3GPP TR 33.841 V16.1.0 (2019-03)

Technical Report

3rd Generation Partnership Project;

Technical Specification Group Services and Systems Aspects;

Security aspects;

Study on the support of 256-bit algorithms for 5G

(Release 16)

** 

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Report is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

5G,algorithms,security,256

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2019, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword 5

Introduction 5

1 Scope 7

2 References 7

3 Definitions and abbreviations 9

3.1 Definitions 9

3.2 Abbreviations 9

4 Threats posed by quantum computing and potential countermeasures 10

4.1 Introduction 10

4.2 Threats to asymmetric cryptography 10

4.3 Threats to symmetric cryptography 11

4.4 Threats to hash algorithms 11

5 Assessment of quantum computing impact timelines 11

5.1 Predicted timescales and resources for quantum computing 11

5.2 Timelines for transitioning asymmetric algorithms 12

5.3 Timelines for transitioning symmetric algorithms 12

5.4 Considerations for assessing timelines 12

6 Impacted NextGen areas 12

6.1 Impacted use of asymmetric cryptography 12

6.1.1 Introduction 12

6.1.2 TLS for service based interfaces 12

6.1.3 NDS/IP for non-service based interfaces 13

6.1.4 SUPI protection 13

6.1.5 OAuth in SBA 13

6.1.6 N32 Application Layer Security 13

6.1.7 Network product software package integrity 13

6.1.8 EAP TLS 13

6.1.9 Ephemeral key agreement in primary authentication 13

6.2 Impacted use of symmetric cryptography 14

6.2.1 Introduction 14

6.2.2 Ciphering algorithms 14

6.2.3 Integrity algorithms 14

6.2.4 OTA mechanism 14

6.3 Impacted use of hash functions 15

6.3.1 Introduction 15

6.3.2 Key derivation function 15

6.3.3 Authentication and key agreement 15

7 Study of full entropy 256-bit keys in the 5G key hierarchy 15

7.1 Risks and mitigations for quasi-random IVs in counter mode 15

7.1.1 The attacks and their cost 15

7.1.2 Applicability to 3GPP use of counter mode 16

7.1.3 Mitigations 16

8 Assessment of the requirement for and impact of a longer MAC 17

8.1 Introduction 17

8.2 Integrity protection for control and user planes 17

8.3 MAC tag length impact on security 18

8.4 Impact of a longer MAC tag on network 18

8.5 Minimal level of integrity protection 18

9 Study of coexistence of different size keys and key size negotiation. 18

9.1 Ensuring system parameters support variable length keys 19

9.2 Ensuring system parameters support variable length MACs, AKA messages etc. 19

9.3 Ensuring key derivation functions support variable length keys 19

9.4 Using 256-bit keys in New RAN with legacy core 19

9.5 Co-existence between 128-bit and 256-bit key lengths 20

9.6 Co-existence between multiple or variable lengths MAC tags 20

9.7 Reduced complexity of key derivation function negotiation 20

10 Study of desired number of 256-bit algorithms 20

10.1 Overview of existing GSMA/3GPP symmetric algorithms 20

10.1.1 Algorithms for authentication and AKA key generation 20

10.1.2 Algorithms for encryption and integrity 21

11 Study the desired performance aspects for the new 256-bit algorithms 21

11.1 Peak data rates 21

11.2 Latency 21

12 Study of key management 22

13 Study of individual algorithm details 22

13.1 Radio interface encryption and integrity algorithms 22

13.1.1 AES 22

13.1.2 SNOW 3G 22

13.1.3 ZUC 22

13.2 AKA algorithms 22

13.2.1 MILENAGE 22

13.2.2 TUAK 22

13.3 Key derivation algorithms 23

13.3.1 HMAC-SHA-256 23

14 Potential requirements 23

14.1 Potential requirements for adoption of post-quantum algorithms 23

14.2 Potential requirements for longer MAC 23

15 Conclusions 23

Annex A: Change history 25

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# Introduction

Quantum computing poses a long-term threat to information security. This may apply to some of the current protection measures in 5G systems. These threats are studied in the present document so as to ensure that 5G systems remain secure also in the future.

The threats will impact symmetric and asymmetric cryptographic algorithms in different ways. The present document focuses on symmetric cryptographic algorithms. In particular, it focuses on the implications of introducing 256 bit cryptographic keys and algorithms.

The study allows informed decisions on why, when, where, and how symmetric cryptographic algorithms used in 3GPP systems could be strengthened to counter the identified threats.

WHY: Commercial applications (e.g., critical infrastructure, financial, medical, and pharmaceutical) and government organizations may need enhanced protection for confidential information.

WHEN: This study establishes a timeline for the introduction of enhanced protection measures. It is currently not clear when, e.g., quantum computers may pose a realistic threat. The timeline may take into account the following factors:

- the number of years that data that are sent protected in 5G (over the various interfaces) need to remain secure;

- the number of years it will take to introduce 256-bit keys in the 5G system (standardization and deployment);

- the number of years it takes to decrypt data protected with 128-bit keys, taking into account technological progress.

The study should also seek to align the security levels and timelines for introducing new asymmetric cryptographic algorithms with those for symmetric cryptographic algorithms in 5G. The reason is that the 5G system also makes use of asymmetric cryptographic algorithms, e.g. in network domain security, in untrusted non-3GPP access, and in 5G identity privacy, and it does not make sense to have different security levels for different types of algorithms in the same release of a 5G system.

WHERE: Not all parts of the 5G system may be affected in the same way. The study should therefore investigate the impacts on UE, gNB, and core network entities separately. As an example, the study may investigate whether encryption between the UE and a gNB in an operator network (where the cleartext is available to the operator in the gNB) and encryption between the UE and a core network entity in a slice are affected by the requirements in the same way.

HOW: The focus of this proposed new work will include, but will not be limited to, supporting 256-bit keys and algorithms, bolstering integrity protection by increasing the size of MAC-I in 5G networks, key derivation, AKA key generation, key distribution, key refresh, negotiation of the key size, and processing of confidential CP/UP/MP information.

# 1 Scope

The present document details the following:

- An assessment of threats and potential countermeasures posed due to quantum computing and a resulting assessment of relevant countermeasures in the 5G system.

- An assessment of the timelines for the introduction of any countermeasures, in particular the increase of key lengths to 256 bits. This includes an alignment with the timeline for strengthening asymmetric cryptographic algorithms used in 5G systems.

- An assessment of which parts of the 5G system will be affected and in which way.

- A Study of full entropy 256 bit keys in the 5G key hierarchy, beginning with the permanent pre-shared key. Including a study into modifying the derivation algorithms in order to derive child keys from the 256-bit master key instead of the 128-bit key.

- A study to determine whether a longer MAC is appropriate for 5G.

- A study of the coexistence of different size keys. In 3GPP networks, 256-bit keys in 5G will need to coexist with 128-bit keys in legacy networks or earlier 5G phases. This includes storage of keys and separate key derivation algorithms both on the UE and in the core network.

- A study into the desired number of 256-bit algorithms, e.g. if two 256-bit AKA key generation algorithm sets are needed.

- A study of the desired performance aspects for the new 256-bit algorithms taking into account software and hardware aspects.

- A study of key size negotiation.

