SCIENCE CHINA
Information Sciences

December 2020, Vol. 63 220301:1–220301:22
https://doi.org/10.1007/s11432-019-2907-4

• REVIEW •
Special Focus on Challenges and New Insights for Network Security in 5G-and-Beyond

An overview of cryptographic primitives for possible use in 5G and beyond

Jing YANG* & Thomas JOHANSSON

Department of Electrical and Information Technology, Lund University, Lund 22100*, Sweden*

Received 9 December 2019/Revised 20 February 2020/Accepted 11 May 2020/Published online 11 November 2020

Abstract This survey overviews the potential use of cryptographic primitives in the fifth-generation mobile communications system (aka 5G) and beyond. It discusses the new security challenges that come with 5G and presents the upcoming security architecture. It shows the use of current cryptographic algorithms and discusses new algorithms or modifications of existing ones, that can be relevant. It also discusses the need for lightweight algorithms to meet the new use cases as well as the general demand for algorithms secure even when large quantum computers are available.

Keywords 5G, cryptographic primitives, lightweight cryptography, post-quantum cryptography

Citation Yang J, Johansson T. An overview of cryptographic primitives for possible use in 5G and beyond. Sci China Inf Sci, 2020, 63(12): 220301, https://doi.org/10.1007/s11432-019-2907-4

1 Introduction

Wireless communication has transformed our society. The introduction of mobile communication as specified and provided by the second, third and fourth generation of mobile communication systems (respectively referred to as 2G, 3G, long-term evolution (LTE)) has changed our lives. Most people today own a mobile phone, keeping it closeby, and conducting not only phone calls, Internet browsing, but also a number of other very convenient services, which may include mobile tickets, money transfer, paying for parking, etc.

Today, researchers and developers are focusing on the next-generation of mobile communication, the so-called 5G system. This next-generation mobile network is currently being specified by the third generation partnership project (3GPP). Mobile carriers have started building 5G networks in some cities and there are already a few mobile phones offering 5G connectivity. The 5G system will be the enabling platform for achieving the original goals set up in IMT-2020 [1], which contains a number of use cases spanning from voice and fast connectivity to new use cases, such as smart city, smart home, industry automation, self-driving vehicles and mission-critical applications.

Based on an International Telecommunication Union (ITU) report "Setting the scene for 5G: opportunities and challenges", 3GPP has identified three typical categories of use cases, specified as enhanced mobile broadband (eMBB), critical communications, and the massive Internet of Things (mIoT). Below is a general description of 5G in the aspects of some general features and requirements for these use cases [2].

• **eMBB.** 5G should meet a performance goal of very high data rates and still low latencies. Requirements include a cell downlink throughput of 20 Gbps (gigabits per second) and a cell uplink of 10

* Corresponding author (email: jing.yang@eit.lth.se)

Gbps. In the data plane (the part of a network that carries user traffic and processes the data requests, also known as the user plane), the latency should be bounded by 4 ms.

• **Critical communications.** 5G should meet a performance goal of ultra-high reliability, a very low latency and also very strong security. For critical communication there is a requirement of a data plane latency of 1 ms and a high reliability of 99.999%.

• **mIoT.** 5G should manage a performance goal of an ultra-high device density and ultra-low energy consumption. In particular, a million devices per square kilometer should be able to be connected.

Besides these requirements, there are several additional performance goals in terms of lowering deployment costs and providing increased availability for different mobility profiles (e.g., users traveling at different speeds). The vision of 5G is not to be achieved through a single 5G network; instead, it will connect and interact with other systems. This will require a flexible system operating across multiple network boundaries.

The 5G network will make use of virtualization and other enabling techniques to achieve the goals of scalability and flexibility [3,4] to support the multiple use cases. In particular, some critical technologies are: network slicing, mobile edge computing (MEC), software-defined networking (SDN), and network function virtualization (NFV).

• **Network slicing.** Logical networks are dedicated to isolated applications. In this way, multiple virtual networks can independently operate on a single physical infrastructure. This affects several of the layers in the network, from the radio interface to the routing and forwarding.

• **MEC.** Computing and storage functions are brought closer to the edge of the network, thereby reducing both latency and the amount of data handled in the core network. This will bring performance benefits and the possibility of new services.

• **SDN.** SDN is a technique to separate the data and control planes (the part of a network that carries signaling traffic, configures and shuts down the data plane), using centralized control. This allows reprogramming switches and routers in a network to meet varying demands.

• **NFV.** NFV is the use of multiple network configurations, with network functions such as IP address allocation, network scaling and firewall settings being virtualized and controlled by demands on the network.

These technologies must interact and eventually be fully integrated in 5G systems, which, however, brings new challenges. One of the main challenges for this integration is the system security solution. It is clear that in this broader picture, where 5G mobile networks are interacting with other networks and where devices have very different demands on security, the 5G system security is a much more complicated and challenging task compared to legacy systems. This may require enhancements from several aspects: some improvements and changes in the architecture and protocols are needed; security should be guaranteed at a higher level, increasing it to 256-bit for some algorithms, owing to the threat of quantum computers; the use of public key cryptography may have to rely on new algorithms based on problems different from the factoring and discrete logarithm problems; lightweight cryptographic algorithms and protocols will have to be analyzed and adopted.

We give an overview of the cryptographic primitives which are currently being used or could potentially be used in 5G and beyond. The rest of this review is organized as follows. We first give the security architecture and mechanisms of 5G and show how and where cryptographic primitives work for this security architecture in Section 2. We then list several potential cryptographic primitives which might be used in 5G for confidentiality and integrity protection in Section 3. After that, we in Section 4 give a review of the lightweight cryptography which would be expected for IoT devices in 5G and present some lightweight cryptographic algorithms and protocols in the post-quantum scenario in Section 5. We lastly conclude the paper in Section 6.

Some frequently used acronyms and references are respectively listed in Tables 1 and 2 [4–28].

Table 1 Summary of main acronyms

Acronym	Definition	Acronym	Definition
3GPP	Third generation partnership project	5G	Fifth generation wireless network
AES-NI	Intel advanced encryption standard new instructions	AKA	Authentication and key agreement
AMF	Access and mobility management function	ARPF	Authentication credential repository and processing function
AUSF	Authentication server function	BS	Base station
CRL	Certificate revocation list	CK	Ciphering key
CRYPTREC	Cryptography research and evaluation committees	EAP	Extensible authentication protocol
eMBB	Enhanced mobile broadband	EPS	Evolved packet system
ECIES	Elliptic curve integrated encryption scheme	FSM	Finite state machine
GE	Gate equivalent	HE AV	Home environment authentication vector
HetNet	Heterogeneous network	HSS	Home subscriber server
IBE	Identity-based encryption	ICB	Initial counter block
IK	Integrity key	IMSI	International mobile subscriber identity
IoT	Internet of Things	ITU	International telecommunication union
KEM	Key encapsulation mechanism	LFSR	Linear feedback shift register
LPN	Learning parity with noise	LWC	Lightweight cryptography
LWE	Learning with errors	MAC	Message authentication code
MEC	Mobile edge computing	NEA	New-radio encryption algorithm
NFV	Network function virtualization	NF	Network function
NRF	NF repository function	NIA	New-radio integrity algorithm
NIST	The National Institute of Standards and Technology	NFSR	Nonlinear feedback shift register
PKI	Public key infrastructure	PRNG	Pseudorandom number generator
PQC	Post-quantum cryptography	RAN	Radio access network
RFID	Radio-frequency identification	SAGE	Security algorithms group of experts
SBI	Service-based interface	SEAF	Security anchor function
SDN	Software defined networking	SIDF	Subscription identifier de-concealing function
USIM	Universal subscriber identity module	SIMD	Single instruction multiple data
SUCI	Subscription concealed identifier	SUPI	Subscription permanent identifier
TLS	Transport layer security	UDM	Unified data management
UE	User equipment	UPF	User plane function

2 5G security architecture and mechanisms

In this section, we describe the 5G security architecture and mechanisms from the 3GPP specifications. We first give an introduction of the 5G architecture, showing how the access network and core network would be evolved and then present the security mechanisms based on the evolved architecture.

2.1 5G architecture

A mobile communication network consists of two parts: the radio access network (RAN) and core network. A RAN connects individual devices through radio connections with their core networks, while the core networks provide services to these users. Compared to the EPS (evolved packet system) architecture in LTE, the pivotal evolution of 5G architecture is the wide adoption of cloud and virtualization technologies to support diversified and flexible services. Existing mobile network architectures were mainly designed to meet the requirements for voice and broadband services, which has proven to be insufficiently flexible in 5G with diversified nodes, interfaces, and services. This becomes one driving force behind leading to the softwarized architecture of 5G. With SDN and NFV technologies being able to support and manage the underlying physical infrastructure, it becomes possible to virtualize the network functions and move them to the cloud and perform the central control, processing and management there. Compared to legacy cellular networks where a large variety of proprietary nodes and dedicated hardware appliances are deployed, the softwarized architecture can reduce the equipment and deployment cost and improve the flexibility and availability to the management and evolution. Furthermore, network slicing makes it possible to design isolated virtual networks dedicated to different services as needed, e.g., vehicular network service, over a single physical architecture, thus satisfying the different requirements of diversified services.

