Evolution of 3GPP over-the-air security

May 10, 2018 • Guillaume Delugré

I have written this page to have a centralized view of the radio interface security inside 3GPP technologies, from 2G to 5G. Getting a clear view of what is going on can be confusing and discouraging as it often requires to browse through dozens of 3GPP documents at the same time.

The most important security functions are listed for each technology: authentication, confidentiality and integrity. I have tried to keep this as short as possible while keeping the most revelant information, such as which radio layers are involved and links to the specifications.

GSM

In this section, `A3` and `A8` are derivation functions implemented as part of the COMP128 algorithm. `A5` functions are the encryption algorithms.

Authentication

The BTS authenticates the MS using a challenge-response scheme relying on a 128-bit shared secret `Ki` stored in the SIM card and the core network.

1. MS ⟵ BTS : *Authentication Request* (128-bit `RAND`)
2. MS ⟶ BTS : *Authentication Response* (32-bit `SRES = A3(Ki, RAND)`)

3. MS ⟵ BTS : *Authentication Reject* if `SRES` is incorrect

Encryption

Traffic encryption is initiated by the BTS after authentication by sending a *Ciphering Mode Command* to the MS. The message contains the encryption algorithm to use. The dedicated channels `TCH` and `DCCH` are then encrypted at the physical layer. The encryption 64-bit key `Kc` is `A8(Ki, RAND)`.

- `A5/0` : no encryption
- `A5/1` : LFSR-based stream cipher, 64-bit key, broken
- `A5/2` : LFSR-based stream cipher, 64-bit key, broken, **prohibited use**
- `A5/3` : KASUMI in OFB mode, 64-bit key extended to 128 bits (3GPP TS 55.216)

In the event of a handover from a UTRAN, the mobile can use its UMTS security context to switch to a 128-bit algorithm. The 128-bit encryption key K_{128} is `HMAC-SHA256(CK || IK, "\x32")` truncated to 128 bits (3GPP TS 33.102, 3GPP TS 33.220).

- `A5/4` : same as `A5/3`, 128-bit key (3GPP TS 55.226)

Integrity

Traffic is not integrity protected in GSM.

GPRS

Authentication

GPRS use a challenge-response authentication scheme similar to GSM.

1. MS ⟵ BTS : *Authentication and Ciphering Request* (`RAND` , algorithm)
2. MS ⟶ BTS : *Authentication and Ciphering Response* (`SRES = A3(Ki, RAND)`)
3. MS ⟵ BTS : *Authentication and Ciphering Reject* if `SRES` is incorrect

Encryption

The traffic is encrypted at the LLC layer. The 64-bit encryption key `GPRS-Kc` is `A8(Ki, RAND)`. Contrary to GSM, the traffic is asymmetrically encrypted by using a `direction` bit. The sequence number of the LLC packet is also used in the computation. Some GPRS encryption algorithms are not publicly documented.

- `GEA0` : no encryption
- `GEA1` : undocumented, LFSR-based stream cipher, 64-bit key, broken, **prohibited use**
- `GEA2` : undocumented, LFSR-based stream cipher, 64-bit key
- `GEA3` : KASUMI in OFB mode, 64-bit key extended to 128 bits, similar to `A5/3` (3GPP TS 55.216)

As with GSM, the device can use a 128-bit algorithm if it already has a UMTS security context. The 128-bit encryption key K_{128} is `HMAC-SHA256(CK || IK, "\x32")` truncated to 128 bits.

- `GEA4` : same as `GEA3` , 128-bit key (3GPP TS 55.226)

Integrity

Traffic is not integrity protected in GPRS.

UMTS

UMTS uses a set of function *f1* to *f9* for security purposes. Derivation functions *f1* to *f5* are not standardized. The specifications provide two example algorithms sets: MILENAGE based on AES (3GPP TS 35.206), and TUAK based on SHA3 (3GPP TS 35.231).

Authentication

UMTS uses a mutual authentication between the mobile and the base station. It relies on a 128 or 256-bit shared secret key `K` stored in the USIM and the core network. The mobile and the network keep track of a 48-bit sequence number `SQN` to prevent replay attacks. The authentication scheme also makes use of a 16-bit `AMF` value that is operator dependent.