- A study into whether the current methods for distribution and refresh of security keys are equally applicable to larger key sizes and can remain the same.

- A study into the Encryption and integrity algorithms that could be needed. This includes 256-bit session/intermediate keys in 5G, may, in some cases, simply entail using larger-key versions of current algorithms, while in other cases new algorithms may need to be chosen altogether.

- Recommendations for suitable requirements for the needed algorithms for use with 256-bit keys and ask ETSI SAGE to provide those algorithms.

# 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*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] Markus Grassl, Brandon Langenberg, Martin Roettler, Rainer Steinwandt (2015): "Applying Grover's algorithm to AES: quantum resource estimates". https://arxiv.org/pdf/1512.04965.pdf

[3] Vlad Gheorghiu, Michele Mosca (2017): "GRI quantum risk assessment report: A resource estimation framework for quantum attacks against cryptographic functions". http://globalriskinstitute.org/download/summary-report-3/

[4] Scott Fluhrer (2017): "Reassessing Grover's Algorithm". https://eprint.iacr.org/2017/811.pdf

[5] Matthew Amy, Olivia Di Matteo, Vlad Gheorghiu, Michele Mosca, Alex Parent, John Schanck (2016): "Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3". https://arxiv.org/pdf/1603.09383.pdf

[6] Daniel J. Bernstein (2009): "Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?" https://cr.yp.to/hash/collisioncost-20090517.pdf

[7] NISTIR 8105 (2016): "Report on Post-Quantum Cryptography". https://csrc.nist.gov/publications/detail/nistir/8105/final

[8] John M. Martinis (2015): "Qubit metrology for building a fault-tolerant quantum computer". https://www.nature.com/articles/npjqi20155.pdf

[9] Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin Lauter: "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms". In Proc. ASIACRYPT 2017. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/09/1706.06752.pdf

[10] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland: "Surface codes: Towards practical large-scale quantum computation". October 26, 2012. https://web.physics.ucsb.edu/~martinisgroup/papers/Fowler2012.pdf

[11] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt: "Applying Grover's algorithm to AES: quantum resource estimates".   
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/04/1512.04965-1.pdf

[12] MIT Technology Review, Practical Quantum Computers. https://www.technologyreview.com/s/603495/10-breakthrough-technologies-2017-practical-quantum-computers/

[13] Thomas Häner, Martin Roetteler, and Krysta M. Svore: "Factoring using 2n+2 qubits with Toffoli based modular multiplication". Quantum Information and Computation, 18(7&8):673–684, 2017

[14] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[15] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[16] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

[17] Void

[18] Void

[19] IETF RFC 7515: "JSON Web Signature (JWS)".

[20] Void

[21] Void

[22] Void

[23] IETF RFC 7519: "JSON Web Tokens (JWT)".

[24] 3GPP TS 33.117: "Catalogue of general security assurance requirements".

[25] NIST Special Publication 800-38B: "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication".

[26] NIST project on Post-Quantum Cryptography. <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography>.

[27] SOG-IS Crypto Evaluation Scheme: "Agreed Cryptographic Mechanisms". <https://www.sogis.org/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.1.pdf>

[28] Dustin Moody (2018): "Let's Get Ready to Rumble - The NIST PQC 'Competition'".

[29] NIST (2016): "Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process". <https://csrc.nist.gov/csrc/media/projects/post-quantum->cryptography/documents/call-for-proposals-final-dec-2016.pdf

[30] Mihir Bellare, Björn Tackmann (2017): "The Multi-User Security of Authenticated Encryption: AES-GCM in TLS 1.3".

[31] IETF Internet draft: draft-mcgrew-iv-gen: "Generation of Deterministic Initialization Vectors (IVs) and Nonces".

[32] Report ITU-R M.2410-0: "Minimum requirements related to technical performance for IMT-2020 radio interface(s)", 2017.

[33] IEEE 802.15.4-2011: "IEEE Standard for Local and metropolitan area networks--Part 15.4: Low Rate Wireless Personal Area Networks (LR-WPANs)", 2011. <https://standards.ieee.org/standard/802_15_4-2011.html>

[34] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)".

[35] IETF RFC 5689: "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)".

[36] 3GPP TS 31.115: "Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications".

[37] ETSI TS 102 225: "Smart Cards; Secured packet structure for UICC based applications".

[38] Improving the Biclique Cryptanalysis of AES, <http://link.springer.com/chapter/10.1007/978-3-319-19962-7_3>.

[39] The ZUC-256 Stream Cipher, <http://www.is.cas.cn/ztzl2016/zouchongzhi/201801/W020180416526664982687.pdf>

[40] ETSI TS 102 226: "Smart Cards; Remote APDU structure for UICC based applications".

[41] GlobalPlatform: "Card Specification, Amendment B".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**quantum computer:** a computer which makes use of quantum-mechanical effects

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

AKA Authentication and Key Agreement

CK Ciphering Key

DHE Diffie-Hellman Ephemeral

EAP Extensible Authentication Protocol

ECDHE Elliptic Curve Diffie-Hellman Ephemeral

ECIES Elliptic Curve Integrated Encryption Scheme

GKDF Generic Key Derivation Function

HKDF HMAC-based Extract-and-Expand Key Derivation Function

HN Home Network

IKEv2 Internet Key Exchange version 2

IPX IP Exchange

JSON JavaScript Object Notation

KDF Key Derivation Function

NAS Non Access Stratum

NDS Network Domain Security

OTA Over The Air

RES RESponse

RRC Radio Resource Control

SBA Service Based Architecture

SEAF Security Anchor Function

SEPP Security Edge Protection Proxy

SUCI SUbscription Concealed Identifier

SUPI SUbscription Permanent Identifier

TLS Transport Layer Security

UE User Equipment

# 4 Threats posed by quantum computing and potential countermeasures

## 4.1 Introduction

A quantum computer is a computer which makes use of quantum-mechanical effects. These effects include superposition, which allows quantum bits (qubits) to exist in a combination of several states at once, and entanglement, which allows connections between separate quantum systems such that they cannot be described independently. There exist quantum algorithms that use these effects to solve certain cryptographic problems more efficiently than they could be solved on a classical computer. However, due to the reliance on physical effects quantum computing is inherently error prone, meaning that circuits for quantum algorithms require extra qubits for error correction. This quantum error correction means that the complexity of the quantum computer required to carry out certain quantum algorithms is greater in practice than in theory. With the advances in quantum computing, the security community feels it is important to start preparing our information security systems to be resilient to this potential threat.

Determining how to cost an attack with a quantum computer is difficult. The number of qubits needs be taken into account, along with the number of quantum gates in the circuit, and the number of sequential evaluations of the gates, called the depth of the circuit. Depth governs how long the calculation will take. With a gate-time of a few nano-seconds, a depth of around 250 would take one year to evaluate.

## 4.2 Threats to asymmetric cryptography

Shor's quantum algorithm for integer factorization runs in polynomial time on a quantum computer. As the security of RSA relies on the hardness of factorizing large integers, the system's security is terminally undermined by quantum computation. A variant of Shor's algorithm enables a quantum computer to calculate discrete logarithms in polynomial time, both over finite fields and elliptic curves. This variant renders several other public-key cryptosystems insecure, including Diffie-Hellman and Elliptic Curve Diffie-Hellman.

