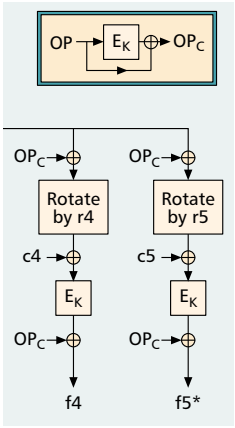
AN INTRODUCTION TO ACCESS SECURITY IN UMTS

GEIR M. KØIEN, TELENOR R&D AND AGDER UNIVERSITY COLLEGE



With the advent of 3G mobile systems a serious effort has been made to create a consistent security architecture based on the threats and risks a 3G system faces.

ABSTRACT

The first generation of cellular mobile communications systems contained few if any security measures to protect the system operator and users. The second generation generally did a lot better, and contained entity authentication and confidentiality protection. Although this was a major improvement, security protection in the second generation left a lot to be desired. With the advent of 3G mobile systems a serious effort has been made to create a consistent security architecture based on the threats and risks a 3G system faces.

PRINCIPLES AND OBJECTIVES FOR UMTS SECURITY

The Universal Mobile Telecommunications System (UMTS) is in many ways an evolution of the Global System for Mobile Communications (GSM). The basic access security mechanisms found in GSM were the starting point for UMTS access security. Of course, the design objectives for the UMTS security architecture were not limited to the existing security solutions in GSM. In fact, there is a separate specification for 3G security principles and objectives [1]. The main principles are:

- **UMTS security will build on the security of second-generation (2G) systems.** The requirement means that existing GSM security features that are needed and robust shall be kept.
- **UMTS security will improve on the security of 2G systems.** UMTS security will address and correct real and perceived weaknesses in 2G systems. This includes the introduction of mutual authentication and strong encryption with 128-bit key length.
- **UMTS security will offer new security features.** UMTS security must cater for new services in the 3G environment. This includes environments with multiple operators/providers interworking to offer new services.

In addition to this, a threat/risk analysis was carried out to define the threats and risks facing a 3G system. The threats and risks were assessed, and a number of security requirements were defined [2]. This was then used as the basis for defining the security features needed in the security architecture. Based on these security features, a set of security mechanisms was defined [3].

THE UMTS SECURITY ARCHITECTURE

THE SCOPE

The UMTS security architecture for access security is defined in the technical specification 3G TS 33.102, "Security Architecture" [3]. The main task of the UMTS security architecture can be summarized as:

- Authenticate the user equipment (UE), specifically the USIM.[1] This includes assuring the UE that it is connected to a valid network.
- Provide the UE and the serving network (SN) with session keys.
- Allow the UE and SN to establish connections protected with the session keys.

There are, of course, other aspects of the security architecture, but authentication, key generation, and encryption/integrity protection of the access link are the main features. We shall now take a closer look at the architecture and shall start off with the basis for authentication, namely an entity identification scheme.

IDENTITIES

A prerequisite for entity authentication is that the entities have well defined unique identities. The primary user identity is the International Mobile Subscriber Identity (IMSI) number (Fig. 1). Note that the IMSI number is not the subscriber number (the so-called MSISDN number). The MSISDN number (or numbers) is a telephone number with full international prefix and is associated with the IMSI number in the operative databases. The MSISDN numbers are (generally) public information, while the IMSI number is intended for system internal identification and routing purposes.

Identity presentation must necessarily precede identity verification (authentication); since it is the authentication procedure that produces the session keys used for encryption, we have a situation where the permanent identity IMSI will be visible on the over-the-air interface. This is undesirable since it allows for subscriber location tracking. To mitigate the problem the SN may issue a local temporary identity called the TMSI (4 bytes with hexadecimal coding) to be used for subsequent identification. The normal procedure is therefore that the UE presents itself with its IMSI the first time it enters a new service area — serving general packet radio service (GPRS) support node (SGSN) or visited location register (VLR). Then, after encryption has commenced,

[1] *The USIM is an application running on the smartcard (UICC). It is independent of the mobile station (MS).*

1536-1284/04/$20.00 © 2004 IEEE

the SN issues a TMSI number to the UE. The TMSI is issued in encrypted form and only used in noncorrelated plaintext form. It is therefore hard to track a particular subscriber since there is no apparent relationship between the IMSI and the TMSI.[2] The use of the TMSI thus provides a measure of identity/location confidentiality.

In addition to the IMSI identifying the USIM, one also has an identity for the mobile station (MS). This identity is called the International Mobile Station Equipment Identity (IMEI) and is a unique identity. The IMEI number will regularly be checked against a database called the Equipment Identity Register (EIR). A stolen handset, provided that the proper legal steps are taken, is registered on the "blacklist" in EIR. Subsequent use of the handset will then be barred by the operator; in addition, the operator may log the (IMSI,IMEI) tuple for use by law enforcement agencies.

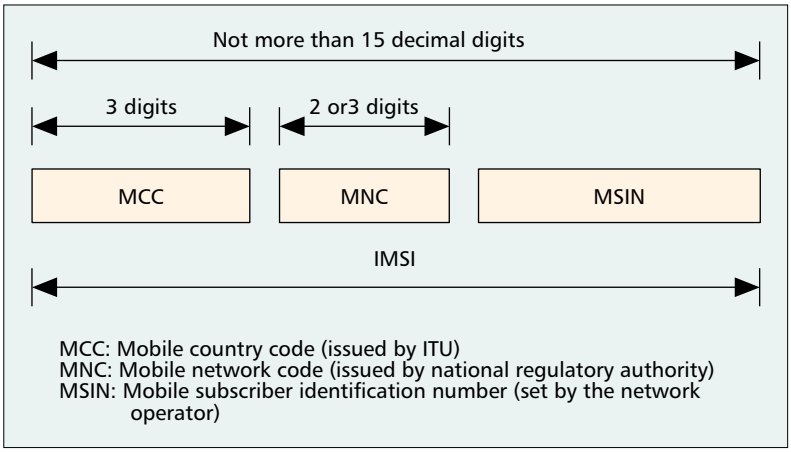


Figure 1. *Layout of the IMSI number.*

Identity Verification (Authentication) and Provision of Session Keys

During the connection setup phase the UE will identity itself by means of either the IMSI or TMSI. The claimed identity must then be verified by the network. To that end the network must execute the authentication procedure.

The security architecture is designed around a mutual authentication procedure that is executed between the user (USIM) and the SGSN/VLR[3] at the network end. The procedure is called UMTS Authentication and Key Agreement (AKA) since in addition to providing authentication services it also includes generation of session keys for confidentiality and integrity protection at the user end. The cryptographic algorithms/functions are defined in a requirements specification [4].

The AKA procedure is executed in two stages (Fig. 2). The first stage involves transfer of security credentials (authentication vector, AV) from the home environment (HE) to the serving network (SN). The HE mainly consists of the home subscription database (HLR) and authentication center (AuC); the SN consists of the parts of the core network that are directly involved in setting up connections. With respect to access security, the SN network elements of interest are the SGSN, which handles packet-switched traffic, and the circuit-switched GSM nodes VLR/MSC (mobile switching center). An operator with a physical access infrastructure will normally have both HE and SN nodes.