Figure 1 shows the 5G architecture reference model from 3GPP specification [5], where the components in the architecture are called network functions (NFs), which used to be different physical elements. In

Table 2 Summary of some references

Aspect	References	Main contribution
5G Security	[5]	3GPP specification of system architecture for 5G
	[6]	3GPP specification, describes the security features and mechanisms to bootstrap authentication and key agreement for application security
	[7]	3GPP specification of the security architecture, i.e., the security features, security mechanisms and the security procedures
	[8]	3GPP specification, specifies the need for cryptographic algorithms with the 256-bit security level
	[9]	3GPP specification, defines the principal purpose and use of different naming, numbering, addressing and identification resources
	[4, 10]	Huawei whitepapers about 5G architecture and security
5G confidentiality and integrity protection	[11]	Proposes an attack on SNOW 3G with complexity 2^{177}
	[12]	Proposes an attack on ZUC with complexity 2^{236}
	SNOW-V [13]	Proposes a new algorithm SNOW-V for 5G use
	ZUC-256 [14]	Specifies the 256-bit version of ZUC
	AES [15]	Specifies the AES algorithm
LWC	[16–19]	ISO/IEC standards for lightweight block ciphers, stream ciphers, hash functions, and asymmetric mechanisms
	[20]	Japan CRYPTREC guideline for lightweight cryptography
	[27]	Outlines some techniques that are defined as replacements for conventional cryptography; discusses some trends in the design of lightweight algorithms
	[21]	Identifies several trends in the design of lightweight algorithms; discusses more general trade-offs facing the authors
Post-quantum LWC	[22–24]	Respectively specifies HB, HB$^+$, HB^{++} protocols
	[25]	Specifies the Lapin protocol
	[26]	Extended protocols of HB protocols and their applicability in practice
	[28]	Lightweight schemes based on LWE

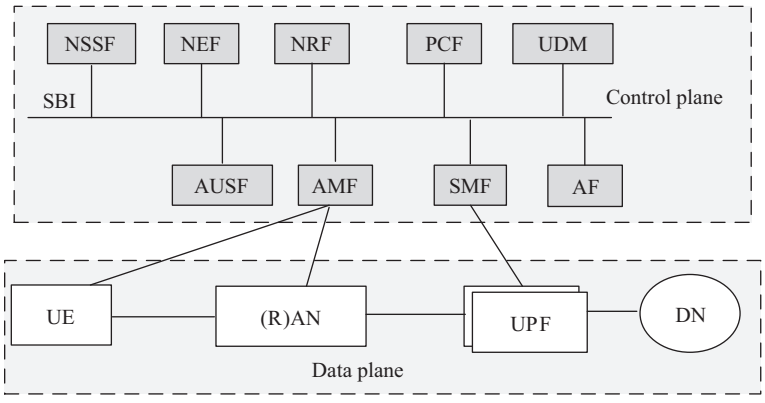

Figure 1 5G system architecture.

5G, these network functions are virtualized and software-based to be services, and thus can be well integrated to the cloud architecture. User equipment (UE) denotes a user and RAN is the radio access network. DN is the data network and the other components are the network functions residing in the core networks. Among them, user plane function (UPF) is a function working in the data plane, while others are in the control plane. Network functions are connected by reference points or service-based interfaces. Below we give some details to the access network and the core network.

2.1.1 *C-RAN*

Cloud/centralized radio access network (C-RAN) can be used in 5G for the radio access network, by utilizing the cloud and virtualization technologies to virtualize and centralize some functions of the base

stations to the cloud, and thus reducing the cost on the deployment and management of the largely increased and densified base stations. The RAN then consists of distributed sites and a cloud center. The RAN real-time functions, mainly at the physical layer and lower MAC layer, e.g., access network scheduling, interference coordination, modulation and coding, are still performed at the sites with dedicated hardware support; while some RAN non-real-time functions in the upper layers with low latency requirements, like intercell handover, cell selection/reselection, user-plane encryption, could be moved to the cloud, where the resources can be shared and information exchanged [4].

This cloudification of RAN will also affect some other aspects of the network. For instance, C-RAN indicates that in 5G, many functions in the RAN which used to be implemented in hardware with specialized hardware support, e.g., IP cores, will now be possible to be implemented in a software environment. It is important to guarantee their efficiency in this case. The implementation of confidentiality and integrity algorithms is such an example and this becomes one of the reasons to consider new software-efficient algorithms for 5G use. The 3GPP has recommended that ETSI SAGE (security algorithms group of experts) starts to evaluate software-efficient 256-bit cryptographic algorithms for 5G [8,10]. We list some cryptographic primitives which might be good candidates in Section 3.

2.1.2 *SBA-based core network*

The architecture of the core network in 5G is defined as a service-based architecture (SBA), where the system functionalities are defined as a set of network functions, like the session management function (SMF), and the access and mobility management function (AMF) in Figure 1. These NFs provide services to other authorized NFs through uniform service-based interfaces (SBI). A special network function called NF repository function (NRF) is introduced in the core network to deal with the service registration and discovery, and maintain NF profile and available NF instances, so that NFs can discover and access each other. Such a service-based architecture enables the usage of network slicing technology to create optimized network for specific services with different performance requirements. Below are some network functions related to the security aspect. For more details and other network functions, we refer to [5].

Unified data management (UDM) stores the keying materials of subscribers, like the long-term key(s) and the home network private key. It also hosts some functions related to data management, like the authentication credential repository and processing function (ARPF), subscription identifier de-concealing function (SIDF). ARPF is responsible for selecting the authentication mechanism based on the subscriber identity and configured policy and computing the 5G home environment authentication vector (HE AV) during an authentication. The SIDF provides service on decrypting a subscription concealed identifier (SUCI) of a user to obtain its long-term identity subscription permanent identifier (SUPI). The details of this process are given in Subsection 2.2.1. One can see the UDM is analogous to the home subscriber server (HSS) in LTE.

The authentication server function (AUSF) is a function in the core network responsible for handling authentication requests for both 3GPP access and non-3GPP access. It stores the long-term subscriber identities and performs full authentication with the UEs. The full authentication process will be given in Subsection 2.2.2.

The security anchor function (SEAF) resides in a visiting network serving as a 'middleman' during the authentication process between a UE and its home network. It can reject an authentication from the UE, but relies on the UE's home network to accept the authentication. It also holds the root key for a visiting network (known as anchor key) to derive other sub-keys to protect signaling and messages happening in this visiting network.

The access and mobility management function (AMF) can be collocated with the SEAF. It receives all connection and session related messages from users but only deals with the connection and mobility management tasks.

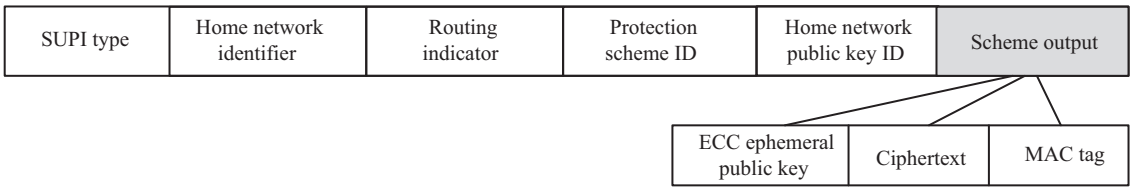

Figure 2 Structure of SUCI.

2.2 Security in 5G

The security mechanisms are becoming tougher and tougher to keep pace with the evolvement of the communication systems. However, no matter how they evolve, the security architectures in mobile communication always focus on the following security features: user identity management to provide identity privacy; mutual authentication between users and networks; key management and derivation to provide confidentiality and integrity protection of data. Below we give details of how these features are achieved in 5G and for more details, we refer to [7].

2.2.1 *Privacy security*

A universal subscriber identity module (USIM) is usually used to manage user keying materials, e.g., identity and the long-term key, at the user side. Each UE has an identifier, called SUPI in 5G (in 3G/LTE, this identifier is often known as the international mobile subscriber identity (IMSI)), which is globally unique and assigned at the manufacturing phase. The SUPI along with other subscription materials is integrity protected within the USIM using a tamper resistant secure hardware component. When performing the primary authentication, a user needs to send its identifier to the network to prove its identity. In legacy cellular networks, e.g., 3G and LTE, the identifier is transmitted over the air by plaintext, making it possible for an attacker to capture it through eavesdropping. In 5G, this threat is prevented through concealing the SUPI with the public key of the home network, which is securely provisioned for the home network and stored at the USIM, to derive a temporary concealed identifier, called SUCI. Then the SUCI, instead of the SUPI, will be transmitted to the home network to be used for performing an authentication. Only the SIDF at the home network holding the correct private key can de-conceal the SUCI to recover the SUPI and verify the user's identity.

The concealment process in 5G uses elliptic curve integrated encryption schemes (ECIESs), which are implemented at both the UE side and the home network side. Upon a new identity request, the UE will freshly generate a SUCI and send it to the network. The structure of a SUCI according to [9] is shown in Figure 2, which includes the information about the SUPI type, home network identifier, routing indicator, protection scheme ID (identifier), home network public key ID, and finally the scheme output of the concealment of the SUPI given a protection scheme.

Ref. [7] specified two supported ECIES protection schemes for 5G, which are respectively Diffie-Hellman primitive X25519 and elliptic curve cofactor Diffie-Hellman primitive. The two supported schemes are implemented both in the mobile equipment (ME) and the SIDF, and are chosen based on the Protection Scheme ID in the SUCI. The calculation of the SUCI is either performed at the USIM or the ME according to the operator. We have to mention here that the two profiles are broken by Shor's algorithm and a new profile resistant against attackers with quantum computers will be required [8]. However, at the time of this writing, as explained in [8]: "Currently there are no recommended quantum safe algorithms to replace existing asymmetric key agreement and key encapsulation mechanisms...It is not yet clear what classes of algorithm will be favoured by the National Institute of Standards and Technology (NIST) and as such it is difficult to predict what key sizes and ciphertext sizes the system will be required to support", the recommendation is "wherever the SUCI is included in a message, the field is suitably sized". Despite this, we still show the procedure of how the (de)concealment is done below, since this is new in 5G compared to existing mobile networks, and for more details, we refer to [7, 29–31].

The concealment process to generate the scheme output at the user side is as follows [29].

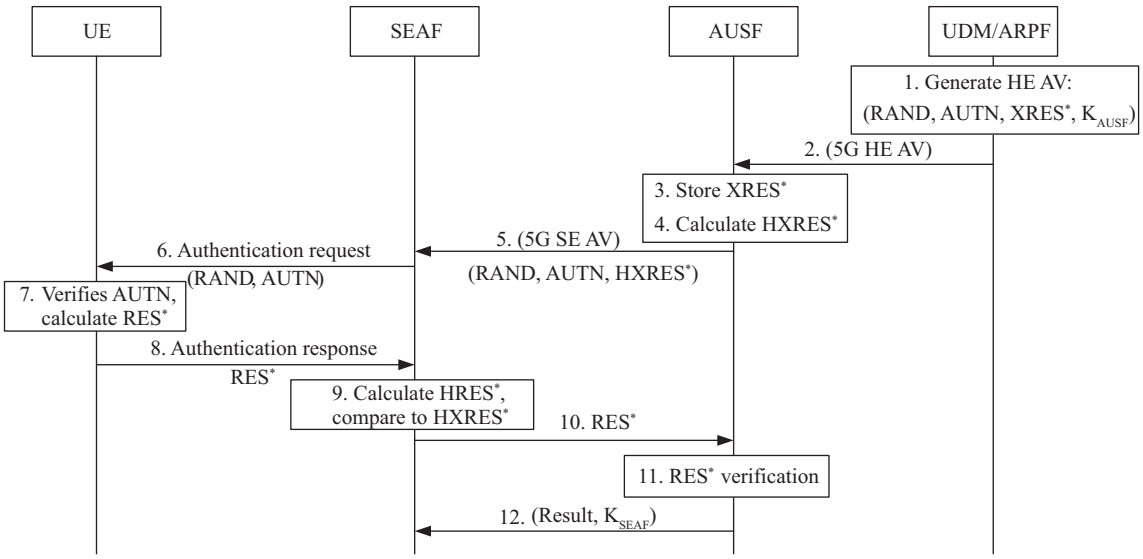

Figure 3 The authentication procedure of 5G-AKA.