To counter the threat of quantum computing to asymmetric cryptography it is necessary to swap existing algorithms for new, quantum-resistant algorithms. However, it should be noted that all quantum-resistant key-exchange algorithms currently being considered are much less studied than traditional public key systems such as RSA and Diffie-Hellman. As such, a balance will need to be struck between countering the quantum threat and ensuring the use of stable and tested systems.

## 4.3 Threats to symmetric cryptography

Grover's search algorithm offers a theoretical quadratic speed-up on unstructured search problems. This is applicable to symmetric key cryptography as, with use of Grover's algorithm, the N-bit key for a cipher is recovered with O(2N/2) serial quantum operations.

The real speed up offered by Grover's algorithm is difficult to evaluate and depends on a variety of factors including the scheme being analysed, the precise functionality of a quantum computer and the necessity for error-correction codes. There has been limited analysis of the effect of Grover's algorithm on 128-bit block ciphers, but various papers [2] and [3] have calculated that, while the security of AES-128 would be reduced with the development of a quantum computer, it would not fall to 64 bits. However, there has not been any concrete study quantitatively examining the impact of quantum computing on stream ciphers. It should also be noted that Grover's algorithm does not parallelise efficiently, suggesting that the security assumptions to apply in this scenario may be different to those in classical computing. It may be more appropriate to consider attacks that run in bounded time, taking into account the likely capabilities of an attacker and the amount of likely parallelisation [4].

To counter this threat to symmetric cryptography from a quantum computer it would certainly suffice to double the key-size of an algorithm, thus doubling the number of bits of classical security. As discussed above there is an ongoing discussion as to whether this response is overly conservative, as the changes would have other business, interoperability and security consequences.

## 4.4 Threats to hash algorithms

Grover's algorithm also poses a threat to the security of hash algorithms. As for symmetric algorithms, the theoretical speed-up is quadratic, but it is difficult to evaluate this in a real world scenario. [5] estimates that a single pre-image attack on SHA-256 would take ~2166 operations, rather than the theoretical 2128. Looking at another measure of cryptographic hash security, there is no known quantum algorithm which finds collisions in general hash functions more efficiently than the most efficient classical algorithms [6].

# 5 Assessment of quantum computing impact timelines

## 5.1 Predicted timescales and resources for quantum computing

It is unclear when a quantum computer that threatens cryptography will become available. However, [7] cites an estimate that a quantum computer capable of breaking 2048-bit RSA may be built by 2030 for a cost of one billion US dollars. At the First Post Quantum Cryptography (PQC) Standardization Conference in 2018, NIST [28] cited another estimate that there is a 1 in 7 chance that some fundamental public-key cryptography will be broken using a quantum computer by 2026, and a 1 in 2 chance of the same by 2031. It is likely that the cost of building a quantum computer will fall rapidly in the years following this. The efficacy of a quantum computer is inherently connected to its fault-tolerance and the requirement for quantum error correcting codes. The estimated number of physical qubits per logical qubit varies by several orders of magnitude (10 - 104) between different types of physical qubits. It is worth noting that for one type, current estimates for one logical qubit are 3600 physical qubits for quantum error correction [10]. Furthermore [8] describes improving fault tolerance in a scalable architecture as "a potential show stopper for the entire effort".

NIST's submission requirements, [29], also discuss expected circuit depth restrictions in assessing quantum attacks. Their estimates for the expected number of quantum gate operations that can be done serially in one year is 240, based on predicted developments in quantum computers.

Two research papers, [9], Table 2 and [13], have estimated the quantum resource needed to break ECC and RSA algorithms based on Shor's algorithm under certain assumptions. They estimate that for current asymmetric cryptographic algorithms 212 logical qubits are required, and 240 and 250 quantum gates with a comparable depth. This implies that commonly used asymmetric cryptographic algorithms are at risk when a quantum computer with 223 physical qubits can be built.

Grassl et al. analysed the quantum resources required to carry out an exhaustive key search for the Advanced Encryption Standard (AES) by using Grover's algorithm, [11]. The paper suggests that a similar number of logical/physical qubits will be required to attack one AES key, but the number of gates required is significantly higher with a total of 286, of depth 281, for AES-128, rising to 2151, depth 2145, for AES-256.

The report [12] states that it is conceivable that a 220 physical qubit system will be available in 10 years, though it does not give an estimate of the cost. If so, a large-scale quantum computer with sufficient qubits for some cryptographic problems could be built in 10-20 years, which is within the lifecycle of 5G systems and would compromise its asymmetric cryptography. However, [11] also notes that with their estimate of the large circuit depth required to implement Grover's algorithm, "it seems challenging to implement this algorithm on an actual physical quantum computer". This is a conclusion shared by the call for proposals for the NIST PQC standardization [29].

## 5.2 Timelines for transitioning asymmetric algorithms

In 2017 NIST launched a study [26] to evaluate and standardize one or more quantum-resistant public key cryptographic algorithms. Draft standards are expected between 2022 and 2024. Currently no quantum-resistant public key algorithms are standardized by NIST as it is assessed that not enough time has been spent analysing them. IETF is planning to introduce quantum-resistant public key algorithms in protocols (TLS, DTLS, IKEv2, X.509, JOSE, etc.) after NIST standardization

## 5.3 Timelines for transitioning symmetric algorithms

The threat to symmetric cryptography from quantum computing is lower than that for asymmetric cryptography. As such there is little benefit in transitioning symmetric algorithms without corresponding changes to the asymmetric algorithms that accompany them.

## 5.4 Considerations for assessing timelines

Some data may need to be protected for many years after transmission i.e. have a long security lifetime. Such applications require data to be protected for a given amount of time after transmission to ensure it remains protected throughout its security lifetime against anticipated threats. An attacker may be able to intercept, record and store encrypted traffic or encrypted session keys with the intention to decrypt when it becomes feasible, e.g. with the development of a cryptographically significant quantum computer. The security lifetime requirement should be taken into account when determining timelines for transitioning to quantum resistant algorithms.

# 6 Impacted NextGen areas

## 6.1 Impacted use of asymmetric cryptography

### 6.1.1 Introduction

In clause 6.1, the assumption is made that an attacker has access to a quantum computer with an implementation of Shor's algorithm for integer factorisation and/or the discrete log problem. The threats to cryptographic protocols and algorithms are considered in isolation, with no mitigating factors discussed. Certain attacks rely on the attacker having access to a network. The difficulty and likelihood of these attacks are not addressed in the present document.

### 6.1.2 TLS for service based interfaces

TLS is supported by all network functions in the 5G architecture, following the profile given in Annex E of TS 33.310 [14]. On the N32 interface it is recommended to use TLS between the SEPP and the next-hop IPX entity and where there are no IPX entities between two SEPPs TLS is used between them. All mandatory and recommended TLS cipher suites use ECDHE or DHE for key agreement. RSA may also be used as a key agreement algorithm. An attacker with access to the network interfaces where these TLS cipher suites are used may be able to collect these communications and decrypt them. This decryption would not need to take place in real time.

All TLS cipher suites which are authenticated by a digital signature algorithm also rely on asymmetric cryptography. An attacker with appropriate access could sign communications as if they were a genuine network function and may be able to inject false messages into a network. Only pre-shared key cipher suites are free from this additional risk.

