeylen* octets from Z1 and [SharedInfo1]. As the point compression is not applied for profile C, the prefix rule for compression type defined in [gg] section 5.1.3 is not be used in profile C, i.e., there is no prefix for the ephemeral public key of Profile C.

Profile D uses Post-Quantum Traditional (PQ/T) hybrid scheme as defined in RFC 9794 [ii] which is a multi-algorithm scheme where at least one component algorithm is a post-quantum algorithm and at least one is a traditional algorithm. The traditional algorithm component uses point compression to save overhead and use the Elliptic Curve Cofactor Diffie-Hellman Primitive (section 3.3.2 of [gg]) to enable future addition of profiles with cofactor h ≠ 1. For curves with cofactor h = 1 the two primitives (section 3.3.1 and 3.3.2 of [gg]) are equal. The post-quantum algorithm component of PQ/T scheme uses ML-KEM as defined in [aa]. Final shared secret key Z1 is derived from combining Z and shared secret generated from ML-KEM [aa]. Use the key derivation function KDF to generate keying data K of length *enckeylen + icblen + mackeylen* octets from Z1 and [SharedInfo1].

### 7.2.X.Y.2.3.1 Profile C (Hybrid 1)

The ME and SIDF implement this profile. The parameters for this profile are the following:

- Identifier : X25519MLKEM768 (Combining X25519 ECDH with ML-KEM-768)

- EC domain parameters : Curve25519 [hh]

- EC Diffie-Hellman primitive : X25519 [hh]

- point compression : N/A

- ML-KEM parameters : Level 3 (k, lattice dimension 3)

- KDF : ANSI-X9.63-KDF [gg]

- Hash : SHA-256

- SharedInfo1 : (the ephemeral public key octet string – see [gg] section 5.1.3)

- Shared secret key Z1 : Z (see [gg] section 5.1.3) || Shared secret field from ML-KEM

- MAC : HMAC–SHA-256

- mackeylen : 32 octets (256 bits)

- maclen : 8 octets (64 bits)

- SharedInfo2 : the empty string

- ENC : AES–256 in CTR mode

- enckeylen : 32 octets (256 bits)

- icblen : 32 octets (256 bits)

### 7.2.X.Y.2.3.2 Profile D (Hybrid 2)

The ME and SIDF implement this profile. The parameters for this profile are the following:

- Identifier : SecP256r1MLKEM768 (Combining secp256r1 ECDH with ML-KEM-768)

- EC domain parameters : secp256r1 [ff]

- EC Diffie-Hellman primitive : Elliptic Curve Cofactor Diffie-Hellman Primitive [gg]

- point compression : true

- ML-KEM parameters : Level 3 (k, lattice dimension 3)

- KDF : ANSI-X9.63-KDF [gg]

- Hash : SHA-256

- SharedInfo1 : (the ephemeral public key octet string– see [gg] section 5.1.3)

- Shared secret key Z1 : Z (see [gg] section 5.1.3) || Shared secret field from ML-KEM

- MAC : HMAC–SHA-256

- mackeylen : 32 octets (256 bits)

- maclen : 8 octets (64 bits)

- SharedInfo2 : the empty string

- ENC : AES–256 in CTR mode

- enckeylen : 32 octets (256 bits)

- icblen : 32 octets (256 bits)

##### 7.2.X.Y.3 Evaluation

TBD

\* \* \* End of Changes \* \* \* \*