The authentication vectors contain sensitive data like *challenge-response* authentication data and cryptographic keys. It is therefore clear that the transfer of authentication vectors between the HLR/AuC and the SGSN/VLR needs to be secured against eavesdropping and modification (i.e., both the transfer's confidentiality and integrity must be protected). The actual transfer mechanism for the AVs is the SS7-based Mobile Application Part (MAP) protocol. The MAP protocol itself contains no security functionality, but a security extension to MAP called MAPsec

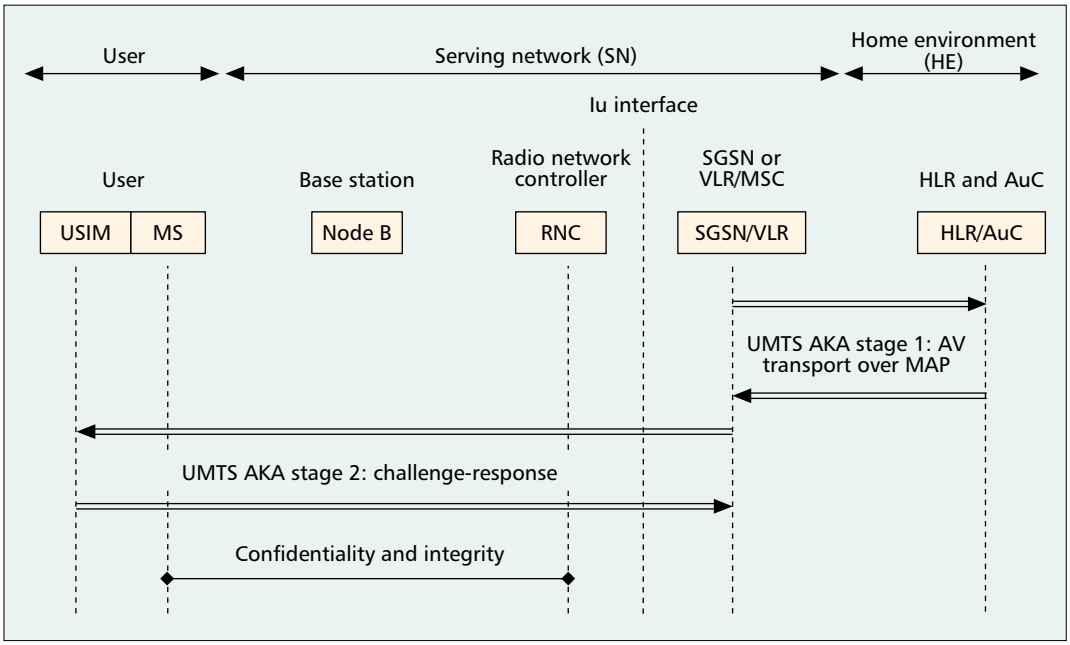


Figure 2. *Simplified UMTS architecture and coverage of the basic access security services.*

[2] *Tracking is still possible, but the adversary will not know who he/she is tracking.*

[3] *Either SGSN (packet switched) or VLR/MSC (circuit-switched).*

Much of the work with the UMTS access architecture has been focused on backward compatibility with GSM/GPRS. From a security point of view, backwards compatibility with a system with weaker security is very undesirable. However, commercial reality dictates that backwards compatibility be had.

Algorithm	Purpose/usage	O: Operator-specific S: Fully standardized	Location
f0	Random challenge generating function	O	AuC
f1	Network authentication function	O – (MILENAGE)	USIM and AuC
f1*	Resynchronization message authentication function	O – (MILENAGE)	—
f2	User challenge-response authentication function	O – (MILENAGE)	—
f3	Cipher key derivation function	O – (MILENAGE)	—
f4	Integrity key derivation function	O – (MILENAGE)	—
f5	Anonymity key derivation function for normal operation	O – (MILENAGE)	—
f5*	Anonymity key derivation function for resynchronization	O – (MILENAGE)	—
f6	MAP encryption algorithm	S	MAP nodes
f7	MAP integrity algorithm	S	—
f8	UMTS encryption algorithm	S – (KASUMI)	MS and RNC
f9	UMTS integrity algorithm	S – (KASUMI)	—

Table 1. *UMTS security algorithms.*

[5] has been developed by the 3G Partnership Project (3GPP). The MAPsec protocol belongs to the Network Domain Security (NDS) work area in 3GPP. NDS covers both the MAPsec specification and specifications for how to protect IP connections on the control plane of the UMTS core network [6]. The topic of UMTS network domain security is considered outside the scope of this article and will not be further discussed here.

The second AKA stage is where the SGSN/VLR executes the one-pass challenge-response procedure to achieve mutual entity authentication between the USIM and the network (SN, HE). A point to be made is that in a two-staged AKA approach, the HE delegates responsibility for executing the security procedures to the SN. There must therefore be a trust relationship between the HE and the SN in this matter. In the GSM environment, this trust relationship was regulated through roaming agreements; the same model should be applicable to UMTS.

The cryptographic functions (f0-f5*) used in the AKA procedure are implemented exclusively in the USIM and AuC. UMTS operators are free to choose any algorithm they want provided it complies with the function input/output specification given in [4]. However, 3GPP has developed the MILENAGE [7, 8] algorithm set to provide the AKA functions. The formal status of MILENAGE is that it is provided as an example algorithm set, but in practice it is the recommended algorithm set for the AKA functions. MILENAGE itself is built around the symmetric block cipher Rijndael. Table 1 depicts the cryptographic functions and their use.

A consequence of having mutual authentication is that the USIM is now an active entity. In GSM, the user could not authenticate the network; hence, the UE could not reject the network. In UMTS, the USIM will attempt to authenticate the network and it is now possible that the USIM will reject the network.

Protection of the Access Link

The *confidentiality* security service is realized by means of encryption. The cryptographic keys to be used are generated by the AKA procedure. The cipher key (CK) is always 128 bits long, but one can control the number of significant bits by configuring the key derivation (f3) function. The default produced by MILENAGE f3 is for a confidentiality key of 128 significant bits.

In GSM, confidentiality is normally terminated in the base station. This is in line with the original design objective where the primary goal was to protect against eavesdropping on the radio interface. However, it has been observed that a substantial amount of the connections between base stations and controllers are based on unsecured radio link hops. For UMTS it seemed justified to extend the coverage of the encrypted connections. Figure 2 shows that in UMTS, encryption takes place between the MS and the radio network controller (RNC).

The *integrity* security service is realized by means of a message authentication code (MAC) mechanism that provides both message authentication and integrity protection against deliberate modifications. The integrity key (IK) is always 128 bits long, but similar to CK, one can also configure IK to have fewer significant bits if required. The default MILENAGE f4 function produces an IK with 128 significant bits. Integrity protection in UMTS covers the same physical range as confidentiality protection does (i.e., integrity protection is employed between the MS and the RNC). While confidentiality in UMTS covers both user-related system signaling and user data, integrity protection only covers system signaling.

Much of the work with the UMTS access architecture has been focused on backward compatibility with GSM/GPRS. From a security point of view, backward compatibility with a system with weaker security is very undesirable. However, commercial reality dictates backward

```
Authentication Vector = AV
{     RAND   :      128-bit;      ---    Pseudo-random number, challenge data;
      XRES   :      32-128 bit;   ---    Expected Response, answer to challenge;
      CK     :      128-bit;      ---    Cipher Key;
      IK     :      128-bit;      ---    Integrity Key;
      AUTN   :      128-bit;      ---    Authentication Token, challenge data;
}

Authentication Token = AUTN
{     SQN    :      48-bit;       ---    Sequence Number;
      AMF    :      16-bit;       ---    Authentication Management Field;
      MAC-A  :      64-bit;       ---    MAC value used for Authentication;
}
```

Figure 3. *The composite AV information element.*

compatibility. To reduce the risks and weakness associated with backward compatibility, a lot of measures have been taken. This article will not discuss this issue further, but suffice it to say that it has been an important topic in the standardization work.

Authentication and Key Agreement

A Two-staged Approach

Implicitly the two-staged AKA assumes a trust model where one requires a level of trust between the roaming partners with respect to handling foreign subscribers. The choice of the two-staged approach has its origin back in the GSM system. The two-staged approach was originally conceived in an environment (Europe) where the national public telephone operators had a monopoly on all telecommunication services. In that environment, the companies were few and the involved parties knew each other well. Largely, the companies were technically adept and could in principle all be trusted. In that setting, HE control was not a deciding issue.

One-Pass Mutual Entity Authentication and Key Agreement

The authentication sequence executed between the SGSN/VLR and the USIM is based on a mutual authentication scheme using a long-term preshared secret key, K (128-bit). The master key K is only stored on the UICC/USIM and in the AuC in the HE. The UICC is a tamper-resistant smartcard subscriber identity module, and the USIM is an application running on the UICC. For security to be maintained, it is a fundamental requirement that K is never exposed or otherwise compromised for the given UICC/USIM during its lifetime.

The AKA sequence is normally initiated by the VLR/SGSN when the network needs to verify the identity of the subscriber. If the SGSN/VLR does not already possess a valid authentication vector (AV) (Fig. 3) for the claimed subscriber identity, it must request at least one AV from the HLR/AuC. The AV is computed by and stored at the AuC node in the HE. The AV is generated by means of the operator-specific authentication functions (f0-f5*).

The f0 function, which produces the RAND part of the challenge, is worth mentioning since it is the only function that is only present at the AuC. Definition 0.0 shows that f0 shall only depend on the internal state to produce another output.

```
(0.0)  f0(internal-state) → RAND
```

It is essential that f0 output does not repeat during the lifetime of the UICC/USIM, since this would allow the possibility that an adversary listened in on the previous challenge-response exchange where the particular value of RAND was used. The adversary would then know the response (RES).[4]

The SGSN/VLR initiates the local AKA procedure by sending the challenge message that contains the random challenge RAND and an authentication token AUTN. Authentication of the network side is based on message authentication of the challenge data (function f1, Definition 0.1). This provides corroboration that only an entity with knowledge of the secret key K could have produced the received challenge. The procedure is shown in Fig. 4. The basic construct of the mutual authentication scheme used in UMTS is derived from a procedure described in ISO/IEC 9798-4 [9, Sec. 5.1.1].

This sequence has two significant characteristics:

- The procedure is executed in a single round-trip (one-pass).
- The procedure augments a basic (unilateral) challenge-response mechanism with a MAC to provide mutual authentication (at the cost of a sequence number mechanism).

The choice of a one-pass AKA scheme attests to the significance attached to the performance of the AKA. One cannot afford to spend more time than strictly necessary during connection setup. The choice of an AKA mechanism based on MACs can be discussed. MAC-based solutions have excellent computational performance, which is essential since the f1 and f2 functions must be executed on the UICC/USIM. One could alternatively have designed the challenge-response mechanism with use of public key technology and digital signatures. Given the fact that the authentication algorithms must be executed under real-time restraints, the 3GPP Security working group (SA3) decided to rely on conventional methods based on MAC functions. MAC functions were already in use in the GSM/GPRS system, and this certainly influenced the decision.

One could alternatively have designed the challenge-response mechanism with use of public key technology and digital signatures. Given the fact that the authentication algorithms must be executed under real-time restraints, one decided to rely on conventional methods based on MAC functions.

[4] *Note that the sequence number mechanism may still provide some protection against such an attack*

The resynchronization procedure, which for performance reasons should not occur too frequently, involves the SGSN/VLR requesting a new and fresh AV from the HLR/AuC. The quality of the selected sequence number management scheme will significantly affect the rate of resynchronization events.

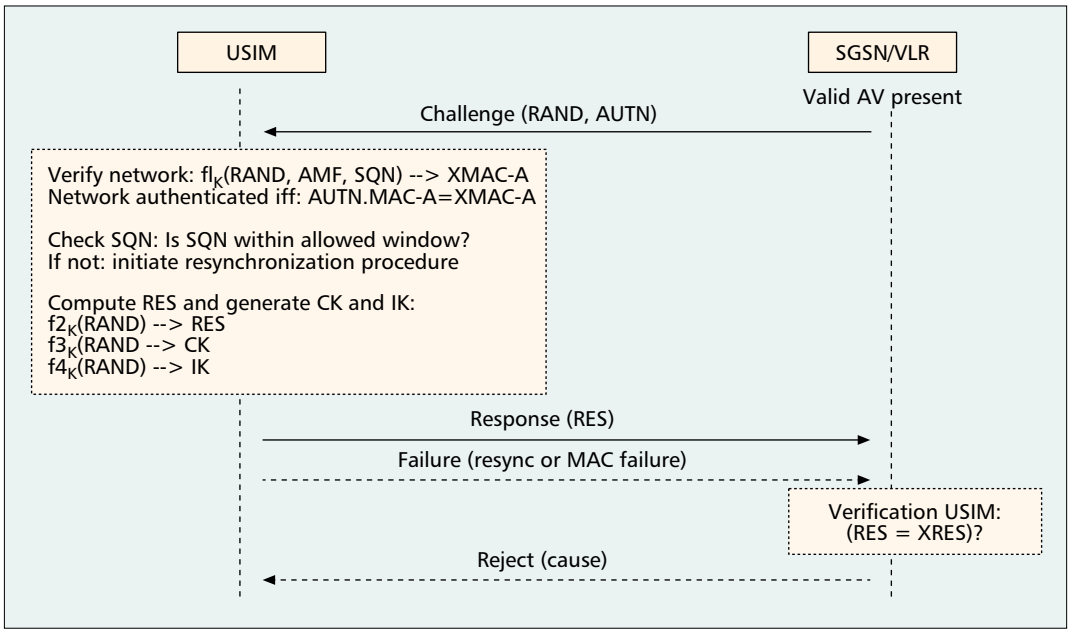


Figure 4. *UMTS authentication and key agreement.*

After receiving the challenge, the USIM will start trying to verify the identity of the network. This is done by executing the f1 function for the received RAND, AUTN tuple. The USIM then compares the computed XMAC-A with the received MAC-A. The network is considered to be verified[5] if XMAC-A is equal to the MAC-A parameter contained in the AUTN information element.

(0.1) $f1_K$(RAND,SQN,AMF) → MAC-A (or XMAC-A)

Then the USIM must verify that the sequence number SQN is within the valid range. This is done by means of a window mechanism.[6] After successful verification, the window is adjusted according to the now accepted challenge. Having accepted the challenge, the USIM must also produce a response (RES) to be sent to the network. It will simultaneously also produce the session keys CK and IK.

(0.2) $f2_K$(RAND) → RES (or XRES)

Subsequently, the SGSN/VLR will verify that the received RES value is identical to value of the XRES that is part of the AV.

(0.3) $f3_K$(RAND) → CK

(0.4) $f4_K$(RAND) → IK

The AKA procedure can also optionally use an anonymity key (AK), produced by the f5 function, to conceal the value of the sequence number stored in the SQN value. The concealment, which aims at making location tracking harder, is done by XOR-ing the AK and SQN. Note that the f5 function must be executed prior to f1 in order to derive the SQN parameter.

(0.5) $f5_K$(RAND) → AK

A consequence of the one-pass nature of the AKA procedure is that the USIM may receive challenges based on valid but expired AVs. In contrast to the case where a challenge is proven to be invalid, the occurrence of an outdated AV is treated as a case of lost synchronization and not as an authentication error. Consequently, the AKA procedure also contains functionality to regain synchronization. The resynchronization procedure, which for performance reasons should not occur too frequently, involves the SGSN/VLR requesting a new and fresh AV from the HLR/AuC. The quality of the selected sequence number management scheme will significantly affect the rate of resynchronization events.

The Operator Specific Functions and the MILENAGE Algorithms Set

The cryptographic functions f0–f5* (Table 1) are in principle operator-specific functions, and there is no need for any interoperability of these functions between roaming partners. The functions themselves reside exclusively on the USIM and in the AuC, both entities under control of the HE. Even so, it was decided to develop an example set of functions to be made available to vendors and operators. The rationale for doing this is to ensure that the UMTS community has a solid set of functions available so as not to delay deployments of UMTS or compromise its security by means of poor authentication functions.

The example algorithm set was developed by the European Telecommunications Standards Institute Security Algorithms Group of Experts (ETSI SAGE) under commission of the SA3 working group. The example algorithms were to be based on a common cryptographic core engine that was explicitly required to be a block cipher. The framework should by design be sufficiently generic to allow operators to replace the cryptographic core if they so wish. The result of the design effort was the MILENAGE framework, which can be made to work with any block

[5] *We do not distinguish here between HE and SN.*

[6] *The exact sequence number management mechanism to be used is in principle decided by the operator, but the example mechanisms found in Annex C in 3G TS 33.102 [3] is expected to be widely adopted.*

cipher with 128-bit blocks under control of a 128-bit key.

The MILENAGE framework, which does not include the pseudo-random number generator function f0, is based on a cryptographic core consisting of the Rijndael block cipher. The choice of Rijndael as the basis for MILENAGE was made before Rijndael became the Advanced Encryption Standard (FIPS-197, [10]) algorithm. The main arguments used by ETSI SAGE for choosing Rijndael points to the algorithm's excellent performance on platforms with limited computing resources, the fact that it was extensively evaluated during the AES selection process, and the fact that it is IPR-free. The performance characteristics are important since the authentication functions must be executed on the fly on smartcards with limited resources. To allow for some degree of operator customization, MILENAGE defines a 128-bit operator variant algorithm configuration field (OP) that is used as input to the functions f1–f5*. OP itself should not be present on the UICC/USIM; instead, the derived (encrypted) element OPC is used (Fig. 5).

The functions f1, f1*, and f2 are explicitly required to be MAC functions [4]. In particular, the requirements state that it should be computationally infeasible to derive the long-term secret K from the parameters received or computed during the AKA procedure. The functions f3, f4, f5, and f5* are so-called key derivation functions. Again, it is explicitly required that K cannot be derived from the other AKA input/output parameters. This is even true for the session keys CK and IK, since the compromise of a session key should never itself lead to compromise of the long-term key K. In practice, as demonstrated by the MILENAGE approach (Fig. 5), there is no principal difference in how the functions are constructed with the exception that functions f1 and f1* take different inputs from the other functions.

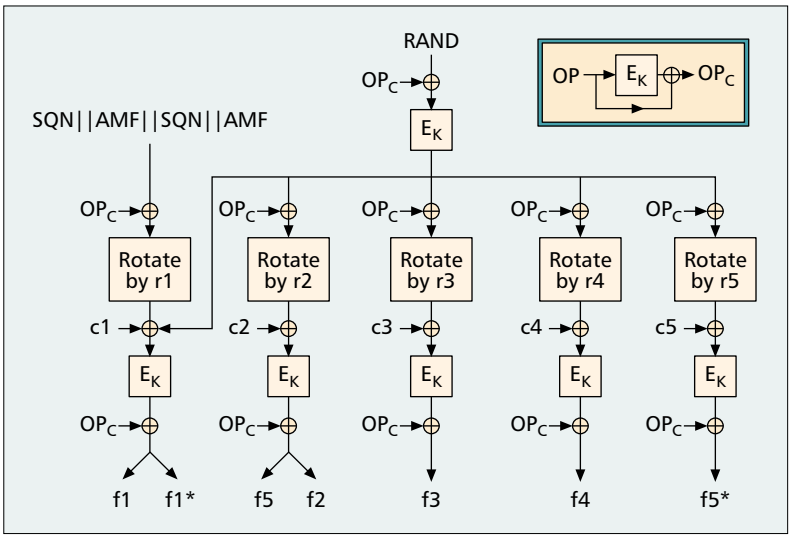


Figure 5. *The MILENAGE algorithm set (from [8, Annex 1]).*

Confidentiality and Integrity Protection

The KASUMI Cryptographic Core

The subscriber module UICC/USIM is issued by the HE; the authentication related security functions on the USIM have their counterpart in the AuC located in the HE. Contrast this to the encryption and integrity functions that are located in the mobile device and the corresponding SN. Here it is essential that the security architecture have fully standardized default encryption (f8) and integrity (f9) functions to ensure smooth roaming to different serving networks. Functions f8 and f9 are specified in [11].

The security architecture allows for 16 different encryption algorithms and 16 different integrity algorithms specified through the UMTS Encryption Algorithm (UEA) identifier and UMTS Integrity Algorithm (UIA) identifier. The standard algorithms are identified as UEA 1 and UIA 1. With the exception that UEA 0 is explicitly defined as a null algorithm, no other assignments to UEA or UIA have yet been made.

The cryptographic core of the standard algorithms is based on the KASUMI [12] block cipher. KASUMI is a Feistel cipher with eight rounds. It operates on a 64-bit data block under control by a 128-bit key. The KASUMI algorithm is derived from Mitsubishi Electric Corporation's MISTY1 algorithm. The design guidelines for MISTY1 were to base it firmly on mathematical/numerical properties, allow it to be reasonably fast in software on any processor, and finally be fast in hardware. MISTY1 was designed to be provably secure against differential and linear cryptoanalysis, and the design is built around small components with known resistance against these two types of attacks. These components are then recursively used in the Feistel network. The S-boxes (S7 and S9) are designed to have minimal average differential/linear probability, to be efficient in hardware, and to have high nonlinearity order.

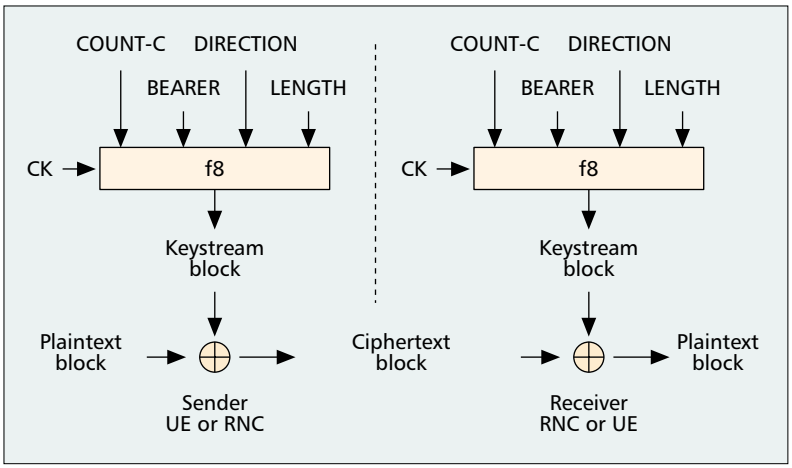
The changes between MISTY1 and KASUMI can broadly be divided into two areas:

- **Modifications to the data encryption part.** These changes were motivated by hardware performance gains, but also to minimize the necessary number of gates to implement KASUMI.
- **Modifications to simplify the key scheduling.** The main motivation here was to reduce the key setup time.

The work to adapt MISTY1 to 3GPP requirements and develop the modes of operation was commissioned by 3GPP SA3 to be done by ETSI SAGE. To ensure that the quality of the KASUMI cryptographic core and functions f8 and f9 was as high as possible, a separate evaluation round was carried out. The reports from the evaluators led to some improvements, but the basic design was not changed. The general conclusion was that the KASUMI algorithms were based on sound design principles, and no practical attacks were found for use within the UMTS context.

Confidentiality Protection

The UMTS access security encryption function f8 is a link layer symmetric synchronous stream cipher. The f8 function is specified to produce a



■ Figure 6. *Use of the f8 algorithm.*

pseudo-random keystream block that is combined with a plaintext block by means of bitwise modulo 2 operations (XOR function). The f8 function takes a 128-bit key CK, but operates internally on 64-bit blocks. The output keystream block can be up to 20,000 bits. The block cipher KASUMI operates in a variant of the output-feedback mode and generates keystream blocks in multiples of 64 bits. In the case of redundant bits, the least significant bits in the last 64-bit block are discarded.

The generic confidentiality function f8 (Fig. 6) takes as input the confidentiality key (CK, 128-bit), a sequence number (COUNT-C, 32-bit) derived from the layer 2 frame number, the radio channel indication (BEARER, 5-bit) and a direction indication (DIRECTION, 1-bit). Additionally, the length (LENGTH, 16-bit) of the keystream block is provided.

Integrity Protection

Integrity protection in UMTS is limited to covering signaling messages between the MS and the RNC. The generic integrity function f9 (Fig. 7) takes as input the integrity key (IK, 128-bit), the message (MESSAGE) to be protected, a sequence number (COUNT-I, 32-bit) derived from the layer 2 frame number, a random value (FRESH, 32-bit), and a direction indication (DIRECTION, 1-bit) value.

The computed MAC-I is included in the signaling message by the sending side. The receiving side computes the corresponding XMAC-I over the message, and data integrity is considered to be verified if the computed XMAC-I and the received MAC-I are identical.

Signaling messages are generally short; thus, the length of the MESSAGE elements are correspondingly short. The actual length of the MESSAGE element presented to the f9 function is longer than the message sent over the air since the five bits used to indicate the bearer channel are extracted from the radio bearer context. The specifications somewhat arbitrarily restrict the size of an f9 input message to 5000 bits.

The standard f9 function is based on the KASUMI block cipher. The f9 function is a variant of the familiar cipher-block-chaining message authentication code construction (CBC-MAC) method. Had a regular CBC-MAC mode been used for KASUMI, it would have been restricted by the internal block size of 64 bits, but a novel chaining technique has allowed the f9 function to maintain a 128-bit internal state. The final output from the KASUMI used in f9 is a 64-bit cipherblock, which is truncated to become the 32-bit MAC value.

Discussion and Brief Analysis UMTS AKA

As has already been established, the two-staged AKA approach implicitly assumes a trust model where the roaming partners must trust each other with respect to security handling of roaming subscribers. This is not always a realistic assumption. It can, of course, be argued that a two-staged approach is better in the sense that it allows for distributed processing and load sharing. On the other hand, the absence of a global AKA procedure executed logically directly between the HE and the USIM, even as an option, can be considered an omission of the system. Then again, a distributed AKA scheme is likely to execute faster for roaming subscribers than a global scheme. Since the processing delays of the AKA sequence affect the call setup time, this is not a negligible argument.

The choice of a one-pass challenge-response mechanism based on MAC functions for the AKA sequence is interesting. One obvious advantage of this approach is that one could retain the signaling sequence and state machine of the GSM AKA transfer protocol. The GSM AKA protocol machine was stable and readily available, a significant but not decisive factor. A factor in favor of a MAC-based scheme is that the UMTS standard is global. In some countries encryption is either restricted or banned, so an AKA scheme based on encryption would possibly have run into political problems.

During the design phase a full mutual challenge-response was discussed, but eventually abandoned for performance reasons since it would require an additional round-trip. The ACTS USECA[7] project also investigated the use of public key technologies for the AKA. Modern smartcards are capable of executing public key algorithms, but in the end the benefits of a public-key-based AKA seemed not to justify the extra cost of more expensive smartcards and slightly higher computational delay. This decision can be discussed, and in some ways it is unfortunate that the UICC/USIM does not have public key encryption capabilities. Such functionality would have been very useful for e-commerce purposes, and it is with some irony that today we are working actively to enhance the UICC/USIM with digital subscriber certificates capabilities [13].

Another aspect of the AKA scheme to note is the sequence number management. It is important that resynchronization events be infrequent. The security architecture [3] contains an example in Annex C. It should be noted that Annex C allows for variations in how sequence numbers are managed. The actual performance is a function of both the configuration

[7] *The project archive is at http://www.isrc.rhul.ac.uk/useca/*

of the sequence number management, behavior of the subscribers, and various non-security-related network optimization features. Operators are advised to closely monitor the performance of the sequence number management mechanism.

One must also scrutinize the MILENAGE algorithm set. When ETSI SAGE decided to use Rijndael, the choice made a lot of sense. However, during 2002 AES and thereby Rijndael have come under some criticism. The algebraic structure of Rijndael is elegant and simple, and some papers suggests that the structure can be very simple indeed if one views the algebraic equations of Rijndael in GF(2^8) [14]. There is not yet a consensus on whether the recent analysis of Rijndael can be converted into realistic attacks. In fact, many respected cryptographers view the analysis as academically interesting, but dispute the possibility of realistic attacks. Should Rijndael be susceptible to attacks based on the recent analysis, MILENAGE would suffer since the long-term secret key K is long-lived (on the order of one to three years). As it stands, even if the computational complexity of Rijndael is shown to be down to 2^{87}, Rijndael will still be sufficiently strong to fulfill its role in MILENAGE for the next few years. However, to be prepared for any eventuality, SA3 will investigate the potential weaknesses of MILENAGE based on Rijndael and also investigate the possibility of developing a second version of MILENAGE based on a different cryptographic core.

Confidentiality Protection

The use of a block cipher in output-feedback mode (OFM) for building a stream cipher is considered fairly common. The construct has some known problems associated with it, specifically that there is a nonzero risk that the cycles produced can be comparatively short. It is highly desirable to avoid having the keystream generator enter exactly the same state during a session since that will lead to output of a predictable keystream block.

The KASUMI f8 algorithm takes as input the session key CK (128-bit), the BEARER (5-bit) identifier, the COUNT-C (32-bit) frame indicator, and the DIRECTION (1-bit) setting. The LENGTH parameter only affects the length of the output keystream block and has no influence on the internal state of the keystream generator.

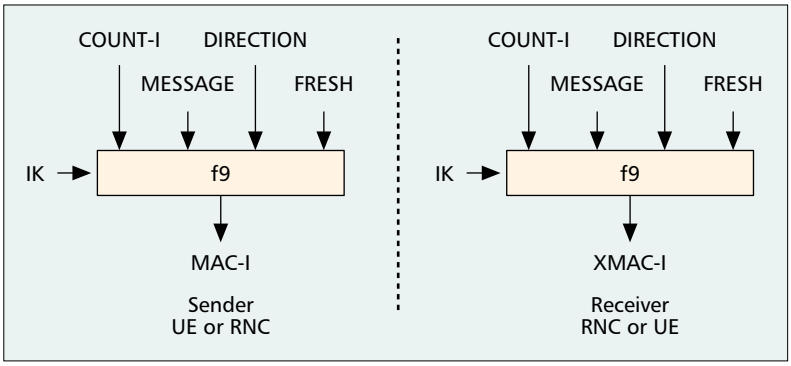


Figure 7. *Use of the f9 integrity algorithm.*

A general property of OFM ciphers is that they can recover from bit failures in the ciphertext. An error in a ciphertext bit will lead to an error in the corresponding plaintext bit, but the error does not spread or propagate further. This property was vital in GSM where the encryption/decryption was done just before modulation (Fig. 8). Error spreading and propagation would have been damaging. However, it is still a desirable property for the f8 function not to spread or propagate errors. This is so since there may be residual errors after deinterleaving and decoding (which includes error correction and error detection). For *transparent* services, timely data is more important than error-free data. Then retransmission is not an option, and one proceeds with data that may still include errors. Obviously, it is desirable for f8 not to spread or propagate any residual error.

Another cipher algorithm attribute that might be considered desirable is the property to self-synchronize in the face of lost ciphertext bits since one may expect a cellular system to occa-

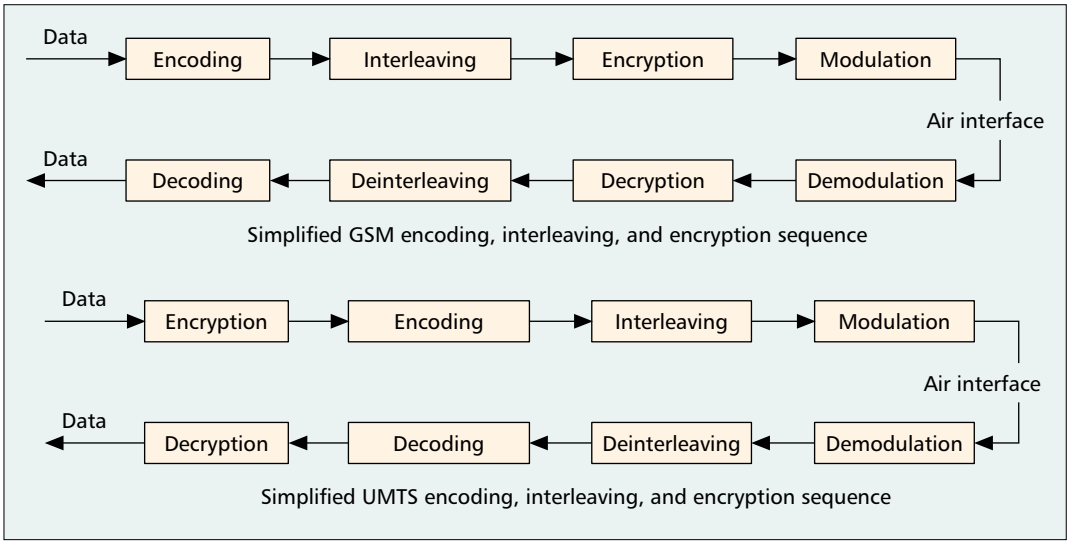


Figure 8. *Simplified encoding(error protection), interleaving, encryption, and modulation schemes.*

The lack of user data integrity protection introduces a problem in countries where encryption is not allowed or otherwise not available, since it opens up the possibility for an attacker to illicitly modify user data or effect session hijacking.

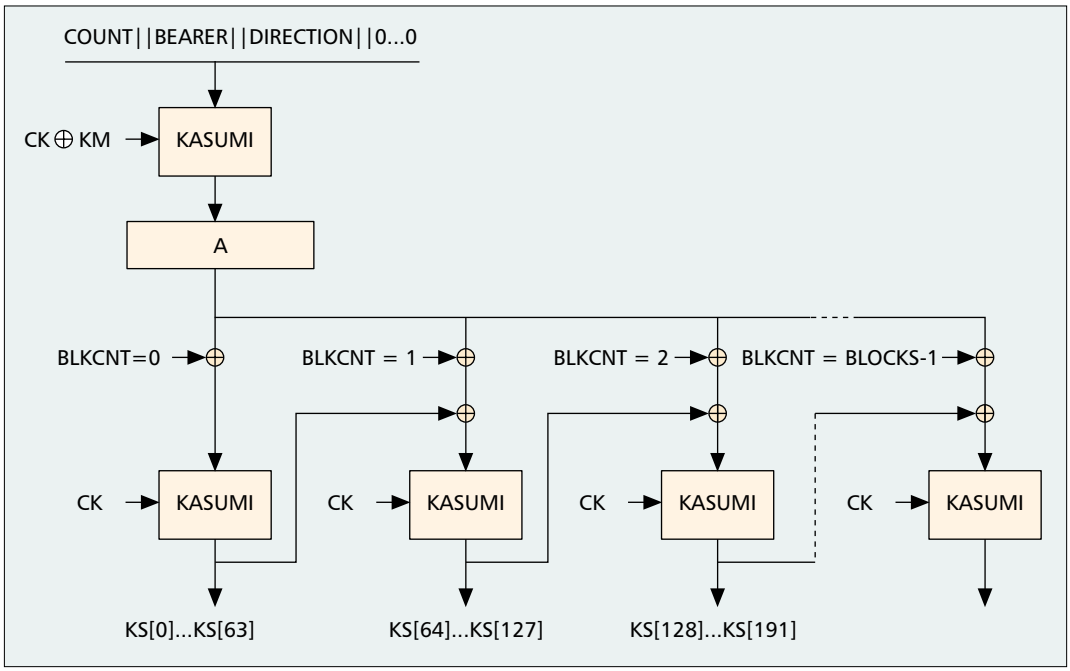


Figure 9. *The f8 keystream generator (from [11, Annex 1]).*

sionally lose bits over the air. However, block ciphers in OFM cannot self-synchronize in the event of lost ciphertext bits. Analysis nevertheless reveals that the KASUMI f8 function will be synchronized at the start of every new block. The f8 algorithm takes the COUNT-C, DIRECTION, and BEARER identifiers from the radio link as input, and it is consequently directly synchronized with the bitstream as it appears to the radio system.

The initialization vector (IV) used in f8 is constructed by concatenating the identifiers COUNT-C, BEARER, and DIRECTION. To create a 64-bit IV, the remaining 26 bits are all set to zero. The IV is used to provide sufficient initial "uniqueness" to the keystream generator.

```
(0.6)  IV = COUNT-C || BEARER ||
DIRECTION || ZERO(26-bit)
```

It is observed that the IV (Definition 0.6) is essentially predictable. It is furthermore noted that an adversary may theoretically be able to provoke radio-related events like handover or location updating at specific points in time. Given the low entropy of the highly structured signaling messages and the fact that large portions of signaling message contents can be deduced from the radio context, an adversary may conceivably be able to execute known plaintext attacks.

However, the OFM variant found in f8 is quite resistant to this type of attack. The predictable IV is used only once for each cipherstream, and the IV is completely independent of the plaintext/ciphertext of the preceding cipherstream. To complicate things further for the would-be adversary, the IV as shown in Definition 0.6 is not used directly. The IV is instead encrypted by one round of KASUMI using a modified encryption key and put in the internal 64-bit register A (Fig. 9). The key used is CK XORed with a key modifier constant KM (Definition 0.7). The net effect is that it is very difficult to analyze and predict the pattern of the used IV based on observations of the radio related parameters COUNT-C, BEARER, and DIRECTION.

```
(0.7)  Key Modifier KM =
0x55555555555555555555555555555555
```

$$(0.8)\quad \mathrm{KASUMI}_{CK \oplus KM}(IV) \rightarrow \mathrm{Register\text{-}A}$$

KASUMI therefore achieves synchronization with the air interface while in practice avoiding the hazards of a predictable IV. Therefore, the conclusion is that while there may be a slight theoretical weakness in this area, the KASUMI f8 function is very unlikely to be vulnerable to this kind of attack.

The KASUMI f8 function was designed to be able to sustain transmission of up to 2 Mb/s simultaneously on the uplink and downlink. However, radio performance development in UMTS has not stopped, and higher bit rates[8] are now being specified. This in itself does not pose a big problem for KASUMI f8. One nevertheless observes that as bit rates keep increasing, the need to redesign the encryption function f8 will become more pressing.

Integrity Protection

Integrity protection in UMTS is limited to coverage of system signaling messages between the MS and the RNC. The fact that user data is not integrity-protected represents a problem under certain circumstances. In particular, there will be situations where encryption is not available and integrity protection is the only line of defense. Thus, the lack of user data integrity protection introduces a problem in countries where encryption is not allowed or otherwise not available, since it opens up the possibility for an attacker to illicitly modify user data or effect session hijacking. In retrospect, it was a mistake not to provide integrity protection for user data.

[8] *High-speed downlink packet access (HSDPA) provides a peak rate in excess of 10Mb/s downlink (UMTS Release 5).*

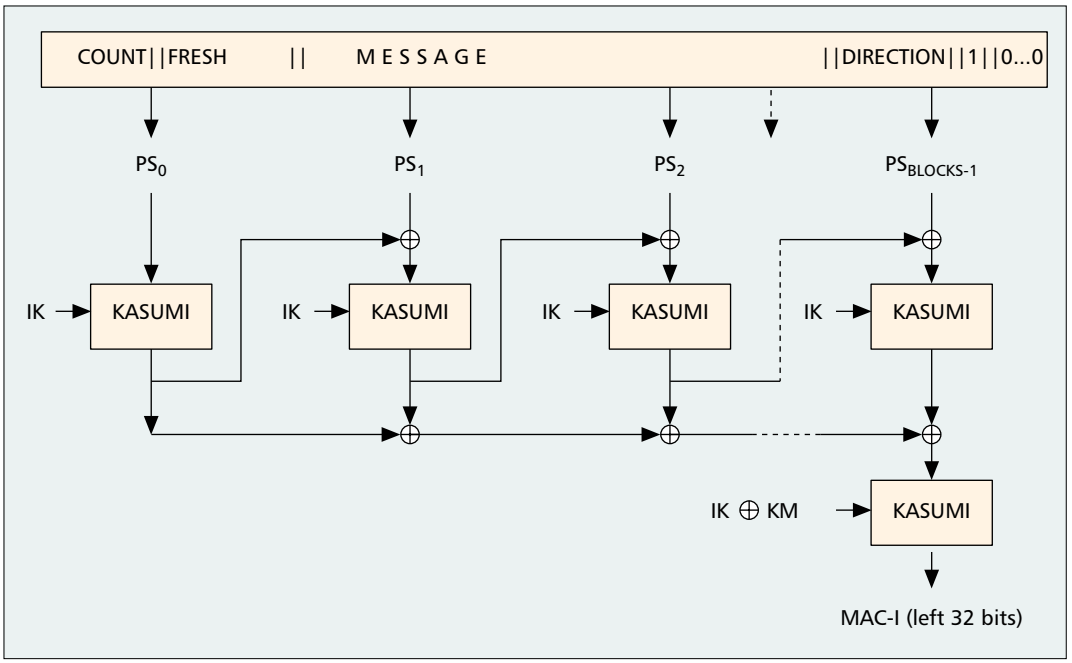


Figure 10. *The f9 integrity algorithm (from [11, Annex 1]).*

The cryptographic strength of the KASUMI f9 integrity protection algorithm (Fig. 10) is found to be sufficient. As one would expect, attacks based on the birthday paradox are possible. Based on a 128-bit internal state, this would require on the order of 2^{65} chosen texts, which is completely outside reach. A variation on the birthday attack found by Mitchell/Knudsen (noted in [15]) requires approximately 2^{48} chosen texts. However, the output of the f9 function is only 32 bits long (Fig. 10). Given this, it seems likely that an adversary would instead attempt a more direct attack based on the length of the cryptographic checksum.

The fact that there are collisions does not mean it is easy to find meaningful collisions. The task of finding meaningful collisions is substantially more difficult than just finding a collision.[9] To emphasize this, the KASUMI f9 function is intended to be used for protection of real-time signaling messages with short expiry periods. It is also unlikely that a successful attack can be mounted without falsifying several consecutive signaling messages. It is in this context that one must evaluate the identified weaknesses.

As mentioned above, the actual MAC value used is only 32 bits wide. This would under most circumstances be considered utterly insufficient. In real life, a balance must be struck between the overhead introduced by longer MAC values and their benefits. For the generally very short signaling messages, even a 32-bit MAC increases the signaling load substantially. At the other end, the provision of longer MAC values would make some signaling messages exceed the layer 2 frame size. The message would then have to be segmented in order to be transmitted. For time-critical sequences this is not always allowed; furthermore, segmentation is performance-wise very undesirable. In the context of the real-time nature of the signaling system, an adversary will not have much time to execute an attack. In this setting, the choice of only a 32-bit MAC value may be considered sufficient even if the length of the checksum is dangerously small.

Alternative Algorithms for f8 and f9

The KASUMI f8/f9 function is just one possible implementation of the f8 (encryption) and f9 (integrity) functions. KASUMI seems solid enough for its intended usage, and while it may not necessarily be a safe choice for its designed lifetime of 20 years, there does not seem to be any immediate danger. One of the independent evaluation groups advised that the suitability of KASUMI f8/f9 be reassessed after five years [15]. Due to the long lead times on development and deployment of new equipment that must be both fully standardized and mandatory to implement, SA3 has already begun preparing for development of a new algorithm. The mandate for the new development effort will also consider the structure of the f8 and f9 modes of operation, not just the core cryptographic engine.

Key Transfer

One area not originally standardized in the 3GPP security architecture is how to securely transfer CK and IK from the core network to the radio network system. The Iu interface (Fig. 1) between the SGSN/VLR and the RNC contained no standardized protection mechanisms. Since the confidentiality and integrity of CK and IK are vital to the security attained over the radio interface, it is imperative that the operator/vendor provide full confidentiality and integrity protection over the Iu interface. UMTS Release 6 has extended the scope of the NDS for IP (NDS/IP [6]) specification, so we have the option of securing IP protocols over the Iu interface. This will solve the key transfer problem for operators that choose to use IP as the preferred transport protocol over the Iu interface.

The reports from the evaluators led to some improvements, but the basic design was not changed. The general conclusion was that the KASUMI algorithms were based on sound design principles, and no practical attacks were found for use within the UMTS context.

[9] *This will in practice amount to finding a meaningful second pre-image of the integrity checksum. For a second pre-image to be meaningful it would have to be a syntactically valid signaling message that would be acceptable in the current state of the protocol state machine and so on.*

Used within its intended scope, the KASUMI f8 algorithm should remain cryptographically safe for years to come. It remains to be seen if it can stay safe for the 20-year life span for which it was designed

SUMMARY

3G IMPROVEMENTS

The access security mechanisms in UMTS are substantially better than its 2G predecessor, GSM. The challenge-response mechanism in UMTS provides mutual authentication and thereby completely removes the very real problem of false base stations in GSM. The example authentication algorithm set in MILENAGE is *much* better than the algorithm sets available to GSM.[10]

The confidentiality algorithm in UMTS is vastly better than its 64-bit GSM equivalent, A5. Used within its intended scope, the KASUMI f8 algorithm should remain cryptographically safe for years to come. It remains to be seen if it can stay safe for the 20-year life span for which it was designed, but since the security architecture allows for additional algorithms to be added at a later stage, this should not be a problem.

The integrity function in UMTS is new since GSM. The integrity mechanism is independent of confidentiality protection and can therefore provide protection in environments where encryption is not allowed or otherwise not available. Provision of an integrity mechanism is also important in the sense that it provides protection against active attacks. An omission in the current integrity protection mechanism is that it does not cover user data.

Overall, the security architecture found in UMTS has some shortcomings but nevertheless marks a large step in the right direction for cellular security.

[10] GSM has several example authentication algorithms. Among these is the infamous COMP128 algorithm, which has been shown to be fundamentally broken.

REFERENCES

[1] 3G TS 33.120, 3G Security; Security Principles and Objectives.
[2] 3G TS 21.133, 3G Security; Security Threats and Requirements.
[3] 3G TS 33.102, 3G Security; Security Architecture.
[4] 3G TS 33.105, 3G Security; Cryptographic Algorithm Requirements.
[5] 3G TS 33.200, 3G Security; Network Domain Security; MAP Application Layer Security.
[6] 3G TS 33.210, 3G Security; Network Domain Security; IP Network Layer Security.
[7] 3G TS 35.205, 3G Security; Specification of the MILENAGE Algorithm Set: An Example Algorithm Set for the 3GPP Authentication and Key Generation Functions f1, f1*, f2, f3, f4, f5, and f5*; Document 1: General.
[8] 3G TS 35.206, 3G Security; Specification of the MILENAGE Algorithm Set: An Example Algorithm Set for the 3GPP Authentication and Key Generation Functions f1, f1*, f2, f3, f4, f5, and f5*; Document 2: Algorithm Specification.
[9] ISO/IEC 9798-4: Information Technology — Security Techniques - Entity Authentication - Part 4: Mechanisms Using a Cryptographic Check Function.
[10] National Institute of Standards and Technology (NIST) FIPS-197, Advanced Encryption Standard (AES) (FIPS PUB 197), Nov. 26, 2001.
[11] 3G TS 35.201, 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification.
[12] 3G TS 35.202, 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI specification.
[13] 3GPP, Document TSGS#14(01)0622, Work Item Description: Support for Subscriber Certificates, SA#14, Tokyo, Japan, Dec. 2001.
[14] S. Murphy and M. J. B. Robshaw, "Essential Algebraic Structure Within the AES," *Proc. Crypto2002*, *LNCS*, vol. 2442, Springer-Verlag, 2002, pp. 1–16.
[15] 3G TS 33.904, 3GPP, SAGE; Report on the Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms (SAGE v. 2.0).

BIOGRAPHY

GEIR M. KØIEN (geir-myrdahl.koien@telenor.com) is a research director at Telenor R&D,Norway. He has worked with cellular systems and security for more than a decade. The last few years he has been Telenor's delegate to the 3GPP security working group (SA3), where he has also served as rapporteur for technical specification in the network domain security area. He is currently on leave to pursue a Ph.D. degree at Agder University College, Norway/ Aalborg University, Denmark.