### 6.1.3 NDS/IP for non-service based interfaces

Interfaces based on DIAMETER or GTP are protected using NDS/IP as specified in TS 33.210 [15]. When confidentiality protection is required, IKEv2 is used to establish a shared secret. The cryptographic algorithm for this is Diffie-Hellman. Whenever IKEv2 certificate based authentication is used an attacker with appropriate access could sign communications in the same way as described for TLS in clause 6.1.2.

### 6.1.4 SUPI protection

In 5G AKA the UE generates a SUCI using a protection scheme based on a home network public key. If the public key encryption scheme used were broken a user could be deanonymized. In the case of ECIES, as specified in Annex C of TS 33.501 [16], an attacker in possession of a HN public key could calculate the private key in advance of a connection, allowing immediate calculation of the SUPI encryption key when the UE public key is seen. In this case, the encryption scheme would offer no privacy protection for the subscriber.

### 6.1.5 OAuth in SBA

In SBA the OAuth 2.0 framework is used for Network Function authorization. The observations on TLS in clause 6.1.2 apply to the OAuth framework, as credentials and access tokens should be sent between entities under TLS.

In addition, there may be other uses of cryptography, for example if the system uses self-encoded access tokens created using JSON Web Tokens [23], which use JSON Web Signatures. If these were broken, an attacker with appropriate access could request resources from a compromised network node that that node should not be authorized to consume.

### 6.1.6 N32 Application Layer Security

Application Layer Security on the N32 interface may be used to secure the communications between a service-producing and a service-consuming network function in different PLMNs. The encryption and integrity protection keys for these communications are derived using TLS key export from an N32-c connection. An attacker with access to this connection would be able to collect the N32-c key agreement exchanges, decrypt them as described in 6.1.2 and use these to derive the encryption and integrity keys used to secure N32-f communications.

When Application Layer Security is used on the N32 interface, any IPX modifications are cryptographically signed using ECDSA as part of the JWS protocol [19]. An attacker with appropriate access could pose as an IPX and sign modifications of N32 messages.

### 6.1.7 Network product software package integrity

TS 33.117 [24] sets out the requirement that "the network product shall support software package integrity validation via cryptographic means", e.g. digital signature. An attacker who knew an authorized software source's public key could calculate the corresponding private key and use this to sign a malicious software package to be delivered to devices.

### 6.1.8 EAP TLS

Private networks using the 5G system may use EAP TLS for authentication and key agreement, as specified in Annex B of [16]. EAP TLS uses public key algorithms for key agreement. An attacker could decrypt communications sent under keys derived from the calculated pre master secret and non-confidential values such as the client and server random. With appropriate access they could also perform injection attacks as described for TLS in clause 6.1.2.

### 6.1.9 Ephemeral key agreement in primary authentication

An ephemeral Diffie-Hellman or ECDH key exchange may be added to 5G-AKA in future releases of 5G. If this is adopted, then an attacker who could collect the public keys of the UE and the network would be able to recover their shared secret. If this attacker also had access to the long-term secret of the UE they would be able to derive the session key for the session and possibly for future sessions depending on implementation.

## 6.2 Impacted use of symmetric cryptography

### 6.2.1 Introduction

In clause 6.2 the assumption is made that an attacker has access to a quantum computer with an implementation of Grover's algorithm. The threats to cryptographic protocols and algorithms are considered in isolation, with no mitigating factors discussed. Certain attacks rely on the attacker having access to a network. The difficulty and likelihood of these attacks are not addressed in the present document.

The asymmetric methods discussed in clause 6.1 are often used to establish a key for a symmetric cipher. Clause 6.2 includes cases where symmetric algorithms are used with pre-shared keys, i.e. the cases which are not broken if the asymmetric key establishment or encapsulation process falls to Shor's algorithm.

### 6.2.2 Ciphering algorithms

128-NEA1, 128-NEA2 and 128-NEA3 use 128-bit keys KUPenc, KRRCenc and KNASenc for User Plane, RRC signalling and NAS signalling encryption respectively. If these algorithms were broken by a quantum computer an attacker could recover the relevant key and decrypt any data encrypted under that stream until the key is updated.

A more resource intensive attack could recover a subscriber's long-term key. This would require an attacker to know or guess parameters related to both the home and serving network, to model the key derivation algorithm (e.g. MILENAGE) and the inputs to the KDF in the key hierarchy. It would also require a quantum circuit modelling the entire key derivation hierarchy. The complexity of this circuit is not well understood at this time. An attacker doing this would be able to decrypt all traffic belonging to that subscriber which was not encrypted at the application layer. Recovery of the subscriber's long-term key would also allow the attacker to pose as the subscriber to the network.

### 6.2.3 Integrity algorithms

128-NIA1, 128-NIA2 and 128-NIA3 use 128-bit keys KUPint, KRRCint and KNASint for User Plane, RRC signalling and NAS signalling integrity protection. If these algorithms were broken by a quantum computer an attacker may be able to recover one of these keys from the MAC-I or NAS-MAC. In the current 5G system the MAC-I and NAS-MAC are 32 bits long meaning that, with a classical attack, at least four messages would be required to recover the key.

Editor's Note: It is for further study how many protected messages would be required to recover the key with a quantum attack in this scenario.

In this scenario an attacker would be required to recover the integrity protection key quickly enough to use it before a key refresh took place. Having done this the recovered key could be used to verify the integrity of a spoofed message. If an attacker wanted to send an arbitrary message they may also need to recover the corresponding encryption key.

### 6.2.4 OTA mechanism

OTA mechanism relies on secured packet structure defined in 3GPP TS 33.115 [36], or Secure Channel Protocols (SCP '80' or SCP '81') to secure communication between a UICC and an OTA server thanks to symmetric cryptography.

**Secured packet and SCP '80'**

**Secured packet and SCP** '80' use security mechanism specified in ETSI TS 102 225 [37] and ETSI TS 102 226 [40]. The keys involved to secure communication are **KIc and KID.** The key lengths depend on the algorithms being used and are defined as being:

- 128 bits for two key 3DES

- 192 bits for three key 3DES,

- 128 bits, 192 bits or 256 bits for AES.

**The choice of secured packet or SCP '80' with AES 256-bit key enables OTA mechanism to resist to an attacker with a quantum computer.**

**SCP '81'**

Secure Channel Protocol '81' is specified in Amendment B of Global Platform Card Specification [41], which is based on PSK TLS and defines PSK TLS key set consisting of two kinds of keys: a PSK TLS key and a DEK (decryption/encryption) key. The list of cipher suites for SCP '81' is the following:

- TLS\_PSK\_WITH\_3DES\_EDE\_CBC\_SHA

- TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA

- TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256

- TLS\_PSK\_WITH\_NULL\_SHA

- TLS\_PSK\_WITH\_NULL\_SHA256

If these algorithms were broken by a quantum computer an attacker could recover the relevant key and possibly decrypt/modify data sent over the PSK TLS-based secure channel.

The DEK key may be used to decrypt or encrypt sensitive data in addition to PSK TLS. Consequently, in scenarios using the DEK key to protect sensitive data, the DEK key could protect against an attacker with a quantum computer.

## 6.3 Impacted use of hash functions

### 6.3.1 Introduction

In 6.3 the same assumptions are made as in clause 6.2.