(1) Generate a pair of elliptic curve cryptography (ECC) ephemeral public/private key pair associated with the elliptic curve domain parameters of the chosen ECIES profile. Usually, the secret key is randomly or pseudorandomly selected and the public key is derived according to the secret key.

(2) Derive a secret key based on the public key of the home network and the ephemeral private key generated in step 1 using the Diffie-Hellman primitive of the chosen ECIES profile, and convert the derived secret key to an octet string.

(3) Derive a key using the key derivation function (ANSI-X9.63-KDF in both ECIES profiles) from the secret key generated in step 2 with a specified length and parse the leftmost and middle parts of the key as the encryption key and an initial counter block (ICB) to encrypt the plaintext using AES-128 in counter mode, and the rightmost part as a message authentication code (MAC) key to generate the tag for integrity protection using HMAC-SHA-256.

Then the scheme out consists of the three parts: ECC ephemeral public key, ciphertext value and the MAC tag value. The maximum size of the scheme out is chosen to allow for the introduction of quantum-resistant protection schemes.

At the home network side, the same ECIES schemes should be implemented and the decryption operation is performed to de-conceal the SUCI and recover the SUPI.

2.2.2 *Authentication*

5G supports two ways of authentication, which are respectively called 5G-authentication and key agreement (5G-AKA) and extensible authentication protocol (EAP)-AKA′ [7]. It is the UDM/ARPF/SIDF at the home network side who decides which authentication method to be chosen. When a user initiates an authentication, it will send an registration request with the SUCI included to the UDM, as described in Subsection 2.2.1. After receiving the request, the SIDF in the UDM de-conceals the SUCI to recover the SUPI and checks the validity. The UDM/ARPF/SIDF then further chooses which authentication mechanism to use according to the subscription information of the UE. Below we give some details of the authentication procedures of 5G-AKA and EAP-AKA′. We ignore some signaling messages but only focus on the authentication messages for simplicity.

5G-AKA. 5G-AKA enhances EPS AKA by providing the home network with the proof of successful authentication of the UE from the visiting network. Figure 3 shows the authentication procedure of 5G-AKA and below are the main steps (the numbers denote the steps in Figure 3).

(1, 2) When requested, the UDM/ARPF shall generate a home network authentication vector HE AV = (RAND, AUTN, XRES*, K_{AUSF}) and send it to the AUSF. RAND is a random number and AUTN is

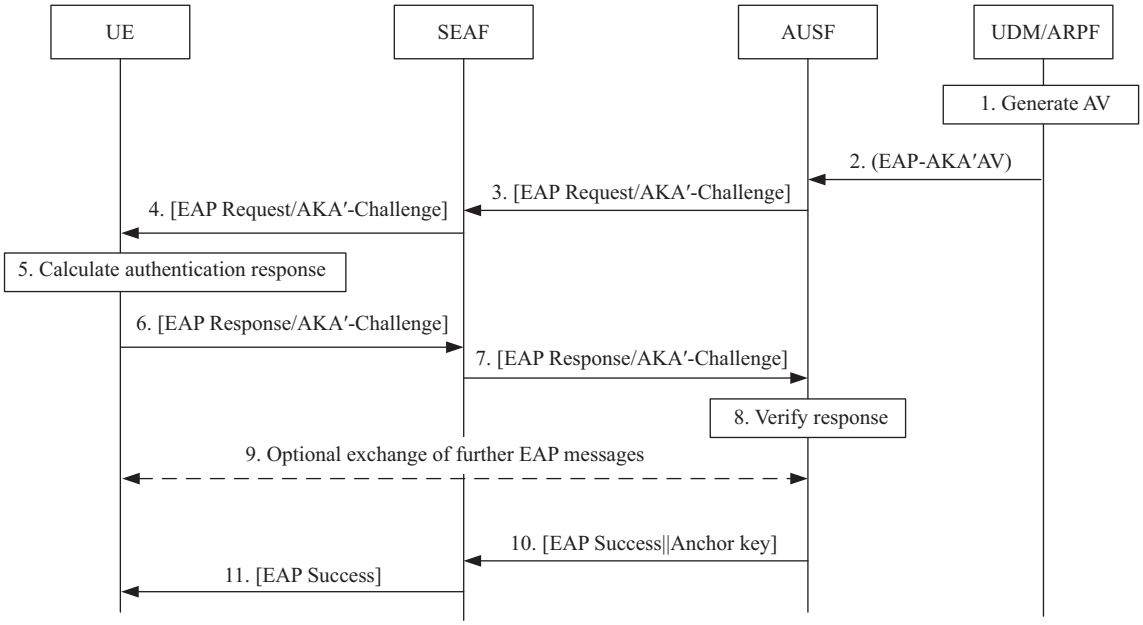

Figure 4 The authentication procedure of EAP-AKA′.

the authentication token. K_{AUSF} and XRES* are generated both using the key derivation function KDF whose detail will be given in Subsection 2.2.3, with the input key as the concatenation of ciphering key (CK) and integrity key (IK), which are two keys derived through the long-term key.

(3–5) When receiving the HE AV, the AUSF shall store XRES* temporarily and further produce a serving network authentication vector SE AV = (RAND, AUTN, HXRES*) and a key K_{SEAF}. HXRES* and K_{SEAF} are respectively derived from XRES* by the SHA-256 hashing algorithm and K_{AUSF} by the KDF algorithm which will be described in Subsection 2.2.3. The SE AV is then sent to the SEAF and the latter forwards (RAND, AUTN) to the UE with an authentication-request message.

(7, 8) The USIM inside the ME verifies the freshness of the authentication vector by checking if AUTN is valid. If so, it computes a response RES, the ciphering key CK and the integrity key IK to the ME and the latter computes RES* from RES using the KDF algorithm. The user then returns RES* to the SEAF.

(9, 10) The SEAF computes HRES* from RES* and compares it with the stored HXRES*. If being the same, the SEAF will send RES* to the AUSF indicating the successful authentication of the user from the serving network point of view.

(11, 12) The AUSF shall compare the received RES* with the stored XRES*. If being the same, the AUSF shall consider the authentication as successful from the home network point of view and indicate the authentication result to the SEAF.

During the whole authentication process, some sub-keys will be generated for confidentiality and integrity protection for subsequent messages.

EAP-AKA′. EAP-AKA′ is also supported in 5G. Under the EAP framework, the UE is the authentication peer, the SEAF takes the role of the pass-through authenticator and the AUSF acts as the backend authentication server. Figure 4 shows the authentication procedure of EAP-AKA′.

(1, 2) The UDM/ARPF shall first generate an authentication vector and subsequently send this transformed authentication vector AV' = (RAND, AUTN, XRES, CK′, IK′) to the AUSF. CK′ and IK′ are two keys derived from CK and IK.

(3, 4) The AUSF shall send the EAP-Request/AKA′-Challenge message to the UE via SEAF transparently forwarding it.

(5–7) The UE verifies the freshness of AV' by checking whether AUTN can be accepted. If so, it computes a response RES and sends the EAP-Response/AKA′-Challenge message to the AUSF via the SEAF transparently forwarding it.

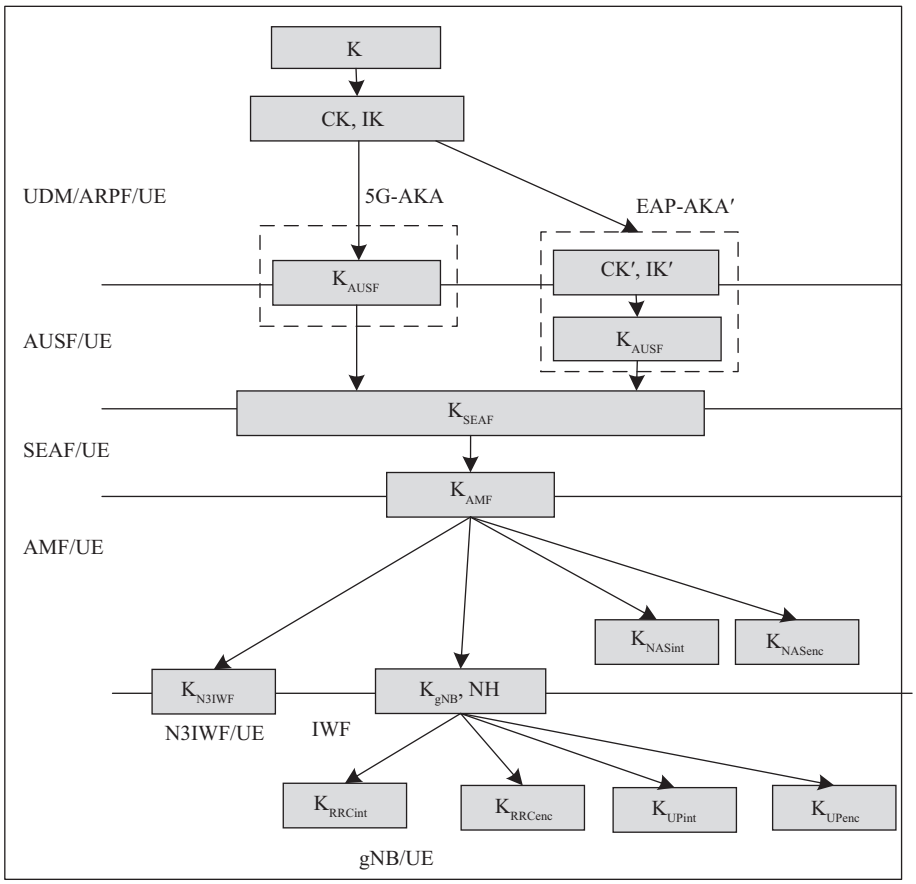

Figure 5 Key hierarchy generation in 5G.

(8–11) The AUSF would verify the response and notify the UDM and UE of the successful authentication.