1. Network generates a 128-bit `RAND` and computes the following values:
   - 48-bit `AK = f5(K, RAND)`
   - `XRES = f2(K, RAND)`
   - `MAC = f1(K, SQN || RAND || AMF)`
2. MS ⟵ NodeB : *Authentication Request* (128-bit `RAND` , `AUTN = SQN ⊕ AK || AMF || MAC`)
3. MS verifies `MAC` and aborts by sending *Authentication Failure* if it is incorrect
4. MS computes `AK` , verifies `SQN` , and aborts by sending *Synchronization Failure* if it is incorrect
5. MS ⟶ NodeB : *Authentication Response* (`RES = f2(K, RAND)`)
6. MS ⟵ NodeB : *Authentication Failure* if `RES` and `XRES` do not match

Encryption

Ciphering is initiated by the network by sending a RRC *Security Mode Command* through DCCH.

The traffic is encrypted at the RLC layer, or MAC layer in case of bearers in transparent mode. As with GPRS, a direction bit is used in the computation as well as a counter from the RLC or MAC header. Moreover, the traffic of each radio bearer is encrypted separately by using a 4-bit bearer id. The 128-bit encryption key `CK` is `f3(K, RAND)` . Encryption algorithms are referred to as *f8*.

- `UEA0` : no encryption
- `UEA1` : KASUMI in OFB mode, 128-bit key, similar to `A5/3` (3GPP TS 35.201)
- `UEA2` : SNOW 3G, 128-bit key (3GPP TS 35.216)

Integrity

Integrity is initiated by the network by sending a RRC *Security Mode Command* through DCCH.

The traffic is integrity protected for non-access stratum at the RRC layer. The computation involves a direction bit, the sequence number of the RRC frame, and a 32-bit nonce value sent by the network in the *Security Mode Command* message. The 128-bit integrity key `IK` is `f4(K, RAND)` . Integrity algorithms are referred to as *f9*.

- `UIA0` : no integrity
- `UIA1` : 32-bit MAC, KASUMI in CBC-MAC mode (3GPP TS 35.201)
- `UIA2` : 32-bit MAC, based on SNOW 3G (3GPP TS 35.216)

LTE

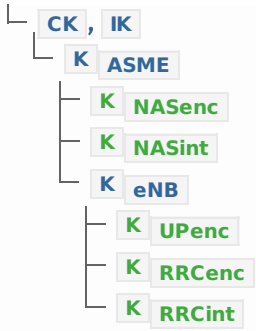
Authentication

LTE uses the same mutual authentication scheme as UMTS.

Whereas in UMTS, the resulting keys `CK` and `IK` are used to protect traffic, in LTE they are used to derive a tree of keys. Two intermediary 256 bit keys are derived:

- K_{ASME} : derived from `CK` , `IK` , as well `SQN` , `AK` (from the `AUTN` token) and the SN id (serving network identity)
- K_{eNB} : derived from K_{ASME} and the counter of uplink NAS messages

Those keys are then further derived into a set of confidentiality and integrity keys. The final tree hierarchy after authentication is:



The derivation functions are based on `HMAC-SHA256` and are described in 3GPP TS 33.401.

Encryption

Ciphering is initiated by the network by sending RRC and NAS *Security Mode Command*.

Traffic is encrypted at the PDCP layer. Three different 128-bit keys are used depending whether on the nature of the traffic:

- K_{NASenc} for *Non-Access Stratum* messages
- K_{RRCenc} for *Access Stratum* messages
- K_{UPenc} for *User Plane* messages

K_{NASenc} is derived from K_{ASME} while K_{RRCenc} and K_{UPenc} are derived from K_{eNB}.

The computation involves a direction bit, a direction dependent 32-bit PDCP counter and a 5-bit bearer id.

- `EEA0` : no encryption
- `128-EEA1` : same as `UEA2` (3GPP TS 33.401)
- `128-EEA2` : AES in CTR mode, 128-bit key (3GPP TS 33.401)
- `128-EEA3` : ZUC, 128-bit key (3GPP TS 35.222)

Integrity

Integrity is initiated by the network by sending RRC and NAS *Security Mode Command*.