### 6.3.2 Key derivation function

There are two kinds of key derivation functions deployed in the 5G system. One is generic key derivation function (GKDF) defined in TS 33.220 [34], which is used for all key derivations in 5GC as shown in Figure 6.2.2-1 and figure 6.2.2-2 in TS 33.501[16]. The other one is the HMAC-based Extract-and-Expand Key Derivation Function (HKDF) specified in RFC 5869 [35], which has been applied to IKEv2, TLS 1.2, TLS 1.3, N32-f, and EAP-AKA' for the key derivation.

The base of GKDF and HKDF is the keyed hash function HMAC-SHA-256.

### 6.3.3 Authentication and key agreement

During 5G AKA, RES\* is sent from the UE to the SEAF in the serving network. An attacker could recover the subscriber's long term key from RES\*, using a quantum circuit modelling the key derivation hierarchy. A similar attack could also be applied to RES in EAP-AKA'. This would require an attacker to know or guess parameters related to both the home and serving network, to model the key derivation algorithm (e.g. MILENAGE) and the inputs to the KDF in the key hierarchy. The complexity of the quantum circuit required for this attack is not well understood at this time.

# 7 Study of full entropy 256-bit keys in the 5G key hierarchy

## 7.1 Risks and mitigations for quasi-random IVs in counter mode

### 7.1.1 The attacks and their cost

When a block cipher is used in counter mode, reuse of the same counter block can in some cases compromise the confidentiality of the system. Three potential attacks are identified.

**Attack 1:** If the same counter block is used twice with the same key, then there is an immediate and serious loss of security. This means that two different plaintext segments are encrypted by XORing with the same keystream block, and hence the attacker can recover the XOR of the two plaintext segments. Depending on the plaintext, this may be enough to recover both plaintext segments in their entirety.

**Attack 2:** A computationally efficient attack is also possible if the same **predictable** counter block is likely to be used by many different users with different keys. (A simple example of this would be if the first counter block used with a newly computed key is always the zero block.) It may then be worthwhile for the attacker to precompute a large rainbow table, based on the mapping from a variable key and that fixed counter value to a variable keystream block. Whenever a known plaintext block is encrypted with that fixed counter value, the attacker can then use the rainbow table to look up the keystream block and (if successful) recover the key. The probability of success on each occasion depends on the number of keys covered by the rainbow table. Building the table may be computationally demanding, but only has to be done once; the time taken for an individual attack instance can then be much lower. (Note: rainbow tables are not the only possible construction for these look-up tables, but they are typically the most efficient.)

**Attack 3:** Finally, a multi-target attack is possible if known plaintext is encrypted with the same counter block by many different keys, and if the attacker can collect 2k of these cipher blocks, and thus 2k blocks of keystream, then the attacker would expect to perform 2n-k trial encryptions of the counter block to find one match with high probability. Note that the blocks may come from different users, and there is no control over which key is recovered.

Attack 1 does not involve any kind of key search, and has very low computational complexity. For attacks 2 and 3, it is not clear that there is any impact of quantum computing through Grover's algorithm which is usually phrased in terms of finding as single input to a function that gives one specific output. However, classical computing may reach zetta-scale power by 2030 if Moore's law continues to hold. That means, it is plausible to expect that the most powerful supercomputer in 2030-40, will be able to perform 270 FLOPS. A cryptographically significant computation is not equivalent to one floating-point operation (say on a Xeon), so this may equate to fewer cryptographic operations per second per super computer, perhaps around 264. Thus, for instance, Attack 3 with n=128 and k=32 may take hundreds, if not thousands, of years. For reference, over the previous 20 years, the leading super-computers of the day cost between 100 and 400 million dollars.

### 7.1.2 Applicability to 3GPP use of counter mode

3GPP NEA algorithms do not use a random field as an IV. The counter block is constructed as follows:

- COUNT, 32 bits, increasing per-message sent under the same key to avoid repeated keystreams, different COUNTs are used for different security associations, initialised to 0.

- BEARER, 5 bits

- DIRECTION, 1 bit

- 0-padding, 26 bits

- COUNTER, 64-bit counter incremented per cipher block, initialised to 0

The use of COUNT protects effectively against Attack 1.

For Attacks 2 or 3, an attacker would need to collect all messages with the same COUNT, BEARER, DIRECTION and COUNTER for the attack, as well as to reliably know the plain text.

Editor's Note: it is FFS to what extent and where (NAS, RRC, UP) predictable plain texts will be repeatedly encrypted with any of the different keys and the same counter value in 3GPP systems.

Either of Attacks 2 or 3 recovers one of the session confidentiality keys.

### 7.1.3 Mitigations

The counter block construction described in the previous clause means that counter blocks will indeed be quite predictable and repetitive. This makes both of Attacks 2 and 3 potentially applicable in theory. In practice, a key length of 128 bits means that both attacks are still very computationally intensive, but they are nevertheless more efficient than a straightforward exhaustive key search.

If the risks are deemed sufficient, a simple mitigation for this threat is to include some random data in part of the counter block to ensure that the counter block is sufficiently unlikely to repeat across large numbers of different users. Alternatively, the key size could be made longer to render the attack impractical, though the attack still implies that the security is affected, and thus to maximise the security of the system, it could be advisable to randomise the IV regardless of the key size.

In the simple solution case, for NEA1/2/3 the 26 bits of 0-padding could be replaced with a random value. For more security, some of the 64-bit counter could be given over to a random field. The random value need only change when the encryption key changes. It needs to be established if this is sufficient, how many bits should be randomised, and if it introduces any further issues.

Sharing random bits consumes bandwidth; if it is preferable to avoid this, it may instead be possible to use some bits that would be known to both parties, but that would vary between instances. Some bits from a temporary subscriber identifier might be suitable, for instance – but the options would depend on the particular link being encrypted.

Many recent security protocols such as TLS 1.3, DTLS 1.3, SRTP, Encrypted Content-Encoding for HTTP, and OSCORE use the unpredictable IV/Nonce Format described in Section 4.4 of [31]. For the 3GPP radio algorithms, a 128-bit randomizer could be used, which would be exclusive-ored (XOR) with the same counter block (COUNT, BEARER, DIRECTION, 0-padding, COUNTER) as the current 3GPP NEA algorithms. This simple solution allows a maximum number of bits to be randomized. In addition, the randomizer can be derived from the same shared secret as the key and kept secret in a similar way as (D)TLS 1.3 and OSCORE, which further strengthens the security, see [30].

# 8 Assessment of the requirement for and impact of a longer MAC

Editor's Note: This section will contain the study on whether a longer MAC is appropriate for 5G. Note that the higher data rates achievable in 5G should be able to accommodate a reasonable MAC-I size increase without suffering significant performance degradation. It is also to be studied whether an integrity algorithm different from the ones standardized for 5G phase 1 needs to be developed.

## 8.1 Introduction

Within 3GPP networks, when a use case requires integrity protection for user plane traffic using a MAC longer than 32 bits, it can currently be achieved using over-the-top application layer integrity protection mechanisms. These require no changes to the 3GPP network. Currently 3GPP networks do not provide native support for a MAC longer than 32 bits, and this support would require changes to the network and protocols.