2.2.3 *Key hierarchy and derivation*

Key derivation. All the key derivations in 5G core network are performed using the Keyed-Hashing for message authentication (HMAC) specified in [6, 32, 33] presented as

$$\text{derived key} = \text{HMAC-SHA-256(Key, } S),$$

i.e., the HMAC algorithm with SHA-256 as the hash function, where Key is the input key and S is a string constructed from the $n+1$ input parameters as follows: $S = \text{FC}\|P_0\|L_0\|\cdots\|P_n\|L_n$. Pi's are the input parameter encodings while Li's are the lengths of Pi's. The parameter FC is controlled as specified in [6] to differentiate between various purposes of KDF in 3GPP system. Using HMAC-SHA-256 to derive (up to) 256-bit keys from 256-bit keys is in line with standard advice about resisting against quantum computers [8].

Key hierarchy. Figure 5 shows the key hierarchy in 5G. The granularity of keys in 5G is higher than that in 3G or LTE. With some keys inherited from 3G/LTE being kept, some additional keys are introduced. The USIM and UDM/ARPF stores the long-term key K, which could expect to be 256-bit, and would use it to derive more sub-keys with the same key derivation function as described before.

The keys related to authentication include K, CK/IK and further derived CK′, IK′ in case of EAP-AKA′. After successful authentication, more keys are derived to protect subsequent messages. Below we give some description of some of them.

K_{AUSF} is a key generated during the authentication. In 5G-AKA, it is generated from CK, IK at the ARPF and included in the HE AV being sent to the AUSF; while in EAP-AKA′, it is generated at the AUSF locally from CK′, IK′. Every time when there is a new authentication request, a new K_{AUSF} should be derived and used to derive the anchor key K_{SEAF}.

K_{SEAF} is an anchor key bound to a serving network during the primary authentication to prevent one serving network from claiming to be a different one. When generating K_{SEAF}, the parameter called "serving network name" should be included into the input of the key derivations, to make sure that the anchor key is specific for authentication between a 5G core network and a UE.

K_{AMF} is a key derived by the ME and SEAF from K_{SEAF}. It will be used to generate the integrity keys K_{NASint} and encryption keys K_{NASenc} for non-access stratum (NAS) signaling with a particular integrity/encryption algorithm which will be described later, keys for RAN K_{gNB}, and keys for the non-3GPP access K_{KN3IWF}.

K_{gNB} is used to derive a new K_{gNB} when performing horizontal or vertical key derivation during handovers, and the keys K_{UPenc}, K_{UPint}, K_{RRCenc}, K_{RRCint}, which are respectively used for the confidentiality and integrity protection of user plane (UP) traffic and radio resource control (RRC) signaling. These keys will be used as the root keys to generate the keystreams to provide confidentiality and integrity protection with the specified new-radio encryption algorithm (NEA)/new-radio integrity algorithm (NIA) algorithms (corresponding to the EPS encryption algorithm (EEA) & EPS integrity algorithm (EIA) in LTE). There are three confidentiality and integrity algorithms being specified in LTE, whose cryptographic cores are respectively AES-128 (in counter mode), SNOW 3G and ZUC-128, all with a 128-bit security level and we refer to [3] for details. These cryptographic cores might continue to be used in 5G, but 3GPP also asks the SAGE group to select and evaluate new possible cryptographic primitives with a 256-bit security level for 5G use [8]. We will list some cryptographic primitives which could be good candidates in Section 3.

When employing the confidentiality and integrity protection algorithms, a key and an initialization vector (IV) will be input to the algorithms and a random-like sequence, called keystream will be generated. In LTE, the key is 128-bit and the IV is 38-bit: a 32-bit counter COUNT, a 5-bit bearer identity BEARER and a 1-bit DIRECTION indicating the direction of transmission [3]. While in 5G, the length of the key can be expected to be 256-bit, and for the IV, SAGE is trying to suggest to SA3 using 128-bit: an additional 90-bit random value alongside COUNT, BEARER and DIRECTION, to protect better against multi-target attacks [34]. The keystream will then be used to encrypt/decrypt the messages by simple xor operation, and to generate the authentication tag to provide the integrity protection of the messages. Usually, the allowed length of the keystream corresponding to one IV value is restricted to resist some attacks, e.g., distinguishing attacks.

3 5G potential confidentiality/integrity algorithms

The driving forces behind the demand for new confidentiality/integrity algorithms are from two aspects. Firstly, as described in Subsection 2.1.1, the upper layers of RAN in 5G can be cloudified and the confidentiality/integrity protection operations would likely be moved to the cloud and implemented there under the software environment without specialized hardware support. This makes it challenging for existing confidentiality/integrity algorithms to achieve the targeted speed of 20 Gbps downlink in 5G, with an exception of AES. The other driving force is that now 3GPP standardization organization is looking towards increasing the security level to 256-bit to resist against quantum computing [8]. For example, the Grover's search algorithm offers a theoretical quadratic speed-up on unstructured search problems, and thus can recover an N-bit key with complexity $O(2^{N/2})$ [8]. It might be risky if we just adopt current confidentiality/integrity algorithms directly for the 256-bit security level without careful inspection.

Below we give some details of four cryptographic primitives which might be potential candidates for confidentiality/integrity protection in 5G based on the information from [35, 36], which are respectively

Table 3 Some performance results of AES-256, SNOW 3G, SNOW-V, and ZUC-256[a)]

Ciphers	Software Env. (plaintext sizes)				Hardware implementation		Attacks
	4096	2048	1024	256	Area	Throughput	
AES-256	34.16 [13]	32.94	30.95	22.67	17232 GEs [37]	50.85 [37]	$2^{254.4}$ [38]
SNOW 3G (256-bit)	8.89 [13]	8.50	7.81	5.38	18100 GEs [39]	52.8 [39]	2^{177} [11]
SNOW-V (256-bit)	54.60 [13]	50.70	45.28	26.37	13041 GEs [13]	358 [13]	2^{256} [13]
ZUC-256	3.50 [40]	3.39	3.17	2.29	12500 GEs [41]	80 [41]	2^{236} [12]

a) The implementations are under different platforms or resources and it is unreasonable to compare them directly according to the figures shown here. We refer to the given references for more details. Columns 2–5 are the throughput under different plaintext sizes: all throughput is measured in Gbps and plaintext sizes are in bytes.

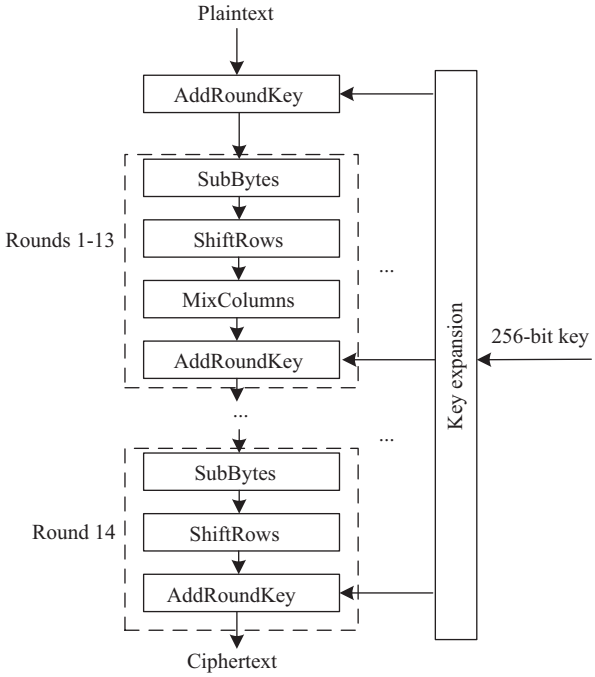

Figure 6 The overall schematic of AES-256.

AES-256, SNOW 3G, SNOW-V, and ZUC-256. Table 3 [11–13,37–41] presents some performance results under software/hardware implementations and the best attacks till now for these ciphers. The hardware implementation area is evaluated in gate equivalent (GE), which is equivalent to the physical area of a single NAND gate. However, it is unreasonable to compare the performance of these ciphers only according to the figures in the table, because the implementation platforms and details may vary. Instead, we refer to the references given in the table from which these results are derived for the details.

3.1 AES-256

AES is a block cipher encrypting and decrypting data in blocks of 128 bits and supports three key lengths, i.e., 128-bit, 192-bit and 256-bit. It repeatedly performs four transformations, which are serially `SubBytes`, `ShiftRows`, `MixColumns`, `AddRoundKey` and these constitute one full round. When working under different key lengths, the required numbers of rounds are different: 10 rounds for AES-128, 12 rounds for AES-192 and 14 rounds for AES-256. Figure 6 shows the overall schematic of AES-256. The `Key Expansion` operation derives round keys from a 256-bit secret key. Before entering each round, the initial round key is bitwise xor-ed with the plaintext block, which is known as "key whitening". After that, the cipher runs 14 rounds, and specially, the `MixColumns` operation is omitted in the last round. Other versions have the similar structure with the key lengths and numbers of rounds being different.

In LTE, AES-128 is used as the core of the confidentiality and integrity algorithms 128-EEA2 & 128-

EIA2. It is respectively used in counter mode for confidentiality protection and in cipher-based message authentication code (CMAC) mode to produce a message authentication code for integrity protection [15].

AES is so popular that it has received special support from mainstream CPU vendors in the form of intrinsic instructions, which makes it fast even in a software environment. For example, some Intel, AMD, and ARM CPUs have been integrated with AES SIMD instructions to provide user-level instructions implementing AES rounds. Much security and cryptography software supports the AES instruction set as well, e.g., OpenSSL. This makes AES more advantageous than other ciphers, because it indicates AES could be quite efficient in the 5G cloudified system.

Considering these, one can expect that AES is highly likely to be kept in 5G, but with the 256-bit version. Ref. [8] mentioned that "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". Ref. [13] tested the throughput of AES-256 (in counter mode) in OpenSSL utilizing Intel advanced encryption standard new instructions (AES-NI) and other optimization techniques under software environment and Table 3 shows some results: it can achieve throughput higher than 20 Gbps when the plaintext sizes are larger than 256 bytes, satisfying the speed requirement of 5G. For the performance under hardware environment, a recent result from [37] provides an area-speed optimized implementation of AES-128 (10 rounds) on NanGate 15 nm technology with throughput of 71.19 Gbps and the area 17232 GEs. This means that with the same design, it is possible to achieve 50.85 Gbps for AES-256 (14 rounds) [13]. Till now, the best attack on AES-256 is a biclique attack resulting in key recovery with computational complexity $2^{254.4}$ [38], which is slightly better than the exhaustive key search.