Traffic is integrity protected at the PDCP layer. Control plane traffic must be protected while user plane traffic must not. Two different 128-bit keys are used depending whether on the nature of the traffic:

- K_{NASint} for *Non-Access Stratum* messages derived from K_{ASME}
- K_{RRCint} for *Access Stratum* messages derived from K_{eNB}

A key K_{UPint} to protect user traffic is also computed by the eNodeB, but is only used between an eNodeB and a relay node.

The computation involves a direction bit, a direction dependent 32-bit PDCP counter and a 5-bit bearer id.

- `EIA0` : no integrity, only for emergency calls
- `128-EIA1` : similar to `UIA2` (3GPP TS 33.401)
- `128-EIA2` : 32-bit MAC, 128-bit AES in CMAC mode (3GPP TS 33.401)
- `128-EIA3` : 32-bit MAC, based on ZUC (3GPP TS 35.222)

LTE D2D ProSe (*Device to Device Proximity Services*)

ProSe is a 3GPP technology that appeared during Release 12. It allows LTE user equipments to discover themselves in a geographic area and communicate with one another through a direct communication channel.

It is principally meant to be a competitor of TETRA for Public Safety, but it can also operate on commercial bands, allowing for other potential usages such as Vehicle-to-Vehicle communication.

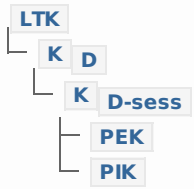
I am not aware of any real-life uses of ProSe but I am including it for the sake of completeness. Its security aspects are defined in 3GPP TS 33.303.

Authentication

UEs are provisioned with a long term pre-shared key. For communicating with another UE, a 256-bit key K_D is negociated. This key is then stored in the UE and can possibly be reused or refreshed at a later point.

Everytime two UEs establish a communication channel, a new 256-bit $K_{D\text{-}sess}$ is derived from K_D and two nonces exchanged between the UEs.

The key derivation functions are based on `HMAC-SHA256`. The key hierarchy upon communication establishment is:



K_D can be negociated between the two UEs in two ways:

1. by separately interacting with a key management server (PKMF) located in the network core.
2. by direct communication with each other. The standard mentions the use of `ECCSI` (*Elliptic Curse-based Certificateless Signatures for Identity-based Encryption*) and `SAKKE` (*Sakai Kasahara Key Encryption*), both defined in RFC 6507 and RFC 6508.

Once two UEs need to communicate with each other, they establish a secure communication channel:

1. $UE_1 \longrightarrow UE_2$: *Direct Communication Request* (`LTK` id, K_D id, algorithms, `Nonce_1`)
2. $UE_1 \longleftrightarrow UE_2$: Authentication and K_D negociation (optional)
3. UE_2 generates `Nonce_2` and computes $K_{D\text{-}sess}$ `= KDF(`K_D`, Nonce_1, Nonce_2)`, as well as `PIK` and `PEK`
4. $UE_1 \longleftarrow UE_2$: *Direct Security Mode Command* (`Nonce_2` , chosen algorithms, `MAC-I`)
5. UE_1 computes $K_{D\text{-}sess}$, `PIK` , `PEK` and verifies `MAC-I` with `PIK`
6. $UE_1 \longrightarrow UE_2$: *Direct Security Mode Complete* (`MAC-I`)
7. UE_2 verifies `MAC-I` with `PIK` . Both UEs then have a synchronized security context.

Encryption

The traffic is encrypted at PDCP layer, with the same algorithms as for LTE. The 128-bit encryption key is `PEK` , derived from $K_{D\text{-}sess}$.

Integrity

The traffic is integrity protected at the PDCP layer, with the same algorithms as for LTE. The 128-bit integrity key is `PIK` , derived from $K_{D\text{-}sess}$.

EC-GSM-IoT (*Extended Coverage GSM for IoT*)

The security of EC-GSM-IoT is described in 3GPP TS 43.020, starting from Release 13.

Authentication

The scheme of EC-GSM-IoT is an integrity-protected version of the authentication and key

agreement (AKA) of UMTS.