Editor's Note: It is FFS whether there are security risks that compromise the integrity of the applications if a longer MAC is not supported by 3GPP networks natively and if so, what they are. It is also FFS how the use of longer MAC can be negotiated securely given that there might be legacy networks that do not support longer MAC.

All (non-null) 5G integrity algorithms use a 128-bit key and result in a 32-bit MAC. If it is deemed that quantum computing requires modification of the confidentiality algorithms to use 256-bit keys, then the integrity algorithms may need to change accordingly. This change is independent of the security requirements on MAC length.

## 8.2 Integrity protection for control and user planes

Currently, 3GPP requires integrity protection only for Control Plane traffic for UMTS, EPS, and Rel-15 of 5GS. Integrity protection of User Plane traffic in Rel-15 of 5GS is optional to use.

It is arguably more expensive to implement larger MAC size for User Plane integrity protection than for Control Plane integrity. However, one can envision a line of services that require levels of OTA integrity protection that are higher than established 32-bit MAC can provide.

## 8.3 MAC tag length impact on security

The MAC-I is fundamental for ensuring that messages sent within the 3GPP system have cryptographic integrity protection ensuring they cannot be forged or modified. Currently, 5G specifies the use of MAC algorithms with 128-bit key and a 32-bit MAC tag length. In the case of NIA1 and NIA2, the 32-bit tag length is obtained by truncating the output of the MAC algorithm; NIA1 natively produces a 64-bit tag, while NIA2 produces a 128-bit tag.

Truncating MAC tags is a common practice, provided that the MAC key length is sufficient to meet the desired security strength of the scheme. However, the MAC tag length does have an impact on security, as it indicates the likelihood that an adversary with no knowledge of the MAC key can present a message and tag that would pass verification. That is, with a 32-bit MAC tag length, a trivial forgery attack would allow an attacker to forge a message after 232/2 attempts on average. Short MAC tags could create an unacceptable security risk in systems that allow an attacker to attempt a large number of messages that would be verified by a given MAC key, depending on the system's tolerance for accepting a forged message. Some operator services (e.g., Ultra-Reliable Communications, Critical Communications, Government Communication) may require levels of integrity protection that are beyond the level achieved by a 32-bit MAC.

While guidance from NIST allows MAC tags as short as 32-bits, it recommends tag lengths of at least 64-bits [25] to reduce the likelihood of accepting forged data. Use of shorter MAC tags may be appropriate in certain constrained use cases where the system is able to limit the number of messages failing verification under a given key. Protocols with high data throughput and long-lived keys should use a MAC tag of at least 64 bits. This guidance is not related to quantum computing. SOG-IS have also made recommendations on MAC length [27]. SOG-IS recommend at least 96 bits, with 64 bits reserved for legacy applications.

Editor's Note: It is for future study and analysis the application of this attack to the specific 3GPP case, i.e. what an attacker can achieve in practice with an attempted forgery attack on 3GPP integrity protection.

Note: There are other attacks against integrity algorithms discussed in [25], including replay attacks and collision attacks, but vulnerability to these attacks is affected by factors other than MAC tag length. While on average 232/2 messages are required to find a collision between two MACs (for a MAC tag length of 32-bits), collision attacks typically rely on finding two MACs which have the same value before truncation. As such, the block size of the algorithm is more important than the MAC tag length for this attack. Replay attacks are prevented by adding COUNT, BEARER and DIRECTION values to a message before calculating the MAC, thus uniquely identifying that session. Therefore, integrity protected communications are not vulnerable to replay attacks, and this is not affected by MAC tag length.

## 8.4 Impact of a longer MAC tag on network

There is no cryptographic overhead in increasing the MAC tag length in the 3GPP integrity algorithms from 32 to 64 bits since the untruncated tag lengths for NIA1 and NIA2 are both at least 64 bits.

Editor's Note: The impact of a longer MAC on the RAN, UE and existing hardware is FFS

## 8.5 Minimal level of integrity protection

Thecurrently adopted 32-bit size of MAC-I and the corresponding level of protection should be minimally accepted when integrity protection is applied in 5GS Rel-16 and beyond.

Editor's Note: It is for future study the appropriate length of MAC-I for messages with different risk levels

# 9 Study of coexistence of different size keys and key size negotiation.

Editor's note: This section needs to be reviewed once content in the clauses above has been added.

Editor's Note: This section will contain study on key size negotiation: The security specification should be flexible so as to be easily adapted or upgraded in the future, particularly taking into consideration roaming situations.

Editor's Note: This section will also contain study on coexistence of different size keys: In 3GPP networks, 256-bit keys in 5G will need to coexist with 128-bit keys in legacy networks or earlier 5G phases. This entails storage of keys and separate key derivation algorithms both on the UE and in the core network.

## 9.1 Ensuring system parameters support variable length keys

If the 5G system supports 128-bit as well as 256-bit keys, it cannot be assumed that any such key is of a fixed length. Accordingly, all messages, fields and other parameters communicating cipher keys and integrity keys in the 5G system should allow for the fact that the associated key has variable length.

The entropy of other keys in the key hierarchy (KSEAF, KAMF, KgNB etc.) is decided by the entropy of long term key. As a general principle, it would be desirable to preserve as much entropy as possible through the key hierarchy. Even if independent 256-bit ciphering and integrity keys derived from 256-bit K is established between the UE and any node, entropy of the keys is 256 bits. Accordingly, messages, fields and other parameters communicating other keys in the hierarchy should allow for the fact that the associated key may have variable length.

Editor's Note: Use of concept of entropy key to describe security parameters is FFS.

The 5GC and 5GRAN are already required to support the transport of 256-bit symmetric keys as per requirement 5.1.3 in [16].

Any public keys in the 5G system may have to be of variable size to provide equivalent security to the different symmetric key lengths. Proposed quantum safe public key algorithms sometimes have very large key sizes (running into many thousands or millions of bits). Accordingly, messages, fields and other parameters communicating public keys should potentially allow for very long message sizes: up to millions of octets.

To avoid making multiple changes in future specifications, it is proposed not to set upper bounds on key lengths in any key e.g. to use LV or TLV constructions rather than zero-padding keys up to a larger fixed length.

## 9.2 Ensuring system parameters support variable length MACs, AKA messages etc.

Using a 256-bit key with – for example – a 32-bit MAC or a 64-bit AUTN or RES looks unusual going forwards. While standardizing increased key lengths, 3GPP should take the opportunity to revise the maximum lengths of other input and output parameters, in order to facilitate a consistent security level.

Accordingly, all messages, fields and other parameters communicating inputs and outputs to AKA, or output from any integrity algorithm in the 5G system, should allow for variable lengths. Again, to avoid making multiple changes in future specifications, it is proposed not to set upper bounds on lengths in any of these constructions e.g. to use LV or TLV constructions.

## 9.3 Ensuring key derivation functions support variable length keys

All key derivation functions should allow for variable length output. In some cases, a derivation function will generate 128-bit keys; in others it will generate 256-bit keys. As identified in point 9.1, intermediate keys in the hierarchy may have variable length as well. To ensure cryptographic separation, the desired output key length or lengths (if several keys are being derived at once) should be provided as an input parameter to the KDF.

## 9.4 Using 256-bit keys in New RAN with legacy core

A special concern arises in cases where a new 5G RAN is combined with a legacy LTE core. The gNB could attempt to negotiate 256-bit keys with the UE in such cases, but the core would not be aware of this.

Editor's Note: It needs to be considered if this approach is workable, or introduces more security problems than it potentially solves.

## 9.5 Co-existence between 128-bit and 256-bit key lengths

If future releases support 256-bit key size, a secure negotiation protocol has to be established to ensure mitigation against bidding down attacks.

Editor's Note: This requires further investigation on how this can be achieved and on the key entropy.

## 9.6 Co-existence between multiple or variable lengths MAC tags

Rel-15 5GS supports 32-bit MAC-I. If future releases support larger MAC-I size(s), a secure negotiation protocol has to be established to ensure mitigation against bidding down attacks.

Editors Note: Granularity of application for larger MAC-I is FFS.

Editors Note: This requires further investigation on how this can be achieved.

## 9.7 Reduced complexity of key derivation function negotiation

Support for both variable length keys and variable KDF algorithms would benefit from addressing negotiation of both the size of the keys to derive and the KDF used to derive the keys in the same procedures. Many existing security mechanisms already support KDF negotiation.

KDF negotiation in the 5G system becomes quite complex because there are many Network Entities that need to negotiate KDFs with the UE, e.g., gNB, AMF, SEAF, AUSF, UDM and potentially introduce additional overhead in the signalling.

If KDF negotiation is supported in the 5G system design consideration should be taken, for example, to reuse the existing signalling, combine two negotiations, etc.

# 10 Study of desired number of 256-bit algorithms

## 10.1 Overview of existing GSMA/3GPP symmetric algorithms

This clause lists and analyses the existing symmetric GSMA/3GPP algorithms for authentication, AKA key generation, encryption and integrity.

### 10.1.1 Algorithms for authentication and AKA key generation

Figure 10.1.1-1lists the GSMA and 3GPP algorithms for authentication and AKA key generation. Legacy algorithms with keys smaller than 128 bits are no longer recommended and should be phased out due to their short key length.

Table 10.1.1-1: Algorithms for authentication and AKA key generation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cipher | Proprietary | Proprietary | Proprietary | AES | Keccak |
| Input Key Size | 128 | 128 | 128 | 128 | 128, 256 |
| Output Key Size | 54 | 54 | 64 | 128 | 128, 256 |
| RES Size | 32 | 32 | 32 | 64 | 32,64,128,256 |
| Name | COMP-128-1 | COMP-128-2 | COMP-128-3 | MILENAGE | TUAK |

3GPP systems has normally specified two cryptographically strong algorithms for each functionality. Together with algorithm negotiation, this means that the 3GPP systems are strong even if a weakness is found in one of the two algorithms. For AKA key generation with 256-bit keys, as well as RES sizes larger than 64 bits, only one algorithm is currently specified (TUAK based on the Keccak sponge function also used in SHA-3).

To follow the principle of having two parallel algorithms (which has served cellular systems well), GSMA/3GPP need to standardize a second AKA key generation algorithm for 256-bit keys and longer RES. One possible option being MILENAGE extended with AES-256 in addition to AES-128.

### 10.1.2 Algorithms for encryption and integrity

Figure 10.1.2-1 lists the GSMA and 3GPP algorithms for encryption and integrity. Legacy algorithms with keys smaller than 128 bits are no longer recommended and should be phased out due to their short key length.

Table 10.1.2-1: Algorithms for encryption and integrity

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cipher | Proprietary | Proprietary | KASUMI | KASUMI | KASUMI | SNOW 3G | SNOW 3G | AES | AES | ZUC | ZUC |
| Key Size | 64 | 64 | 64 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| Mode | XOR | XOR | f8-mode | f8-mode | CBC-MAC | XOR | CW-MAC1 | CTR | CMAC | XOR | CW-MAC2 |
| Type | ENC | ENC | ENC | ENC | INT | ENC | INT | ENC | INT | ENC | INT |
| MAC Size | - | - | - | - | 32 | - | 32 | - | 32 | - | 32 |
| GSM | A5/1 | A5/2 | A5/3 | A5/4 |  |  |  |  |  |  |  |
| GPRS | GEA1 | GEA2 | GEA3 | GEA4 | GIA4 | GEA5 | GIA5 |  |  |  |  |
| UMTS |  |  |  | UEA1 | UIA1 | UEA2 | UIA2 |  |  |  |  |
| LTE |  |  |  |  |  | 128-EEA1 | 128-EIA1 | 128-EEA2 | 128-EIA2 | 128-EEA3 | 128-EIA3 |
| NR |  |  |  |  |  | 128-NEA1 | 128-NIA1 | 128-NEA2 | 128-NIA2 | 128-NEA3 | 128-NIA3 |

All existing integrity algorithms uses a MAC length of 32 bits. In addition to the algorithms listed in Figure 10.1.2-1, the terms A5/0, GEA0, UEA0, UIA0, EEA0, EIA0, NEA0, NIA0 are used as different names for the NULL algorithm.

# 11 Study the desired performance aspects for the new 256-bit algorithms

## 11.1 Peak data rates

An essential requirement for the new 256-bit algorithms is the ability to achieve the peak data rates of the radio access network they are protecting. If not, the ciphering algorithms may become a latency or throughput bottleneck. The minimum requirement for downlink peak data rates in 5G/IMT-2020 is 20 Gbps [33]. The 256-bit algorithms should be able to achieve such peak rates both when implemented in hardware or software on commodity CPUs. However, all previous 128-bit algorithms standardized for 3G and 4G have been used also in later generations. The 256-bit algorithms should therefore not only be able to achieve the peak data rates of 20 Gbps in 5G/IMT-2020, but preferably also the peak data rates of future generations of mobile networks.

## 11.2 Latency

One of the requirements of 5G/IMT-2020 [32] is ultra-low latency communication with only 1ms end-to-end latency. To achieve this, it is important that the 256-bit algorithms have as low latency as possible. As the traffic is typically encrypted and decrypted several times, the latency of the 256-bit algorithms will be added several times to the end-to-end latency.

# 12 Study of key management

Editor's Note: This section will study key management, key distribution and key refresh. It should include whether the current methods for distribution and refresh of security keys are equally applicable to larger key sizes and can remain the same.

# 13 Study of individual algorithm details

## 13.1 Radio interface encryption and integrity algorithms

### 13.1.1 AES

It would be natural to adapt the encryption and integrity algorithms based on 128-bit AES to use 256-bit AES instead. There are published attacks that are notionally very slightly better than exhaustive search on 256-bit AES [38], but in practice it remains a completely standard choice of strong algorithm.

The AES-modes used in Rel-15 of the 5G system are 128-bit AES in CTR mode and 128-bit AES in CMAC mode. These modes were introduced in 3GPP in Rel-8 for LTE. If 256-bit AES is to be introduced, newer AES-modes, e.g. GCM, could be taken into consideration in a possible normative phase for possible performance improvements.

### 13.1.2 SNOW 3G

The internal state of the SNOW 3G keystream generator is more than 512 bits. Furthermore, SNOW 3G is adapted (strengthened) from SNOW 2.0, which was designed in the first place to support keys up to 256 bits. While more study is needed, it seems plausible that SNOW 3G (and the encryption and integrity algorithms based on it) may be able to support a 256-bit key without significant change.

### 13.1.3 ZUC

ZUC-256 [39], the successor to ZUC, supports 256-bit key length. The ZUC-256 is specifically designed for the 5G System that will support 256-bit algorithms. The internal state of ZUC is 560 bits. ZUC-256 uses the same key stream generation function as ZUC, so when ZUC-256 is implemented in hardware, it can reuse the circuit designed for ZUC. The MAC tag generation of ZUC/ZUC-256 is different from AES and SNOW. ZUC-256 supports 32/64/128-bit MAC tag. It is recommended that the evaluation of ZUC-256 should be started as soon as possible to determine if it can meet the security requirements of 5G systems.

## 13.2 AKA algorithms

### 13.2.1 MILENAGE

MILENAGE is built around AES. The same observations apply as in clause 13.1.1. It should be reasonably straightforward to create a 256-bit version of MILENAGE (although some slight construction changes would be needed to produce 256-bit output values).

### 13.2.2 TUAK

TUAK is already designed to accommodate 256-bit keys, and produce 256-bit outputs, if required.

## 13.3 Key derivation algorithms

### 13.3.1 HMAC-SHA-256

The KDF used for many purposes in 3GPP is based on HMAC-SHA-256. Using HMAC-SHA-256 to derive (up to) 256-bit keys from 256-bit keys is in line with standard advice.

# 14 Potential requirements

## 14.1 Potential requirements for adoption of post-quantum algorithms

There is an established long term threat to the asymmetric algorithms used in 5G. As there are currently no approved quantum safe replacements for existing asymmetric algorithms, methods which rely on these algorithms should be designed so that quantum safe mechanisms can be introduced later in a way that is protected from bid down attacks.

Where non-3GPP protocols are used in the 5G system, software implementations should be kept up to date to ensure quantum safe algorithms can be adopted once they are available. The 3GPP profiles for these protocols in [15] should also be updated to include support for quantum safe algorithms.

3GPP methods that use Diffie-Hellman or other asymmetric algorithms for key agreement, should be extensible to include new quantum safe key agreement mechanisms and tolerant of longer messages to deliver these key agreements. The SUPI privacy mechanism available in 5G is already extensible and when quantum safe key encapsulation mechanisms are approved an appropriate one should be chosen and added as a default SUPI encryption profile.

It should be assumed that new applications of asymmetric algorithms to the 3GPP system will be required to transition to post-quantum alternatives in the lifetime of the system. Therefore, the impact of longer key sizes and cipher text sizes should be assessed when designing protocols using asymmetric cryptography.

Once a cryptographically relevant quantum computer is believed to exist it should be possible for network products to validate software updates using quantum safe signature algorithms.

Evaluation of 256-bit ciphering and integrity algorithms should begin, to ensure that such algorithms are available if they are required for the 5G system.

A consistent level of security should be maintained in the 5GC and 5GRAN, taking into account encryption, physical security and respective threat models.

## 14.2 Potential requirements for longer MAC

Some 5G operator services may potentially require greater assurance of integrity than that provided by a 32-bit MAC. A bid-down protected mechanism to support a 64-bit MAC for integrity protection of the control and user plane in 5GS Rel-16 could be considered.

# 15 Conclusions

While timelines for quantum computing are difficult to estimate, it is possible that a cryptographically relevant quantum computer will be developed within the lifetime in the 5G system. Any such computer is likely to terminally undermine the security of all asymmetric algorithms in common use today. The effect of a quantum computer on symmetric algorithms is more difficult to assess, but the existence of a quantum computer will change the applicable security model.

The majority of asymmetric algorithms in use in the 5G system are used as part of a non-3GPP protocol. For example, IPsec and TLS are used widely, while JWS is used in N32 Application Layer Security. The present document recommends that the 3GPP profiles for these protocols, defined in [15], are updated to include support for quantum-safe algorithms once these are available.

All mandatory non-null SUPI encryption profiles rely on the security of elliptic curve cryptography, which is broken by Shor's algorithm. Therefore, to protect SUPI from an attacker with a quantum computer a new profile will be required. Currently there are no recommended quantum safe algorithms to replace existing asymmetric key agreement and key encapsulation mechanisms. Recommendations from NIST are expected between 2022 and 2024. Despite this, the 5G system needs to be prepared to transition to quantum safe algorithms. It is not yet clear what classes of algorithm will be favoured by NIST and as such it is difficult to predict what key sizes and ciphertext sizes the system will be required to support. However, these could be hundreds of thousands or even millions of bytes. Therefore, it is recommended that wherever the SUCI is included in a message, the field is suitably sized.

Where new security features are introduced into the 5G system, it can be assumed that any asymmetric cryptography will be required to transition to quantum safe algorithms at some point in the future. These security features should be assessed to ensure that the large key and cipher text sizes do not introduce performance issues and should be designed to be extensible to new algorithms in a bid down protected manner.

It should be noted that in many cases it is already possible to apply 256 bits of classical security to 5G User Plane traffic. In particular, 256-bit block ciphers can be used in IPsec and HTTPS traffic that is carried over the RAN. In addition, the 5GC and 5GRAN are already required to support the transport of 256-bit keys and a 256-bit long term key can be used as input to TUAK.

Since the practical impact of Grover's algorithm is not yet well understood the post-quantum security of symmetric algorithms is difficult to evaluate at this time. However, based on current estimates for the requirements to implement Grover's algorithm, the present document has concluded that there is no immediate need to transition to 256-bit key lengths.

Despite the above conclusion, the 5G system may be required to support 256-bit algorithms in future releases. As the evaluation of new algorithms takes time, it is proposed that this work should start now.

Annex A:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2017-11 | SA3#89 | S3-173462 | - | - | - | S3-173247, S3-173248, S3-3281, S3-173478 and S3-173385 as agreed at SA3#89 | 0.1.0 |
| 2018-03 | SA3#90bis | S3-180936 | - | - | - | Added S3-180937, S3-180938, S3-180171, S3-180939 and S3-180718 as agreed at SA3#90bis | 0.2.0 |
| 2018-04 | SA3#91 | S3-181559 | - | - | - | Added S3-181558, S3-181459, S3-181560, S3-181561, S3-181563 and S3-181278 | 0.3.0 |
| 2018-05 | SA3#91bis | S3-182074 | - | - | - | Added S3-182073 | 0.4.0 |
| 2018-08 | SA3#92 | S3-182676 | - | - | - | Added S3-182675 | 0.5.0 |
| 2018-09 | SA3#92bis | S3-183169 | - | - | - | Added S3-182980, S3-183165, S3-182961, S3-182978, S3-183014, S3-183166 and S3-183167 | 0.6.0 |
| 2018-11 | SA3#93 | S3-183758 | - | - | - | Added S3‑183757, S3‑183759, S3‑183760, S3‑183761, S3‑183762,  S3‑183763, S3‑183764, S3-183767, S3‑183768 and S3‑183765 | 0.7.0 |
| 2018-12 | SA#82 | SP-181034 |  |  |  | EditHelp/MCC revision. Presented for approval | 1.0.0 |
| 2018-12 | SA#82 |  |  |  |  | Upgrade to change control version | 16.0.0 |
| 2018-09 | SA#83 | SP-190098 | 0001 | - | D | Editorial changes recommended by Edithelp | 16.1.0 |