3.2 SNOW 3G

SNOW 3G is a word-oriented stream cipher being used as the core of one of the confidentiality and integrity algorithms for both 3G and LTE networks [42] with the 128-bit security level. Figure 7 shows the overall schematic of SNOW 3G. It consists of a linear part linear feedback shift register (LFSR) and a non-linear part referred to as finite state machine (FSM), both defined over GF(2^{32}). The LFSR part consists of 16 cells, denoted as $(s_{15}, s_{14}, \ldots, s_1, s_0)$, each containing 32 bits and thus giving 512 bits in total; while the FSM has three internal 32-bit registers $R1, R2$, and $R3$, connected by operations like S-transform, arithmetic addition, and xor. At each iteration, the output of FSM is xor-ed with s_0 from the LFSR to generate a 32-bit keystream symbol. After that, the FSM and LFSR are respectively updated: the FSM takes s_{15}, s_5 from the LFSR and updates $R1, R2$, and $R3$; after that, the LFSR is updated with every value in a cell being shifted to the right one and s_{15} is updated according to the generating polynomial. This word-wise construction makes SNOW 3G efficient in both software and hardware environment. In [13], authors implemented SNOW 3G under the software environment, with throughput larger than 8.5 Gbps for plaintext with sizes larger than 4096 bytes. In a recent hardware integrated implementation of SNOW 3G and ZUC using 65 nm target technology library [39], it can achieve 52.8 Gbps for SNOW 3G standalone with around 18100 GEs. Till now, there is no efficient cryptanalysis against SNOW 3G with the 128-bit security level.

SNOW 3G might be kept in 5G but with the key size set to 256 bits [36], with the advantage that existing components in hardware can be reused. However, in [11], a correlation attack resulting in key recovery with complexity 2^{177} and a distinguishing attack with compleixty 2^{172} were proposed. This indicates that if the key length in SNOW 3G would be increased to 256 bits, the 256-bit security level for long keystreams cannot be achieved. Therefore, if SNOW 3G is to be used in 5G, some countermeasures should be carefully taken to resist against the attacks, for example, restricting the lengths of keystreams as done in 3G/LTE systems. However, the main drawback of SNOW 3G is its performance in software that might be a potential performance bottleneck for cloudified 5G system.

3.3 SNOW-V

The SNOW-V cipher was proposed in [13] in 2019 as the 256-bit version of SNOW to meet the 5G requirement on cryptographic primitives in terms of both the encryption speed and security level, with

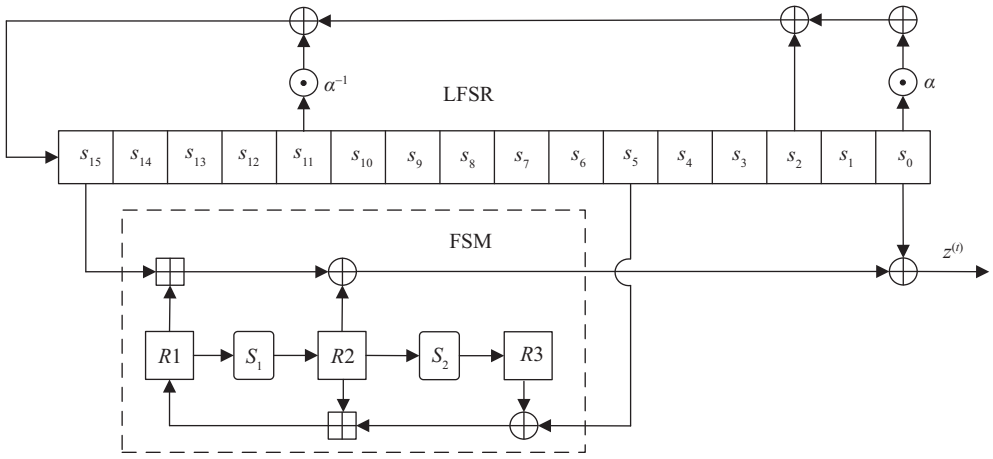

Figure 7 The keystream generation phase of the SNOW 3G cipher.

"V" standing for "Virtualization". The main goal of SNOW-V is to achieve high efficiency under software environment and to strengthen the security level to 256 bits. The general structure of SNOW-V inherits the design and security principles of SNOW 3G, with a linear LFSR and a non-linear FSM, but both are updated to better align with vectorized implementations to make it efficient under the software environment. SNOW-V also adopts a new security feature FP(1) in the initialization.

Figure 8 shows the overall schematic of SNOW-V. The LFSR part is a circular construction consisting of two LFSRs (named LFSR-A and LFSR-B) defined by two polynomials, each feeding into the other. They both have 16 states and every state is an element from GF(2^{16}), but with different generating polynomials, thus giving 512 bits in total. Such an LFSR structure can achieve the maximum cycle length of $2^{512}-1$. Each time when updating the LFSR, LFSR-A and LFSR-B are clocked eight times, which means 256 bits of the total 512-bit state will be updated, and after that two 128-bit taps $T1$ and $T2$ derived respectively from $(b_{15}, b_{14}, \ldots, b_8)$ and $(a_7, a_6, \ldots, a_0)$ will be fed to the FSM.

The FSM has three 128-bit registers $R1$, $R2$, and $R3$. Two full AES rounds are used serving as two large 128-bit S-boxes, a refactored version of 32-bit S-boxes used in SNOW 3G design. At each iteration, the FSM takes $T1$ and $T2$ from the LFSR as the inputs and produces a 128-bit keystream symbol. Then registers $R2$ and $R3$ are updated respectively from $R1$ and $R2$ through one full AES encryption round with the round key set to be zero, while $R1$ is updated from $R2, R3$ and $T2$ by a combination of xor, arithmetic addition and a permutation σ operations.

An AEAD mode of operation is also provided for SNOW-V employing Galois message authentication code (GMAC) integrity and authentication algorithm to further provide authentication to messages. A distinct advantage of GMAC in SNOW-V is that the H-key (the hash key) used in the function GHASH_H is newly generated for every (K, IV) pair, which is not the case in, e.g., AES-GCM mode, where the H-key is derived from the K only. Designers also give the software and hardware implementations and corresponding performance. For the software implementation, large registers and vectorized SIMD instructions, such as AVX2 set of instructions (intrinsics), are employed. The speed can be higher than 22 Gbps for encrypting plaintexts with sizes larger than 256 bytes. Four possible hardware implementations are also provided according to the resource of a device, and the throughput can be expected to be higher than 358 Gbps with around 13041 GEs. We refer to [13] for more details.

Both internal [13] and external [43] evaluations were made on SNOW-V and the results indicate it should be secure. In [44], authors made a deep and thorough analysis in regards to guess-and-determine attacks, and found an attack with complexity 2^{406} that establishes the upper bound for the security of SNOW-V, which also shows a good security margin of the algorithm.

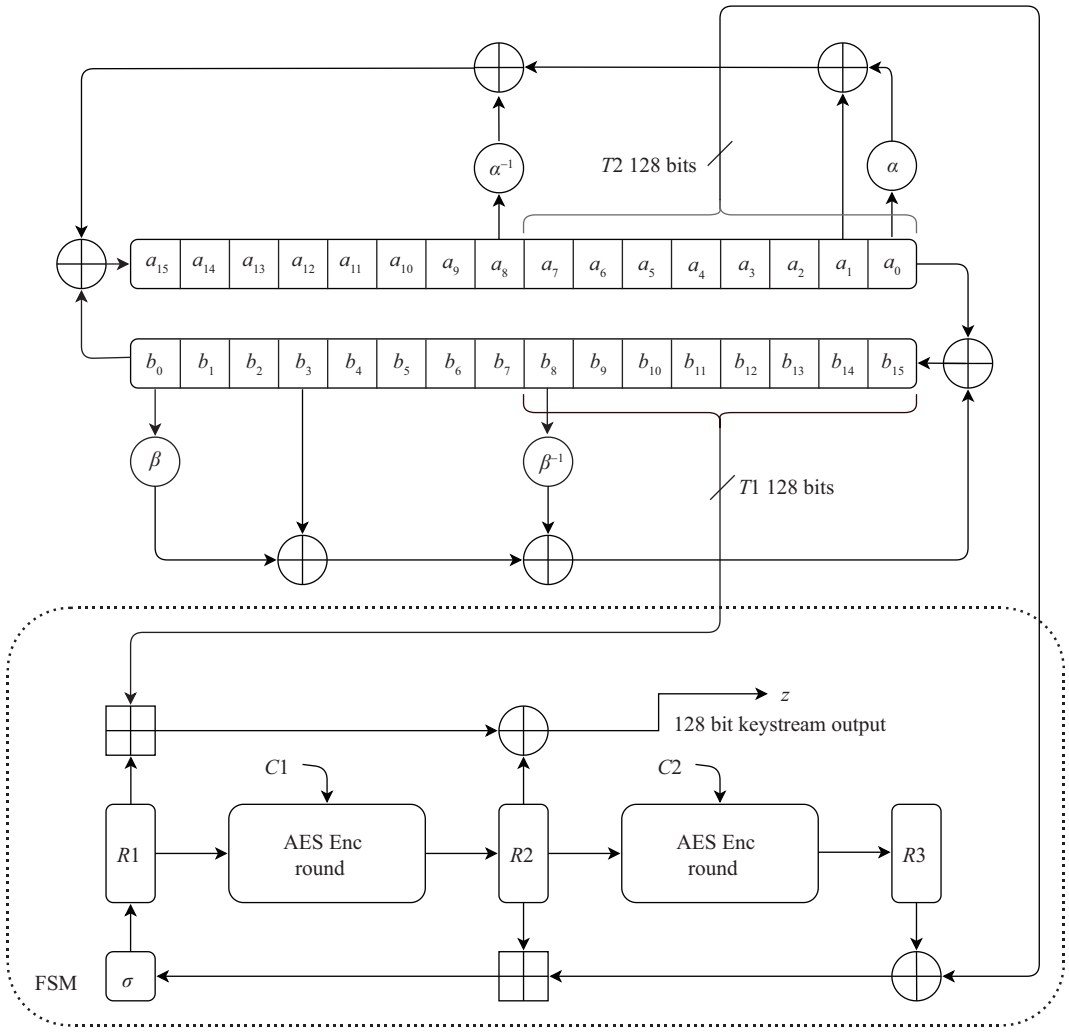

Figure 8 The keystream generation phase of the SNOW-V cipher.

3.4 ZUC-256

ZUC-128 is the core of the confidentiality and integrity algorithms 128-EEA3 & 128-EIA3 in LTE. ZUC-256 was proposed in January 2018 as the 256-bit version of ZUC [14], to satisfy the 256-bit security level requirement of 5G system. Different from SNOW-V which is almost re-designed from its predecessor SNOW 3G, ZUC-256 keeps the same structure as ZUC-128 while only the initialization and message authentication code generation phases are improved. This gives ZUC-256 some advantages in terms of reusability of existing hardware.