1. MS ⟵ BTS : *Authentication and Ciphering Request* (`RAND` , `AUTN` , encryption and integrity algorithms, `MAC-GMM`)
2. MS performs UMTS AKA, derives integrity key Ki_{128} and verifies `MAC-GMM`
3. MS ⟶ BTS : *Authentication and Ciphering Response* (`SRES` , `MAC-GMM`)
4. Network verifies `MAC-GMM` and verifies `SRES` as for UMTS

Encryption

Traffic is encrypted at the LLC layer. The 128-bit encryption key Kc_{128} is `HMAC-SHA256(CK || IK, "\x32")` truncated to 128 bits.

- `GEA0` : no encryption
- `GEA4` : same as `GEA3` , 128-bit key (3GPP TS 55.226)
- `GEA5` : undocumented, based on SNOW 3G, 128-bit key (3GPP TS 55.251)

Integrity

Traffic integrity is mandatory for the control plane. Messages are integrity protected at the GMM or LLC layers. The 128-bit integrity key Ki_{128} is `HMAC-SHA256(CK || IK, "\x38")` truncated to 128 bits.

- `GIA4` : undocumented, 32-bit MAC, based on KASUMI in CBC-MAC mode (3GPP TS 55.241)
- `GIA5` : undocumented, 32-bit MAC, based on SNOW 3G (3GPP TS 55.251)

5G-NR

The information in this section may be subject to change in the future. The security architecture of 5G is described in 3GPP TS 33.501.

Authentication

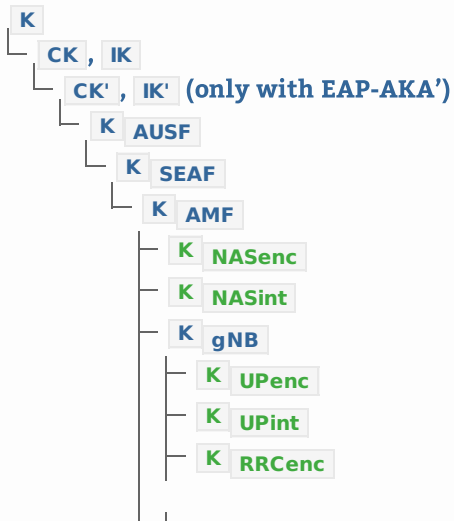
5G-NR supports two authentication schemes: EAP-AKA' and 5G AKA.

EAP-AKA' is described in RFC 5448.

5G AKA is a hardened version of the UMTS authentication scheme. It goes as follows:

1. MS ⟵ gNB : *Authentication Request* (`RAND` , `AUTN`)
2. MS performs UMTS AKA and computes `RES`
3. MS derives `RES` into `SRES* = KDF(CK || IK, RES, RAND)`
4. MS ⟶ gNB : *Authentication Response* (`RES*`)
5. MS ⟵ gNB : *Authentication Failure* if `RES*` and `XRES*` do not match

Once authentication is performed, a hierarchy of tree is derived for the various network components, until generating the keys used for encryption and integrity.



K_{N3IWF} is an IKEv2 key that can be used to connect to the network core through a non-3GPP connection.

The derivation functions involved (such as the one used for derivating $SRES^*$) are based on HMAC-SHA256 and are described in 3GPP TS 33.501.

Encryption

The traffic is encrypted at the PDCP layer. Similarly to LTE, a set of three different 128-bit encryption keys are used for user plane, NAS and AS: K_{UPenc}, K_{NASenc}, K_{RRCenc}. Encryption algorithms are the same as for LTE.

- NEA0 : no encryption
- 128-NEA1 : identical to 128-EEA1 (SNOW 3G)
- 128-NEA2 : identical to 128-EEA2 (AES-128 CTR)
- 128-NEA3 : identical to 128-EEA3 (ZUC)

Integrity

The traffic is integrity protected at the PDCP layer. As for LTE, AS and NAS are integrity protected using K_{RRCint} and K_{NASint}. However, 5G-NR also allows to optionally protect user plane using K_{UPint}. Integrity algorithms are the same as for LTE.

- NIA0 : no integrity
- 128-NIA1 : identical to 128-EIA1 (based on SNOW 3G)
- 128-NIA2 : identical to 128-EIA2 (AES-128 CMAC)
- 128-NIA3 : identical to 128-EIA3 (based on ZUC)

Comments