Figure 9 shows the overall schematic of ZUC-256 algorithm, which is the same as in ZUC-128. It consists of three layers: the top layer is a LFSR of 16 stages, the bottom layer is a nonlinear function referred to as F, while the middle layer called bit-reorganization (BR) layer is a connection layer between the LFSR and F. Different from common stream ciphers which are usually defined over binary fields GF(2) or extension fields of GF(2), the LFSR in ZUC is defined over a prime field GF(p), with $p = 2^{31} - 1$ and this makes it more complicated for cryptanalysis. Every time when updating, the value in a cell shifts to the right one and s_{15} is updated according to the generating polynomial. The BR layer extracts some lower or higher 16-bit parts from some states in the LFSR and forms four 32-bit words $X0, X1, X2, X3$ with the first three being fed to F and the last one xor-ed with the output of F to finally generate the keystream symbol. Registers in F are defined over GF(2^{32}) and updated by a combination of operations like arithmetic addition, xor, cyclic shift ($\lll 16$) and S-transform ($S * L_1, S * L_2$). For more details on the design of

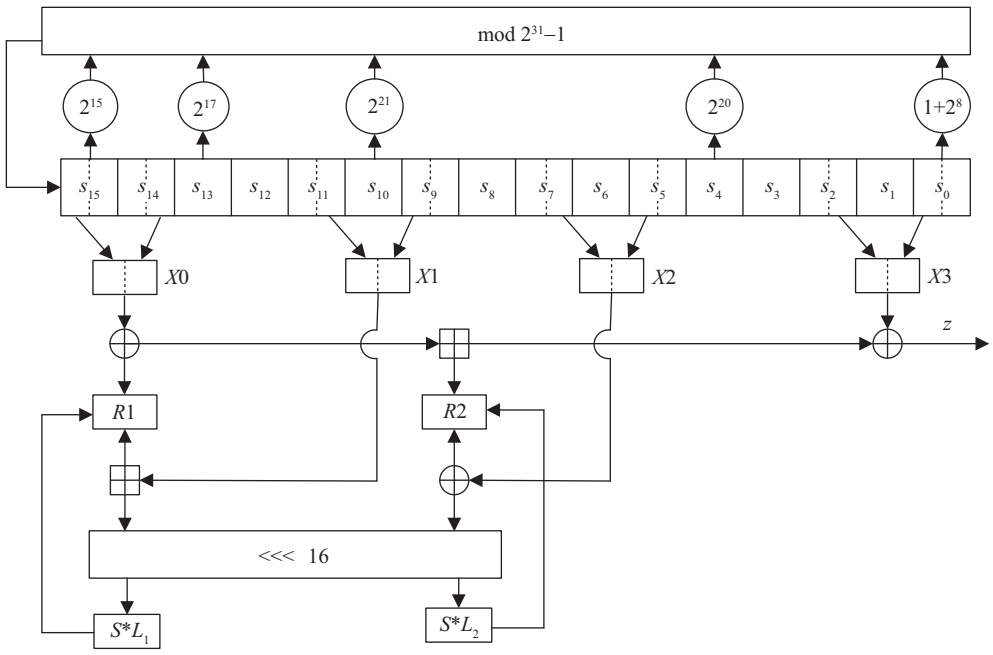

Figure 9 The keystream generation phase of the ZUC-256 cipher.

ZUC-256, we refer to the original design document [14] and the specification document of ZUC [45]. Ref. [40] gives a software implementation of ZUC, where it can achieve speed 0.7547 bits/cycle, i.e., 3.17 Gbps if we consider a CPU @4.20 GHz as in [13], for plaintext with sizes larger than 1024 bytes [40]. A recent hardware implementation of ZUC using pipeline architecture on 65 nm ASIC technology [41] can provide throughput up to 80 Gbps with 12500 GEs.

In July 2018, a workshop on ZUC-256 was held and some general cryptanalysis were presented, but no obvious weaknesses of ZUC-256 were found at that time. However, in November 2019, the authors in [12] gave a distinguishing attack on ZUC-256 with complexity $O(2^{236})$. The authors specified in the paper that "although the attack is only 2^{20} times faster than exhaustive key search, the result indicates that ZUC-256 does not provide a source with full 256-bit entropy in the generated keystream, which would be expected from a 256-bit key". Similarly, like the SNOW 3G case, the attack might not pose a immediate threat for ZUC-256 as long as the length of the keystream corresponding to one IV value is restricted. Perhaps, it would be good if ZUC could perform faster in the virtualization environments, but on the other hand it can reuse existing components in hardware, which is another type of advantage.

4 Lightweight cryptographic primitives

Conventional cryptographic primitives, e.g., the ones mentioned in Section 3, are designed for 'large' devices, e.g., servers, desktops, and smart phones, which do not need to worry much about the resources. The performance is the main focus for these cases. While in 5G, there are massive devices among which many are resource-constrained, e.g., the nodes in embedded systems, radio-frequency identification (RFID) and sensor networks, in terms of physical size, power supply, and storage. Such devices usually have difficulties affording conventional cryptosystem and there is a high demand for lightweight cryptographic primitives. The research on lightweight cryptography (LWC) targeting this problem has been on the rise in recent years.

While there is no generally agreed strict definition on LWC, the criteria below from [46] can be used to tell what a 'lightweight' cryptographic primitive could be like. The main concern of LWC is about the implementation area, the power consumption, and the throughput.

Criteria. A cryptographic primitive is said to be lightweight if the hardware area of the implementation is less than 2000 GEs, its power consumption is very small and it supports a sustained throughput

of 1 bit per clock cycle at a clock speed of 2 MHz. For a cryptographic primitive together with a mode (e.g., authenticated encryption), the GE requirement will be loosened up to 3000 GEs.

It is long commonly set that modern lightweight ciphers should occupy less than 2000 GEs [26]. The power consumption quantifies the amount of power needed to use the cipher, which especially matters for battery-powered devices. Throughput is the amount of data processed per time unit.

There are also some other factors that might matter for some specific devices, e.g., latency, which corresponds to the time taken to get the keystream output. Low latency is required for real-time applications, e.g., on-vehicle devices. For some LWC designed for software environment, the memory size (RAM, ROM), corresponding to the amount of storage needed during each instance of the cipher, and code size also matter.

Because there is always a trade-off between the performance, security, and implementation cost for a cipher, lightweight ciphers usually indicate simpler operations and consequently lower security margins. It is important to guarantee the security satisfies an expected security level. They should be resistant to common attacks, like differential/linear cryptanalysis, guess and determine attacks, etc.

4.1 LWC activities

Till now, there are several national or international activities aiming to select and evaluate lightweight cryptographic systems, and some schemes have been recommended, standardized or even used in commercial products. ISO/IEC issued a standard with number 29192 targeting specially on LWC, and the standardized algorithms include: for example, the block ciphers [16], PRESENT [47], CLEFIA [48]; the stream ciphers [17], Trivium [49] and Enocoro [50]; the hash functions [18], PHOTON [51], Spongent [52] and Lesamnta-LW [53]; and mechanisms using asymmetric techniques [19]. Some algorithms standardized in other projects can also be regarded as lightweight, e.g., Grain-128a [54], MISTY1 [55], and HIGHT [56].

In 2004, European eSTREAM project was initiated to select efficient stream ciphers targeted for restricted hardware environment and high-speed software environment. Some stream ciphers selected from this project can be regarded as lightweight, like bit-based Grain, Trivium. The Japanese Cryptography Research and Evaluation Committees (CRYPTREC) is also working on LWC: the Lightweight Cryptography Working Group was established in 2013 and they published a comprehensive technical report about LWC in 2017 [20].

In 2015, NIST also initiated a standardization project called "Lightweight Cryptography" to call for, evaluate, and select lightweight cryptographic algorithms [1)]. There were 57 proposals submitted originally and 56 of them were selected as Round 1 candidates; after the first evaluation phase, 32 of the 56 survived to continue to Round 2 in August, 2019. The Round 2 evaluation is expected to last twelve months from September, 2019.

4.2 Design trends and promising LWC primitives

The design of symmetric LWC primitives can be mainly based on several constructions, like Feistel networks, substitution-permutation network (SPN), feedback shift register (FSR), addition/rotation/XOR (ARX) [57]. From these primitives, one can generalize some commonly used operations and design trends taking the implementation aspect into consideration. For example, small S-boxes, e.g., 5-bit, 4-bit, 3-bit, and modular additions are usually the main components used to provide nonlinearity, while for the linear operations, maximum distance separable matrices, bit-permutations, and XOR are commonly used; the algorithms are usually using smaller internal states, shorter blocks and key sizes. People are also trying to find asymmetric LWC systems for, e.g., authentication and key management. ECC is considered as a promising primitive for these systems because it has a relatively shorter key length and thus achieves a faster processing speed while requiring less memory.

Below we give a general review of the LWC design in different categories and give some popular lightweight cryptographic primitives. For more details, we refer to [20, 21].

1) NIST lightweight cryptography project. https://csrc.nist.gov/projects/lightweight-cryptography.

4.2.1 *Block ciphers*

Block ciphers are the most common choice to build a lightweight symmetric cipher. Some popular lightweight block ciphers are like SIMON, XTEA, HIGHT, PRESENT, LED, KATAN, SPECK, etc. From these ciphers, one can see that the block and key sizes are getting reduced: most lightweight block ciphers use only 64-bit blocks and the key sizes are usually 80-, 96-, or 128-bit. But in the NIST LWC project, the key size is required to be at least 128-bit.

Block ciphers are often based on the Feistel structure or SPN. The non-linearity is still mainly provided by S-boxes, but much smaller, which can be implemented by look-up tables (LUT) or bit-sliced based algorithms using bitwise operations such as AND, XOR. However, LUT usually requires high storage and is the operation leaking the most information, while bit-sliced S-boxes require only a limited number of logical operations to be evaluated, thus making bit-sliced S-boxes a popular choice for the design of lightweight algorithms [21].

ARX-based structures, either based on Feistel structure or SPN, are becoming popular, which use modular Addition to provide nonlinearity and Rotations/XOR to provide diffusion and thus get the name. The Addition, Rotation, and XOR operations are quite cheap in implementation, making ARX-based ciphers be among the best performers for micro-controllers. Some ARX-based LWC primitives are like HIGHT, XTEA, LEA.

Below we show an example of lightweight block cipher, PRESENT, which is one of the most popular lightweight cryptographic candidates.

PRESENT is a lightweight block cipher proposed in 2007 and included as an ISO/IEC standard in 2012 [47]. It is one of the most promising lightweight block ciphers for a replacement of AES in a resource-constrained environment.

PRESENT is based on SPN and consists of 31 rounds. The block length is 64 bits and the key size can be 80-bit or 128-bit. The substitution layer is 16 parallel application of a 4-bit S-box which was designed taking the implementation aspect into consideration, while the permutation layer is a regular bit-permutation from 64 bits to 64 bits with high hardware efficiency. The original design document [47] implement PRESENT-80 (80-bit key) in VHDL and synthesize it for the virtual silicon standard cell library, which occupies 1570 GEs and gives 200 Kbps throughput at 100 KHz.

4.2.2 *Stream ciphers*

Stream ciphers relatively occupy larger implementation area than block ciphers owing to the common rule that the internal state size of a stream cipher should be at least twice the security parameter to resist against the time-memory-data tradeoff attack [58]. Owing to the simple implementation and fast speed, stream ciphers also play an important role in LWC.

The LFSR, either based on bits or words, is the most commonly used component for stream ciphers, serving as a good source of randomness. Nonlinear components, e.g., S-boxes and boolean functions, are then used to disrupt the linearity of LFSR. Many new design ideas for stream ciphers appeared since the eSTREAM project, like the nonlinear FSR (NFSR)-based and permutation/sponge-based stream ciphers. Most lightweight stream ciphers are bit-based. Below we show such an example, Grain.

Grain is a stream cipher which was submitted to eSTREAM in 2004 and included into the final eSTREAM portfolio. It is designed primarily for restricted hardware environments. It consists of two feedback shift registers, one linear (LFSR) and one nonlinear (NFSR), and a boolean function taking bits from both the LFSR and NFSR to give the output. Recently, the AEAD version of Grain, Grain-128AEAD, was proposed and submitted to the NIST LWC project [59]. It is the only AEAD proposal based on pure stream cipher which survived to the second round. Ref. [60] gives hardware implementations of Grain-128AEAD, which can achieve 2.32 Gbps under 2.32 GHz and occupies 2695 GEs under 65 nm library.

4.2.3 *Hash functions*

Modern hash functions usually require large amounts of memory to store both their internal states and the blocks they are operating on. This significantly hinders the performance of a hash function in an IoT device where 8-bit processors and controllers are usually used and the memory capacity is just a few kilobytes. There are some studies targeted on lightweight hash algorithms and as mentioned, PHOTON [51], SPONGENT [52] and Lesamanta-LW [53] are included in the ISO/IEC standards for lightweight hashing methods. These hashing methods usually have significant reduction of the internal states, e.g., 256 characters' input, and require much smaller memory footprints.

4.2.4 *Asymmetric primitives*

Asymmetric cryptography requires a significantly larger implementation area (at least 10000 additional GEs) than symmetric cryptography under a same security level [61], which makes it less popular than the latter for LWC. However, asymmetric cryptographic primitives are essential in some cases, e.g., key exchange, authentication. Till now, ISO/IEC 29192-4 has included elliptic light (ELLI), cryptoGPS and ALIKE as the lightweight asymmetric cryptographic primitives [19]. Some lattice-based and code-based post-quantum cryptographic algorithms will also be potential options, treated in Section 5.

5 Lightweight cryptographic algorithms in the post-quantum scenario

The new use case of massive IoT requires 5G systems to identify and communicate with one million devices per square-kilometer. These devices will typically have very limited computational resources or very small hardware footprints. They may be small sensors that additionally have a strong requirement of very low energy consumption. How a 5G system will handle security challenges in this scenario does not seem to be developed in more detail at this moment. In addition to this scenario, we also have the recent development in quantum computation, that basically renders all public key algorithms based on the factoring and discrete logarithm problems. Future solutions must rely on problems that are not known to be solved in polynomial time using a large quantum computer. In this section we mention some of the problems and corresponding possible solutions.

Lightweight authentication protocols for ultra-constrained devices. In the most extreme case, constrained devices may be close to the case of RFID tags, for which the algorithms are hardware implemented and the number of gates is strongly limited. The problem of device authentication has been studied a lot for RFID tags. We assume that the base station and the device are sharing a common key value. The task is for the base station to authenticate the device through a procedure that is as simple as possible to implement in hardware. Existing recent such authentication protocols are often based on the hardness of the learning parity with noise (LPN) problem, a problem not known to be efficiently solved by a quantum computer.

Research on these types of protocols was initiated by the design of the protocols HB and HB$^+$ [22,23], which then became the prototypes for many protocols that base their design on the LPN problem or possibly variants of it.

The HB protocol [22] was designed to be very simple. The verifier (base station) and the prover (device) share a secret bitstring $\boldsymbol{x}$ of length l. The protocol consists of several rounds, all the same. At the beginning of round i, the verifier selects a random challenge $\boldsymbol{a}_i$, being a bitstring of length l, and sends it to the prover, who replies with $z_i = \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle + e_i$, where $\langle\rangle$ denotes the inner product operation and e_i is a biased random noise bit satisfying $P(e_i = 0) = \nu$ for a fixed probability $\nu < 1/2$. Then, the verifier checks whether the received bit z_i is equal to $\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle$. If this is the case, the response is called correct and otherwise incorrect. The HB protocol is secure against passive attacks assuming that the LPN problem with the parameters l and ν cannot be efficiently solved.

The HB$^+$ protocol [23] was later proposed to resist active attacks. As an extended protocol, the prover and the verifier share an additional length l binary secret $\boldsymbol{y}$. At the beginning of round i, the prover generates a length l random bit string $\boldsymbol{b}_i$ (a so-called blinding factor) and sends it to the verifier.

Afterward, as in the HB protocol, the verifier generates a challenge $\boldsymbol{a}_i$ and sends it to the prover. Then, the prover computes $z_i = \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle + \langle \boldsymbol{b}_i, \boldsymbol{y} \rangle + e_i$ and sends it to the verifier for verification. It was shown in [62] that security could be optimized if the length of $\boldsymbol{y}$ would be larger than the length of $\boldsymbol{x}$.

The protocols run in r rounds. For each round the verifier is getting more confident. In the end, authentication is considered successful if the number of incorrect answers is less than $p \cdot r$, for a suitably selected p. Otherwise, the device is rejected. If the noise probability ν is chosen close to 0.5 then a huge number of rounds is required to achieve reliability. If ν is small, the corresponding LPN problem gets easier to solve. Typical choices for the noise probability are $\nu = 0.25$ or $\nu = 0.125$. For 80-bit security, the protocols will then require length 512 for the secret bitstring (and $512 + 80$ bits for HB$^+$) [62].

The proofs of security for HB and HB$^+$ were simplified and extended to the parallel versions in [63,64], which means that several rounds can be performed at the same time. A large number of generalizations of the HB protocols have been given and we refer to [26] for more details on all these extended protocols and their applicability in practice. We mention the HB^{++} protocol [24] and the Lapin protocol [25], where the latter reduces communication by relating to the ring-LPN problem.

Although the protocols may be efficiently implemented, some additional costs will be associated with such a solution. The device will have to store a copy of the shared key in some trusted way in memory. More importantly, the protocols require the use of randomness, so a random number generator needs to be implemented. It is sometimes argued that randomness can be generated from physical properties like thermal noise, oscillation jitter, or other methods. It can however be difficult to ensure a sufficient level of entropy in these methods and breakdowns for individual devices may happen. The option can then be to use pseudorandom number generators (PRNGs) implemented through, say a stream cipher like Grain or the block cipher PRESENT in a CTR mode. But then such a symmetric primitive can be used in a straightforward manner to realize device authentication without the need for any random numbers at all [65].

Lightweight public-key crypto algorithms for constrained devices. The main advantage of public key crypto schemes over any symmetric scheme is that the public key retrieval and validation can be performed using a certificate-based public key infrastructure (PKI) on untrusted channels. This is common in internet protocols such as transport layer security (TLS). Keys that have been exposed can additionally be revoked if the system keeps certificate revocation lists (CRLs). Public key solutions have not been used much in current and legacy standards for mobile communication. However, the advantages of public key crypto might change this fact.

Recently, NIST has initiated an effort to standardize post-quantum cryptography (PQC) [66]. The algorithms should be general-purpose algorithms and the project covers both signature, encryption, and key exchange/encapsulation mechanisms (KEMs). The most common class of algorithms is those based on lattice problems, often using the learning with errors (LWE) problem, or some related problems. These algorithms are described for a security level of 128 bits and upwards, and are not lightweight in general. However, some of them might be scaled down to a somewhat lightweight form. We mention the encryption/KEM scheme LAC, that is perhaps the simplest of the remaining proposals [67].

An attempt to build lightweight schemes based on LWE was done in [28]. It uses a Ring-LWE public key encryption scheme and shows how the techniques of ciphertext compression and error correcting codes can be used to reduce complexity. The experimental public key encryption and authentication system was implemented on an 8-bit AVR target, which significantly outperformed elliptic curve and RSA-based proposals.

Another interesting concept for this use case might be identity-based encryption (IBE), in which parties and devices use their personal identifiers, such as user names or serial numbers, as the public keys. Thus, the key distribution and management become simplified in the multi-user scenarios, which is very desirable for ad-hoc networks in the emerging IoT. In [68], the authors show the practical use of IBE constructed from the Ring-LWE assumption, and they provide evidence that it can be implemented on embedded platforms like ARM Cortex-M0 and ARM Cortex-M4.

6 Conclusion

In this survey we have discussed various cryptographic algorithms that are currently included or could potentially be included in the 5G standard. We have also discussed a number of techniques and primitives that may be relevant in the new 5G use cases, like massive IoT. If public key crypto primitives will be used, they will most likely have to rely on lattice-based problems (or code-based problems) [69]. We briefly described a few such attempts for lightweight applications.

Acknowledgements This work was in part financially supported by Swedish Foundation for Strategic Research (Grant No. RIT17-0005), and ELLIIT Research Program. The author Jing YANG is also supported by the scholarship from National Digital Switching System Engineering and Technological Research Center, China. We would like to thank all anonymous reviewers for providing valuable comments to the manuscript.

References

1 ITU-R. IMT Vision-Framework and overall objectives of the future development of IMT for 2020 and beyond. Recommendation ITU-R M.2083-0. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf
2 ITU-R. Minimum requirements related to technical performance for IMT-2020 radio interface. Report ITU-R M.2410-0. https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf
3 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security Architecture (Release 16). 3GPP TS 33.401 (V16.1.0). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296
4 Gabriel B. Cloud RAN & the Next-Generation Mobile Network Architecture. White Paper. https://www-file.huawei.com/-/media/CORPORATE/PDF/mbb/cloud-ran-the-next-generation-mobile-network-architecture.pdf?la=en
5 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; System Architecture for the 5G System (5GS); Stage 2 (Release 16). 3GPP TS 23.501 (V16.3.0). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144
6 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) (Release 16). 3GPP TS 33.220 (V16.0.0). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2280
7 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; Security Architecture and Procedures for 5G system (Release 16). 3GPP TS 33.501 (V16.1.0). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169
8 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). 3GPP TS 33.841 (V16.1.0). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3422
9 3rd Generation Partnership Project. Technical Specification Group Core Network and Terminals; Numbering, Addressing and Identification (Release 16). 3GPP TS 23.003 (V16.1.0). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=729
10 Huawei. Partnering with the Industry for 5G Security Assurance. White Paper, 2019. https://www-file.huawei.com/-/media/corporate/pdf/trust-center/huawei-5g-security-white-paper-4th.pdf
11 Yang J, Thomas J, Alexander M. Vectorized linear approximations for attacks on SNOW 3G. In: Proceedings of the 27th Annual Fast Software Encryption Conference, 2020
12 Yang J, Thomas J, Alexander M. Spectral analysis of ZUC-256. In: Proceedings of the 27th Annual Fast Software Encryption Conference, 2020
13 Patrik E, Thomas J, Maximov A, et al. A new SNOW stream cipher called SNOW-V. IACR Trans Symmetric Cryptol, 2019, 20: 1–42
14 ZUC Design Team. The ZUC-256 Stream Cipher. 2018. http://www.is.cas.cn/ztzl2016/zouchongzhi/201801/W020180126529970733243.pdf
15 NIST. Announcing the advanced encryption standard (AES). Federal Information Processing Standards Publication 197. https://www.cisco.com/c/dam/en/us/products/collateral/security/anyconnect-secure-mobility-client/fips.pdf

16 International Organization for Standardization. Information Technology - Security Techniques - Lightweight Cryptography - Part 2: Block Ciphers. ISO/IEC 29192-2:2012. https://www.iso.org/standard/56552.html
17 International Organization for Standardization. Information Technology - Security Techniques - Lightweight Cryptography - Part 3: Stream Ciphers. ISO/IEC 29192-3:2012. https://www.iso.org/standard/56426.html
18 International Organization for Standardization. Information Technology - Security Techniques - Lightweight Cryptography - Part 5: Hash-functions. ISO/IEC 29192-5:2016. https://www.iso.org/standard/67173.html
19 International Organization for Standardization. Information Technology - Security Techniques - Lightweight Cryptography - Part 4: Mechanisms Using Asymmetric Techniques. ISO/IEC 29192-4:2013. https://www.iso.org/standard/56427.html
20 CRYPTREC Lightweight Cryptography Working Group. CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography). 2017. https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf
21 Alex B, Léo P. State of the art in lightweight symmetric cryptography. IACR Cryptology ePrint Archive, 2017. https://www.semanticscholar.org/paper/State-of-the-Art-in-Lightweight-Symmetric-Biryukov-Perrin/532441547d905feae7a65f635594585c96d2987b
22 Nicholas J H, Manuel B. Secure human identification protocols. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2001. 52–66
23 Ari J, Stephen A W. Authenticating pervasive devices with human protocols. In: Proceedings of Annual International Cryptology Conference. Berlin: Springer, 2005. 293–308
24 Julien B, Herv C, Emmanuelle D. HB++: a lightweight authentication protocol secure against some attacks. In: Proceedings of the 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2006. 28–33
25 Stefan H, Eike K, Vadim L, et al. Lapin: an efficient authentication protocol based on ring-LPN. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 2012. 346–365
26 Frederik A, Matthias H, Vasily M. Lightweight authentication protocols on ultra-constrained RFIDs - myths and facts. In: Radio Frequency Identification: Security and Privacy Issues. Cham: Springer, 2015. 1–18
27 Buchanan W J, Li S C, Asif R. Lightweight cryptography methods. J Cyber Secur Tech, 2017, 1: 187–201
28 Markku-Juhani O S. Ring-LWE ciphertext compression and error correction: tools for lightweight post-quantum cryptography. In: Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, 2017. 15–22
29 SECG. SEC 1: Recommended Elliptic Curve Cryptography (Version 2.0). 2009. http://www.secg.org/sec1-v2.pdf
30 SECG. SEC 2: Recommended Elliptic Curve Domain Parameters (Version 2.0). 2010. http://www.secg.org/sec2-v2.pdf
31 Adam L, Mike M, Sean T. Elliptic curves for security. IETF RFC 7748, 2016. https://www.rfc-editor.org/info/rfc7748
32 Hugo K, Mihir B, Ran C. HMAC: keyed-hashing for message authentication. IETF RFC 2104, 1997. https://www.rfc-editor.org/rfc/pdfrfc/rfc2104.txt.pdf
33 International Organization for Standardization. Information Technology -Security Techniques - Hash-Functions - Part 3: Dedicated Hash-Functions. ISO/IEC 10118-3:2004. https://www.iso.org/standard/39876.html
34 ETSI SAGE. Observations and questions on 256-bit security goals. S3-200929. https://www.3gpp.org/FTP/tsg_sa/WG3_Security/TSGS3_99e/Docs
35 ETSI SAGE. Expectations and requirements for 256-bit algorithms. S3-190107. https://www.3gpp.org/DynaReport/TDocExMtg--S3-94--33863.htm
36 ETSI SAGE. 256-bit algorithm candidates. S3-194534. https://www.3gpp.org/FTP/Meetings_3GPP_SYNC/SA3/Docs
37 Rei U, Sumio M, Naofumi H, et al. A high throughput/gate aes hardware architecture by compressing encryption and decryption datapaths toward efficient cbc-mode implementation. Cryptology ePrint Archive, 2016. https://eprint.iacr.org/2016/595
38 Andrey B, Dmitry K, Christian R. Biclique cryptanalysis of the full AES. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2011
39 Gupta S S, Chattopadhyay A, Khalid A. Designing integrated accelerator for stream ciphers with structural similarities. Cryptogr Commun, 2013, 5: 19–47
40 Roberto A, Billy B B. Faster 128-EEA3 and 128-EIA3 software. In: Proceedings of the 16th International Conference on Information Security, Cham: Springer, 2015. 199–208
41 Liu Z B, Zhang Q L, Ma C Q, et al. HPAZ: a high-throughput pipeline architecture of ZUC in hardware. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016. 269–272
42 ETSI/SAGE. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2, Document 2: SNOW 3G Specification (version 1.1). 2006
43 Carlos C, Matthew D, Sean M. A security evaluation of the SNOW-V stream cipher. Private Correspondence, 2020
44 Jiao L, Li Y Q, Hao Y L. A guess-and-determine attack on SNOW-V stream cipher. Comput J, 2020. doi: 10.1093/comjnl/bxaa003
45 ETSI/SAGE. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3, Document 2: ZUC Specification. 2011. https://www.gsma.com/aboutus/wp-content/uploads/2014/12/eea3eia3zucv16.pdf
46 Guang G. Securing Internet-of-Things. In: Proceedings of International Symposium on Foundations and Practice of Security. Berlin: Springer, 2018. 3–16
47 Andrey B, Lars R K, Gregor L, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of International

Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2007. 450–466
48 Shirai T, Kyoji S, Toru A, et al. The 128-bit blockcipher CLEFIA. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 2007. 181–195
49 Christophe D C. Trivium: a stream cipher construction inspired by block cipher design principles. In: Proceedings of International Conference on Information Security. Berlin: Springer, 2006
50 Dai W, Kota I, Jun K, et al. Enocoro-80: a hardware oriented stream cipher. In: Proceedings of the 3rd International Conference on Availability, Reliability and Security, Barcelona, 2008. 1294–1300
51 Guo J, Peyrin T, Poschmann A. The PHOTON family of lightweight hash functions family. In: Proceedings of Advances in Cryptology-Crypto. Berlin: Springer, 2011. 222–239
52 Bogdanov A, Knežević M, Leander G, et al. SPONGENT: the design space of lightweight cryptographic hashing. IEEE Trans Comput, 2012, 62: 2041–2053
53 Hirose S, Ideguchi K, Kuwakado H, et al. A lightweight 256-bit hash function for hardware and low-end devices: Lesamnta-LW. In: Proceedings of International Conference on Information Security and Cryptology. Berlin: Springer, 2011. 151–168
54 Ågren M, Hell M, Johansson T, et al. Grain-128a: a new version of Grain-128 with optional authentication. Int J Wirel Mobile Comput, 2011, 5: 48–59
55 Mitsuru M. New block encryption algorithm MISTY. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 1997
56 Deukjo H, Jaechul S, Seokhie H, et al. HIGHT: a new block cipher suitable for low-resource device. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2006
57 María N P. Lightweight cryptography. In: Proceedings of Summer School on Real-world Crypto and Privacy, Sibenik, 2018
58 Frederik A, Vasily M. On lightweight stream ciphers with shorter internal states. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer, 2015
59 Martin H, Thomas J, Meier W, et al. Grain-128AEAD-A lightweight AEAD stream cipher. NIST Lightweight Cryptography project, Round 2. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates
60 Jonathan S, Martin H, Mattias S, et al. Efficient hardware implementations of Grain-128AEAD. In: Proceedings of International Conference on Cryptology in India. Cham: Springer, 2019. 495–513
61 Eisenbarth T, Kumar S, Paar C, et al. A survey of lightweight-cryptography implementations. IEEE Design Test Comput, 2007, 24: 522-533
62 Éric L, Pirre-Alain F. An improved LPN algorithm. In: Proceedings of International Conference on Security and Cryptography for Networks. Berlin: Springer, 2006. 348–359
63 Jonathan K, Ji S S. Parallel and concurrent security of the HB and HB+ protocols. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2006. 73–87
64 Katz J, Shin J S, Smith A. Parallel and concurrent security of the HB and HB+ protocols. J Cryptol, 2010, 23: 402–421
65 Martin F, Sandra D, Johannes W. Strong authentication for RFID systems using the AES algorithm. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2004. 357–370
66 Chen L, Stephen J, Yi-Kai L, et al. Report on post-quantum cryptography (NISTIR 8105). 2016. https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf
67 Lu X H, Liu Y M, Jia D D, et al. LAC. 2019. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions
68 Tim G, Tobias O. Towards lightweight identity-based encryption for the post-quantum-secure Internet of Things. In: Proceedings of 2017 18th International Symposium on Quality Electronic Design (ISQED), 2017. 319–324
69 NIST. Post-Quantum Cryptography Standardization Project. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization