(19) DANMARK

(10) **DK/EP 2291946 T4**

Patent- og
Varemærkestyrelsen

(12) Oversættelse af ændret
europæisk patentskrift

(51) Int.Cl.: ***H 04 L 9/08 (2006.01)*** ***H 04 L 9/32 (2006.01)*** ***H 04 W 12/04 (2009.01)***
H 04 W 12/06 (2009.01)

(45) Oversættelsen bekendtgjort den: **2020-08-31**

(80) Dato for Den Europæiske Patentmyndigheds
bekendtgørelse om opretholdelse af patentet i ændret form: **2020-06-03**

(86) Europæisk ansøgning nr.: **08784926.1**

(86) Europæisk indleveringsdag: **2008-07-21**

(87) Den europæiske ansøgnings publiceringsdag: **2011-03-09**

(86) International ansøgning nr.: **EP2008005960**

(87) Internationalt publikationsnr.: **WO2009146729**

(30) Prioritet: **2008-06-06 US 59386 P**

(84) Designerede stater: **AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR**

(73) Patenthaver: **Telefonaktiebolaget LM Ericsson (publ) , 164 83 Stockholm, Sverige**

(72) Opfinder: **NORRMAN, Karl, Stigbergsgatan 32A, S-116 28 Stockholm, Sverige**
NÄSLUND, Mats, Stopvägen 95, S-168 36 Bromma, Sverige

(74) Fuldmægtig i Danmark: **Marks & Clerk (Luxembourg) LLP, 44 rue de la Vallée, B.P. 1775, L-1017 Luxembourg, Luxembourg**

(54) Benævnelse: **GENERERING AF KRYPTERINGSNØGLE**

(56) Fremdragne publikationer:
WO-A-2005/032201
WO-A-2007/062882
WO-A-2008/054320
US-A1- 2003 053 629
US-B1- 7 131 006
3GPP: "3RD GENERATION PARTNERSHIP PROJECTS; TECHNICAL SPECIFICATION GROUP SERVICES AND SYSTEM ASPECTS; RATIONALE AND TRACK OF SECURITY DECISIONS IN LONG TERM EVOLVED (LTE) RAN/3GPP SYSTEM ARCHITECTURE EVOLUTION (SAE) (RELEASE 8)" INTERNET CITATION, [Online] XP002445696 Retrieved from the Internet: URL:http://www.3gpp1.net/ftp/spces/archive /33%5Fseries/33.821/> [retrieved on 2007-08-06] cited in the application

DESCRIPTION

Technical Field

[0001] The present invention generally relates to a technique for generating cryptographic keys. Particularly, the invention relates to a cryptographic key generation technique that provides a high-level of security.

Background

[0002] The Authentication and Key Agreement protocol (AKA) is a challenge-response based protocol that uses symmetric cryptography. The main goals of AKA include mutual authentication by two entities communicating with each other and establishment of cryptographic keys for protecting the communication exchanged in-between. A variant of AKA is the UMTS AKA, included in the security architecture standardized by 3GPP for 3G mobile communication networks in the Technical Specification 3G TS 33.102.

[0003] The basic concept of UMTS AKA is shown in Fig. 1. Referring to this figure, the UMTS AKA protocol is run between a user equipment (UE) and a network entity (NE). The network entity initiates the AKA by sending a user authentication request to the UE. Along with the request, a random challenge, or random code (RAND), and an Authentication Token (AUTN) are sent to the UE. Upon receipt of the RAND and the AUTN, the UE, among other things, computes a Cipher Key (CK) and an Integrity Key (IK) and then uses them for ciphering and integrity functions.

[0004] The 3GPP is also undertaking the standardization of so-called "beyond-3G" communication networks. System Architecture Evolution (SAE) and Long Term Evaluation (LTE) are two closely-related aspects of the beyond-3G network. Compared with conventional 3G networks, a network based on SAE/LTE may impose higher and/or more security requirements. For instance, more cryptographic keys for securing the communication at different levels may be needed. The 3GPP has, in another standard-related document, 3GPP TR 33.821, recommended a key hierarchy for deriving more cryptographic keys for use in SAE/LTE.

[0005] Fig. 2 shows this key hierarchy. At the very top of the hierarchy is a key K, a long-term cryptographic key shared between the Universal Subscriber Identity Module (USIM) of the UE and the Authentication Center (AuC) residing in the network. One level down is a pair of cryptographic keys CK and IK which are derived by the UE, particularly by the USIM thereof, in a same or similar manner as the UMTS AKA operation mentioned above. Further down in the hierarchy is a key K_{ASME} which is derived by the UE from CK, IK, and, if necessary, some other parameters. Once derived, K_{ASME} is transferred from AuC to the access network, particularly

to the Access Securing Management Entity (ASME) of the SAE/LTE network, and then shared between the UE and the network. When the access network is based on LTE technology, the ASME's functionalities are handled by a Mobility Management Entitiy (MME).

[0006] The key K_{ASME}, and the keys "below" it in the hierarchy, may be derived by applying a certain cryptographic function. For instance,

K_{ASME} = KDF(CK || IK, 0x02 || PLMN_ID || <other_parameter>)

where KDF is based on a Generic Bootstrapping Architecture (GBA) key derivation function (KDF). One GBA KDF is specified in 3G TS 33.220.

[0007] The GBA KDF may make use of cryptographic hash functions such as the Secure Hash Algorithm (SHA) hash functions. Among many SHA hash-functions, SHA-256 is a highly secure variant since it is considered collision resistant and acts like a pseudorandom function. As its name suggests, SHA-256 is a Secure Hash Algorithm hash function with a digest (output) length of 256 bits. The PLMN_ID is an identifier of the network serving the UE.

[0008] It has been realized that, in order to achieve a high-level of security, it is not sufficient to base the GBA KDF function mainly on CK and IK only. The rationale for this is the risk that a given UE might get the same CK twice, or two different UEs may get the same CK. In such cases, the "uniqueness" of the inputs to the KDF is undermined, and a collision between different UEs (using the same K_{ASME}) may occur.

[0009] As a general remark, while it is certain that KDF(x) produces the same key as KDF(y) if x = y, the converse may not always hold. That is, even if $x \neq y$, it may still happen that KDF(x) = KDF(y). However, this is an unlikely event since the KDF is recommended to be based on SHA-256 which, as mentioned, has been designed to be collision resistant. Thus, for the technique described herein, it can be safely assumed that KDF(x) = KDF(y) if and only if x = y. This assumption allows the technique described herein to be focused on assuring "uniqueness" of the inputs to the KDF.

[0010] The standardizing body of the GBA KDF specification (ETSI/SAGE, the Special Algorithm Group of Experts) has noted the above problem and recommended including the UE's Private User Identity (IMPI) in <other_parameter> to avoid collisions between different UEs. As a further recommendation, a random code such as the RAND may also be included in <other_parameter>. This is described in a liaison statement from ETSI/SAGE to 3GPP SA3 (in 3GPP document number S3 - 030219).

[0011] However, it has been found that the above recommendations still can not guarantee the "uniqueness" of the inputs to the KDF. This can be seen from the below analysis of the security property of the GBA KDF function and its usage in SAE/LTE for one and the same UE (e.g. one and the same IMPI).

[0012] Firstly, the following basic construction is considered:
KDF(CK, IMPI).

[0013] Since it has been assumed that IMPI = IMPI' (when the UE is fixed), this basic construction will lead to collision for two inputs (CK, IMPI), (CK', IMPI') if and only if CK = CK'.

[0014] Secondly, another construction is considered, which is closer to the actual GBA KDF:
KDF(CK || IK, IMPI).

[0015] However, including IK into the inputs does not change the above collision property as one might believe at first. That is, KDF(CK ‖ IK, IMPI) will be equal to KDF(CK'‖ IK', IMPI) if and only if CK = CK'. To understand why including IK would not help, it is necessary to consider how CK and IK are produced by the cryptographic algorithm executed on the UE.

[0016] A typical UE-side cryptographic algorithm is the Milenage algorithm which is shown in Fig. 9. In Fig. 9, Ek denotes the Advanced Encryption Standard (AES) algorithm, also known as the Rijndael algorithm, using key K (stored in the AuC and USIM of UE). Consider now what happens if CK = CK'. Since AES is a permutation (a one-to-one mapping), this implies that the intermediate value (occurring at the fat arrow) is uniquely determined by the outcome of f3 which happens to be CK. But this implies that the value at the fat arrow when producing CK must be the same as the value occurring at the same place when CK' was produced. This in turn means that the values occurring as input to f4 must be the same and consequently, the same f4-values must occur. As it happens, f4 *is* IK. Thus it has been shown that CK = CK' if and only if IK = IK'.

[0017] Next, an "improved" construction according to the recommendation of the standardizing body (SAGE), i.e., including RAND in the inputs, is considered:
KDF(CK || IK, RAND || IMPI).

[0018] Assume that CK = CK' (and thus IK = IK'). It is hoped that the use of RAND will guarantee uniqueness. However, this is not true. Consider again the "relevant" part of the Milenage algorithm that produced CK and IK from RAND: As shown in Fig. 9, there is a situation in which the value at the fat arrow corresponding to RAND is the same as that corresponding to RAND'. But again, AES (Ek) is a permutation so that the *inputs* must also be equal, i.e. RAND = RAND'. (The fact that AES is dependent on K does not help since a fixed UE is assumed and thus the same K will occur in both cases.)

[0019] In other words, it has been shown that (CK ‖ IK, RAND ‖ IMPI) = (CK' ‖ IK', RAND' ‖ IMPI) if and only if RAND = RAND'. In the SAE/LTE case, the PLMN_ID may also be included in the inputs, but since it is highly likely that the UE stays in the same network several times, this parameter PLMN_ID cannot be relied upon for the purpose of guaranteeing uniqueness.

[0020] An alternative approach to attempt to avoid collision could be to use another algorithm

than AES for the cryptographic processing of the f3 and f4 algorithms. Specifically, the analysis above was based on the fact that AES is a permutation. It would therefore be possible to use a non-permutation (many-to-one mapping) instead of AES. This is problematic for two reasons. First of all, existing USIMs must be adapted to be suitable for the 3GPP SAE architecture. Secondly, by choosing a non-permutation function, one actually increases the probability that two outputs of e.g. f3 will collide.

[0021] The lack of uniqueness of the inputs can be a serious security issue. Since collision will occur if and only if RAND = RAND', and since RAND is 128 bits, the collision is expected to occur after about 2^(128/2) = 2^64 authentications (this is the so-called "birthday paradox"). Clearly, this is lower than the targeted security level of GBA (which is 128 bits). For LTE the case is even worse, since LTE is required to provide a security level of 256 bits. Thus, the high collision probability is a significant obstacle to providing the required security level in SAE/LTE.

[0022] WO 2005/032201 A relates to enhanced security design for cryptography in mobile communication systems. 3GPP Change Request 33.105 version 3.4.0 relates to anonymity key computation during re-synchronization. WO 2004/075584 A relates to a first cryptographic key and a second cryptographic key created by a mobile radio terminal and by a computer of a home network by using authentication key materials. The first cryptographic key is transmitted to a computer of a visited network, and the second cryptographic key is transmitted to an application server.

[0023] Accordingly, there is a need for a solution that avoids the collisions mentioned above. The solution should ideally also work with already deployed USIMs and not require replacing all USIMs. The present disclosure provides such a solution in accordance with the independent claims. Various embodiments of the solution are set out in the dependent claims.

[0024] According to a first aspect, a method for generating a cryptographic key is provided. The method involves, among others, the following features: The cryptographic key is used for, among others, protecting the communication between two entities. The method is carried out by the first entity. The method forms a part of an Authentication and Key Agreement procedure which is initiated by the second entity. The method comprises providing at least two parameters, wherein the first parameter either comprises or is derived from a set of cryptographic keys which have been computed by the first entity by running the Authentication and Key Agreement procedure; and the second parameter either comprises or is derived from a token having a different value each time the Authentication and Key Agreement procedure is initiated by the second entity for the first entity (in other words, the value of the token is never the same for any two Authentication and Key Agreement procedures); and applying a key derivation function to generate a cryptographic key based on the provided parameters.

[0025] The expression "a parameter comprises X" may mean that the variable X, in its string format, forms the parameter or a part thereof. The expression "a parameter is derived from X" may mean that the parameter is the result of applying certain functions, such as mathematical functions, to at least the variable X. Examples of the functions include, but are not limited to,

arithmetic operations, logic operations, string operations, and any combination thereof. The arithmetic operation may be addition, subtraction, multiplication, etc., and any meaningful combinations thereof.

[0026] The logic operation may be AND, OR, Exclusive OR (xOR), NOT, etc., and any meaningful combinations thereof. The string operation may be Concatenation, Reverse, Replace, etc., and any meaningful combinations thereof. Further, the arithmetic operation, the logic operation and the string operation may be combined.

[0027] Particularly, the token mentioned above may comprise or be derived from a sequence number (SQN) indicating the number of times that the Authentication and Key Agreement procedure has been initiated by the second entity for the first entity. With each initiation, the SQN may be incremented by the second entity. This mechanism ensures that the token has a different value for each Authentication and Key Agreement procedure initiated.

[0028] The token can take many forms. In one case, the SQN itself may be the token. Alternatively, the token may be derived from the SQN using an algorithm involving certain mathematical operations, such as at least one of an arithmetic operation, a logic operation and a string operation. For instance, the token may comprise or be derived from an Authentication Token (AUTN) constructed by the second entity based on the SQN and delivered to the first entity. This construction and delivery may be part of the Authentication and Key Agreement procedure.

[0029] Specifically, the token may comprise an exclusive OR of the SQN and an Anonymity Key (AK). More specifically, the token may be a concatenation of the exclusive OR of the SQN and the Anonymity Key (AK), an Authentication and Key Management Field (AMF), and a Message Authentication Code (MAC). This concatenation may be expressed as

token = AUTN = (SQN xOR AK) || AMF || MAC

or

token = function (AUTN) = function ((SQN xOR AK) || AMF || MAC)

[0030] The second parameter may further comprise or be derived from a random challenge, or random code (RAND). The RAND may be generated by the second entity and delivered to the first entity as part of the Authentication and Key Agreement procedure. The second parameter may yet further comprise or be derived from an identifier of the first entity. This identifier may be a Private User Identity (IMPI) or an International Mobile Subscriber Identity (IMSI). Even further, the second parameter may comprise or be derived from an identifier of a communications network and particularly the serving network of the first entity. For example, this identifier could be a Public Land Mobile Network Identifier (PLMN_ID).

[0031] Specifically, the second parameter may comprise or be derived from a concatenation of 0x02, a PLMN_ID, a RAND, an IMPI or IMSI, and the token. This could be expressed as

0x02 || PLMN_ID || RAND || IMPI || token.

[0032] When the token is the SQN itself, the above becomes
0x02 || PLMN_ID || RAND || IMPI || SQN;
and when the token is the AUTN, the above becomes

0x02 || PLMN_ID || RAND || IMPI || AUTN.

[0033] With respect to the first parameter used in the method, this parameter comprises or is derived from a set of cryptographic keys which have been obtained by the first entity by running the Authentication and Key Agreement procedure. The set of cryptographic keys may comprise or be derived from a Cipher Key (CK) and an Integrity Key (IK).

[0034] The CK and IK may be the cipher key and integrity key computed by the first entity based on an AUTN and an RAND. The AUTN and the RAND may be delivered from the second entity. This computation as well as the delivery of the AUTN and the RAND may form parts of the Authentication and Key Agreement procedure.

[0035] In one implementation, the first parameter may comprise or be derived from a concatenation of CK and IK. This may be mathematically expressed as
CK || IK.

[0036] The method described herein generates a cryptographic key. This key may be shared at least by the first entity and the second entity, in any subsequent communication there between. In certain implementations, this key may be the K_{ASME} referred to in the "key hierarchy" of Fig. 2, which may be shared by the first entity and an Access Security Management Entity (ASME) of the second entity.

[0037] The method may be extended to comprise applying one or more further key derivation functions so as to generate more cryptographic keys. Such generation is based on, or makes use of, the cryptographic key generated in the basic, unextended method described above, e.g. K_{ASME}.

[0038] The cryptographic keys generated by the extended method may include at least one of a set of cryptographic keys for protecting the Non-Access Stratum (NAS) traffic; a set of cryptographic keys for the protection of Radio Resource Control (RRC) traffic; a set of cryptographic keys for the protection of User Plane (UP) traffic; and an intermediate cryptographic key, such as K_{eNB}, for deriving the cryptographic keys for protecting the RRC traffic and/or the cryptographic keys for protecting the UP traffic. For an easier understanding of these keys, reference is made to Fig. 2 which illustrates the key hierarchy used in SAE/LTE.

[0039] Specifically, the set of cryptographic keys for protecting the NAS traffic may comprise a key for protecting the NAS traffic with an encryption algorithm (K_{NASenc}) and/or another key for protecting the NAS traffic with an integrity algorithm (K_{NASint}). Similarly, the set of cryptographic keys for the protection of RRC traffic may comprise a key for protecting the RRC traffic with an encryption algorithm (K_{RRCenc}) and/or another key for protecting the RRC traffic with an integrity algorithm (K_{RRCint}). Further, the set of cryptographic keys for the protection of UP traffic may comprise a key for protecting the UP traffic with an encryption algorithm (K_{UPenc}).

[0040] For the technique described herein, the "first entity" may be a user equipment, such as a mobile station. The "second entity" may be an entity located within a communications network, hence a "network entity". Particularly, the second entity may be located in a SAE/LTE network.

[0041] The second entity may comprise an Authentication Center (AuC)/Home Subscriber Server (HSS) and a Mobility Management Entity (MME). The MME may be responsible for the initiation of the Authentication and Key Agreement procedure for the first entity. The cryptographic keys generated may be generated by the AuC/HSS and be shared by the first entity and the MME. The AuC/HSS may increment the SQN, particularly each time the Authentication and Key Agreement procedure is initiated for the first entity. Further, the AuC/HSS may also construct the AUTN based on the SQN.

[0042] The Authentication and Key Agreement procedure referred to herein may be performed by the first and second entities in a cooperative manner. For instance, the Authentication and Key Agreement procedure may be based on the UMTS AKA protocol.

[0043] The key derivation function referred to by the method may be a Generic Bootstrapping Architecture (GBA) key derivation function. A Generic Bootstrapping Architecture key derivation function may employ a Secure Hash Algorithm (SHA) hash function. In particular, a Secure Hash Algorithm hash function with a digest of a length of 256 bits (SHA-256) may be employed.

[0044] According to another aspect, a computer program product is provided. The computer program product comprises program code portions for performing the steps of the method described herein when the computer program product is executed on a computer system for a computing device. The computer program product may be stored on a computer-readable reporting medium.

[0045] In general, the solution can be practiced by means of hardware, software, or a combined hardware/software approach.

[0046] As for a hardware realization, a device adapted to generate a cryptographic key for a communications entity is provided. The device embodies, among others, the following features: The device can perform an Authentication and Key Agreement procedure, of which the

generation of the cryptographic key may be a part thereof. The device comprises a first component adapted to provide at least two parameters, wherein the first parameter may comprise or be derived from a set of cryptographic keys having been computed by the communications entity by running the Authentication and Key Agreement procedure, and the second parameter may comprise or be derived from a token having a different value each time the Authentication and Key Agreement procedure is initiated for the communications entity. The device further comprises a second component adapted to execute a key derivation function so as to generate a cryptographic key based on the provided parameters. As said above, the token may take many possible forms.

[0047] The token may comprise or be derived from a SQN indicating the number of times the Authentication and Key Agreement procedure has been initiated for the communications entity. In one implementation, the SQN itself is the token. Alternatively, the token may be derived from the SQN using an algorithm involving at least one of arithmetic operation, logic operation and string operation. For instance, the token may comprise or be derived from an AUTN that is constructed based on the SQN and delivered to the communications entity, wherein this construction and delivery form parts of the security operation. For instance, the token may be a concatenation of the Exclusive-OR of the SQN and an Anonymity Key (AK), an Authentication and Key Management Field (AMF), and a Message Authentication Code (MAC). Specifically, this may be expressed as

token = AUTN = (SQN xOR AK) || AMF || MAC.

[0048] In addition to the token, the second parameter may also comprise or be derived from a RAND. The RAND may be delivered to the communications entity as part of the security operation. Further, the second parameter may comprise or be derived from an identifier of the communications entity. An example of the identifier is a Private User Identity (IMPI) of the communications entity. Even further, the second parameter may comprise or be derived from an identifier of the serving network of the communications entity. This identifier could be a Public Land Mobile Network Identifier (PLMN_ID).

[0049] A particular example of the second parameter may comprise or be derived from a concatenation of 0x02, a PLMN_ID, a RAND, an IMPI or an IMSI, and the token. For instance, the second parameter may be expressed as

0x02 || PLMN_ID || RAND || IMPI || token.

[0050] When token is the SQN, the above becomes

0x02 || PLMN_ID || RAND || IMPI || SQN;

and when the token is AUTN, the above becomes

0x02 || PLMN_ID || RAND || IMPI || AUTN.

[0051] As mentioned above, the first parameter may comprise or be derived from a set of cryptographic keys. Particularly, this set of cryptographic keys may comprise a Cipher Key (CK) and an Integrity Key (IK) which have been computed by the communications entity as part of the security operation. Alternatively, the set of cryptographic keys may be derived from the Cipher Key and the Integrity Key.

[0052] As a particular implementation, the first parameter may comprise or be derived from a concatenation of CK and IK, which may be expressed as
CK || IK.

[0053] The device can generate not only the cryptographic key based on the provided first and second parameters, but also more cryptographic keys based on the cryptographic key generated. In doing so, the device may be adapted to apply one or more further key derivation functions so as to generate the more cryptographic keys based on the cryptographic key having been generated.

[0054] These "more cryptographic keys" may comprise at least one of a set of cryptographic keys for the protection of Non-Access Stratum (NAS) traffic, a set of cryptographic keys for the protection of Radio Resource Control (RRC) traffic, a set of cryptographic keys for the protection of User Plane (UP) traffic, and an intermediate cryptographic key K_{eNB} for deriving the cryptographic keys for the protection of RRC traffic and/or the cryptographic keys for the protection of UP traffic.

[0055] The communications entity referred to above may be a user equipment, such as a mobile station (e. g., a mobile telephone or a network card).

[0056] According to a further aspect, a user equipment comprising the device presented above is provided. The user equipment may be a mobile station.

[0057] According to yet a further aspect, a system comprising the user equipment mentioned above is provided. The system also comprises a network entity. The network entity may be used within a SAE/LTE network. The network entity may comprise an AuC/HSS and a MME. The MME may be responsible for initiating the security operation for the user equipment. The AuC/HSS may generate the cryptographic key. The cryptographic keys generated may be shared by the user equipment and the MME. The AuC/HSS may increment the SQN, particularly each time the security operation is initiated for the user equipment. Further, the AuC/HSS may also construct the AUTN based on the SQN.

Brief Description of the Drawings

[0058] In the following, the cryptographic key generation technique will be described with

reference to exemplary embodiments illustrated in the drawings, wherein:

Fig. 1
is a diagram showing the basic concept of the UMTS AKA protocol;
Fig. 2
is a block diagram illustrating a key hierarchy proposed for SAE/LTE system;
Fig. 3
is a block diagram showing a device embodiment;
Fig. 4
is a block diagram showing a system embodiment;
Fig. 5
is a block diagram showing a method embodiment;
Fig. 6
is a block diagram showing a procedure of the UMTS AKA operation, Generation of an Authentication Vector by a network entity;
Fig. 7
is a block diagram showing another procedure of the UMTS AKA operation, Authentication and Key Establishment;
Fig. 8
is a block diagram showing the general authentication function performed by the UE as part of the UMTS AKA operation;
Fig. 9
is a block diagram showing a particular cryptographic algorithm for performing the above authentication function at the UE; and
Fig. 10
is a block diagram showing a particular detail of the above cryptographic algorithm.

Detailed Description

[0059] In the following description, for purposes of explanation and not limitation, specific details are set forth, such as particular sequences of steps, interfaces and configurations, in order to provide a thorough understanding of the cryptographic key generation technique. It will be apparent to those skilled in the art that the technique may be practiced in other embodiments that depart from these specific details. For example, while the technique will primarily be described in context with the UMTS AKA protocol and in the SAE/LTE network environment, it will be apparent to the skilled person that the technique can also be practiced in connection with other security protocols, architectures, or environments.

[0060] Moreover, those skilled in the art will appreciate that the functions explained herein below may be implemented using software functioning in conjunction with a programmed microprocessor or general purpose computer. It will also be appreciated that while the

technique is primarily described in the form of methods and devices, the technique may also be embedded in a computer program product as well as in a system comprising a computer processor and a memory coupled to the processor, wherein the memory is encoded with one or more programs that may perform the function disclosed herein.

[0061] Fig. 3 shows an embodiment of a device 100 adapted to generate a cryptographic key for a communications entity (not shown in Fig. 3). The communications entity is adapted to run a security operation. The device 100 comprises a first component 102 and a second component 104. The first component 102 is adapted to provide at least two parameters, figuratively shown at the arrows 106 and 108.

[0062] The first parameter 106 comprises or is derived from a set of cryptographic keys 110 and 112. (Although two keys are shown in the figure, the set of cryptographic keys may include any number of keys.) The set of cryptographic keys has been computed by the communications entity by running the security operation. The derivation of the set of cryptographic keys 110 and 112 into the first parameter 106 is figuratively shown as a block 114. The second parameter 108 comprises or is derived from a token 116. The token 116 has a different value each time the security operation is initiated for the communications entity. The derivation of the token 116 into the second parameter 108 is figuratively shown as a block 118. The second component 104 of the device 100 is adapted to run a key derivation function to generate a cryptographic key 120 based on the provided parameters 106 and 108.

[0063] Referring to Fig. 4, an embodiment of a system 200 comprising the device 100 mentioned above is shown. The device 100 may be comprised in a communications entity 202, which may be a UE, such as a mobile station. Of course, the communications entity 202 may be any suitable kind of communications entity capable of accommodating the device 100. Further, the system comprises a network entity 204, which may reside in a SAE/LTE network. The network entity 204 may comprise an AuC or HSS and a MME. It may also be another communications entity in a SAE/LTE network.

[0064] Corresponding to the cryptographic key generation device 100 shown in Figs. 3 and 4, a diagram 300 illustrating an embodiment of a method for generating a cryptographic key is shown in Fig. 5. The key generated is used for protecting the communication between two entities. The first entity 302 may correspond to the communications entity 202 as shown in Fig. 4, and the second entity 304 may correspond to the network entity 204 of Fig. 4. The first entity may be a UE. However, the embodiment is not limited to a UE-network entity scenario. Instead, it can be applied to any two communications entities in general.

[0065] The MME may be responsible for initiating the security operation for the communications entity 202. The cryptographic keys generated may be shared by the MME and the communications entity 202.

[0066] Particularly, the method embodiment is carried out by the first entity 302 as part of a security operation figuratively illustrated at the arrow 300', which is initiated by the second

entity 304 (particularly by the MME thereof) for the first entity 302. The embodiment itself comprises two steps, 306 and 308. Step 306 provides at least two parameters (106 and 108 of Fig. 3). The first parameter comprises or is derived from a set of cryptographic keys (110 and 112 as shown in Fig. 3) which have been computed by the first entity 302 by running the security operation 300'. The second parameter comprises or is derived from a token (116 as shown in Fig. 3) which has a different value each time the security operation 300' is initiated by the second entity 304 for the first entity 302. At the second step 308, a key derivation function is applied to generate a cryptographic key (120 as shown in Fig. 3) based on the provided parameters (106 and 108 as shown in Fig. 3).

[0067] Below, substantial details are given to explain the cryptographic key generation technique with a particular emphasis on how the technique can successfully avoid the key-collisions between two UEs, or more importantly, between two distinct executions of the security operation for one and the same UE.

[0068] The cryptographic key generation may be part of the UMTS AKA operation. The UMTS AKA is based on the implementation that the UE, particularly the USIM thereof, and the AuC/HSS in the UE's Home Environment (HE) share a user specific secret key K, certain message authentication functions f1, f2 and certain cryptographic key generation functions f3, f4, f5. In addition the USIM and the AuC/HSS keep track of counters, or sequence numbers SQN_{UE} and SQN_{HE} respectively to support network authentication. For instance, the AuC/HSS may increment the SQN_{HE}, particularly each time the security operation is initiated for the first entity. The UMTS AKA operation comprises a number of procedures, including Generation of Authentication Vectors (AV), and Authentication and Key Establishment.

[0069] The purpose of the AV procedure is to provide the SN/VLR (or MME) with an array of fresh AVs from the UE's HE to perform a number of user authentications. Generation of Authentication Vectors by the HE is illustrated in Fig. 6. Referring to this figure, upon receipt of a request from the SN/VLR, the AuC/HSS sends an ordered array of n Authentication Vectors AV (1...n) to the SN/VLR. Each AV comprises a random number (or random challenge) RAND, an expected response XRES, a cipher key CK, an integrity key IK and an authentication token AUTN.

[0070] The AuC/HSS starts with generating a fresh sequence number SQN and an unpredictable challenge RAND. Subsequently the following values are computed:

- a message authentication code MAC = f1(SQN ‖ RAND ‖ AMF) where f1 is a message authentication function;
- an expected response XRES = f2 (RAND) where f2 is a (possibly truncated) message authentication function;
- a cipher key CK = f3 (RAND) where f3 is a key generating function;
- an integrity key IK = f4 (RAND) where f4 is a key generating function; and
- an anonymity key AK = f5 (RAND) where f5 is a key generating function.

[0071] Finally the authentication token AUTN = (SQN xOR AK) ‖ AMF ‖ MAC is constructed. It may be constructed by the AuC/HSS. Here, AK is an anonymity key used to conceal the SQN as the latter may expose the identity and location of the UE. The concealment of the SQN is to protect against passive attacks. Use of AK may be optional. When AK is not used, the value AK = 000...0 may figuratively be used instead.

[0072] The array of AVs is sent back to the requesting SN/VLR in an authentication response. Each AV is valid for one (and only one) authentication and key agreement between the SN/VLR and the USIM.

[0073] The next procedure of the UMTS AKA operation, Authentication and Key Establishment, is to mutually authenticate and establish new cipher and integrity keys between the SN/VLR and the UE. This process is illustrated in Fig. 7. Referring to this figure, when the SN/VLR initiates an authentication and key agreement, it selects the next AV from the array and sends the parameters RAND and AUTN to the UE. The USIM checks whether AUTN can be accepted and, if so, produces a response RES which is sent back to the SN/VLR. Particularly, the UE's procedures are shown in Fig. 8.

[0074] Referring to Fig. 8, upon receipt of RAND and AUTN the UE first computes the anonymity key AK = f5 (RAND) (or uses AK = 000...0) and retrieves the sequence number SQN = (SQN xOR AK) xOR AK. Next the UE computes XMAC = f1 (SQN ‖ RAND ‖ AMF) and compares this with MAC which is included in AUTN. If they are different, the UE sends *user authentication reject* back to the SN/VLR with an indication of the cause and the UE abandons the procedure. Else, the UE verifies that the received SQN is in the correct range.

[0075] If the SQN is considered to be in the correct range, the UE computes RES = f2 (RAND) and includes this parameter in a *user authentication response* back to the SN/VLR. Finally the UE computes the cipher key CK = f3 (RAND) and the integrity key IK = f4 (RAND). To improve efficiency, RES, CK and IK could also be computed earlier at any time after receiving RAND. The UE may store RAND for re-synchronization purposes.

[0076] Upon receipt of user authentication response the SN/VLR compares RES with the expected response XRES from the selected authentication vector. If XRES equals RES then the authentication of the user has been accepted. The newly computed keys CK and IK will then be transferred by the USIM and the SN/VLR to the entities which perform ciphering and integrity functions.

[0077] From the above, it can be seen that the UMTS AKA operation is based on a pair (RAND, AUTN) and AUTN comprises or is derived from a sequence number, SQN, as

$$AUTN = (SQN \text{ xOR } AK) \,||\, AMF \,||\, MAC$$

where AK is an anonymity key, which may be produced by Milenage (see Fig. 9) from output "f5" above.

[0078] The below function is a first solution to the collision problem set out above:
KDF(CK || IK, RAND || IMPI || SQN)
where SQN has thus been included in the inputs. Now, even if two RANDs are the same, i.e., RAND = RAND', the fact that SQN always increases (by e.g., one) will ensure that inputs are different, unique, or distinct.

[0079] An alternative solution is to use
KDF(CK || IK, RAND || IMPI || AUTN).

[0080] This solution may be simpler to implement since AUTN can be used "as is" from the AKA signaling. However, the "uniqueness'" of the inputs in this case may not be obvious since
AUTN = (SQN xOR AK) || AMF || MAC
and even if SQN ≠ SQN', it cannot be immediately seen that (SQN xOR AK), (SQN' xOR AK') will be distinct as AK could potentially "cancel" the differences. However, below, the distinctness of (SQN xOR AK) can be proven.

[0081] Suppose that
(CK || IK, RAND || IMPI || AUTN) = (CK' || IK', RAND' || IMPI || AUTN').

[0082] It has already been shown that this implies CK = CK', IK = IK', and RAND = RAND'. It thus remains to be checked if it could be that AUTN = AUTN'. This checking may be translated into checking if
(SQN xOR AK) || AMF || MAC = (SQN' xOR AK') || AMF' || MAC'.

[0083] Assume without loss of generality that AMF = AMF' and MAC = MAC'. Then it is only necessary to check if the following could hold:
SQN xOR AK = SQN' xOR AK'.

[0084] Recall that it is expected that RAND = RAND'. Referring to the Milenage algorithm shown in Fig. 9, this implies that AK= AK' (as they were produced from the same RANDs). Thus, it had to be that
SQN = SQN',
which is a contradiction since, as already noted, SQN always "steps up" and thus SQN ≠ SQN'.

[0085] Thus, it is proven the second solution also guarantees the uniqueness of inputs to the KDF function.

[0086] As a general solution, instead of using SQN or AUTN to achieve the uniqueness, any token having a different value each time the UMTS AKA operation is initiated by the network for the UE is feasible. For instance, SQN xOR AK (forming part of AUTN) may be used since it (by

the above analysis) has the required uniqueness property.

[0087] The cryptographic key generation technique described here above presents numerous advantages. For example, it guarantees uniqueness of KDF inputs. Hence, it successfully avoids the commissions brought about by possible identical inputs. With this technique, the cryptographic key generated shall be able to meet, for example, the high-level security requirements in SAE/LTE systems. As a further advantage, the technique can be implemented based on already deployed USIMs without requiring any USIM replacement. Another specific advantage with using AUTN rather than SQN is that the invention can be implemented in the mobile terminal (outside the USIM).

[0088] Although embodiments of the cryptographic key generation technique have been illustrated in the accompanying drawings and described in a foregoing description, it will be understood that the technique is not limited to the embodiments disclosed herein. The technique is capable of numerous re-arrangements, modifications and substitutions without departing from the scope of the invention.

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO2005032201A [0022]
- WO2004075584A [0022]

GENERERING AF KRYPTERINGSNØGLE

PATENTKRAV

1. Fremgangsmåde til generering af en krypteringsnøgle (120) til beskyttelse af mobilkommunikation mellem to enheder (202, 204), hvor fremgangsmåden udføres af den første enhed (202, 302) som en del af en godkendelses- og nøgleoverensstemmelses- (AKA for Authentication and Key Agreement) procedure baseret på en UMTS AKA-protokol, der initieres af den anden enhed (204, 304), hvilken fremgangsmåde er kendetegnet ved trinnene med:

- tilvejebringelse (306) af mindst to parametre (106, 108), hvor den første parameter (106) omfatter eller er afledt af et sæt af krypteringsnøgler (110, 112), der er beregnet af den første enhed (202) ved at køre AKA-proceduren, og den anden parameter omfatter eller er afledt af et token (116) med en forskellig værdi, hver gang AKA-proceduren initieres af den anden enhed (204, 304) for den første enhed (202, 302); og
- anvendelse (308) af en nøgleafledningsfunktion til at generere en krypteringsnøgle (120) baseret på de tilvejebragte parametre (106, 108);

hvor tokenet (116) er en sammenkædning af det eksklusive ELLER af et sekvensnummer <SQN> og en anonymitetsnøgle <AK>, et godkendelses- og nøgleforvaltningsfelt <AMF> og en kode for meddelelsesgodkendelse <MAC>, hvor SQN indikerer det antal gange, AKA-proceduren er blevet initieret af den anden enhed (204, 304) for den første enhed (202, 302), og hvor AK er en krypteringsnøgle, der er frembragt ved hjælp af en nøglegenereringsfunktion f5 ved anvendelse af en vilkårlig udfordring i henhold til UMTS AKA-protokollen.

2. Fremgangsmåde ifølge et hvilket som helst af de foregående krav, hvor sættet af krypteringsnøgler (110, 112), der indgår i den første parameter (106), eller hvoraf den første parameter (106) er afledt, omfatter eller er afledt af en talnøgle (Cipher Key) <CK> (110) og en integritetsnøgle (Integrity Key) <IK> (112).

3. Fremgangsmåde ifølge et hvilket som helst af de foregående krav, der endvidere omfatter trinnet med:

- anvendelse af én eller flere yderligere nøgleafledningsfunktioner til at generere flere krypteringsnøgler baseret på den genererede krypteringsnøgle (120).

4. Fremgangsmåde ifølge krav 3, hvor de flere krypteringsnøgler omfatter mindst én af følgende:

- et sæt af krypteringsnøgler til beskyttelse af Non Access Stratum- <NAS> trafik;
- et sæt af krypteringsnøgler til beskyttelse af radiokapacitetsstyrings- (Radio Resource Control) <RRC> trafik;
- et sæt af krypteringsnøgler til beskyttelse af brugerplans- (User Plane) <UP> trafik og
- en mellemliggende krypteringsnøgle <K_{eNB}> til afledning af krypteringsnøglerne til beskyttelse af RRC-trafik og/eller krypteringsnøglerne til beskyttelse af UP-trafik.

5. Fremgangsmåde ifølge et hvilket som helst af de foregående krav, hvor den første enhed (202, 302) er et brugerudstyr.

6. Fremgangsmåde ifølge et hvilket som helst af de foregående krav, hvor den anden enhed (204, 304) er en netværksenhed.

7. Fremgangsmåde ifølge krav 6, hvor den anden enhed (204, 304) ligger i et System Architecture Evolution- <SAE>/Long Term Evolution- <LTE> netværk.

8. Fremgangsmåde ifølge krav 6 eller 7, hvor den anden enhed (204, 304) omfatter et/en Authentication Center <AuC>/Home Subscriber Server <HSS> og en Mobility Management Entity <MME>.

9. Fremgangsmåde ifølge et hvilket som helst af de foregående krav, hvor godkendelses- og nøgleoverensstemmelsesproceduren sammen udføres af første (202, 302) og anden (204, 304) enheder.

10. Computerprogramprodukt kendetegnet ved computerprogramkodedele til udførelse af trinnene med fremgangsmåden ifølge et hvilket som helst af de foregående krav, når computerprogramproduktet køres på et computersystem.

11. Computerprogramprodukt ifølge krav 10, hvor computerprogramproduktet lagres på et computerlæsbart registreringsmedie.

12. Anordning (100), der er tilpasset til at generere en krypteringsnøgle (120) til en mobilkommunikationsenhed (202, 302), der er tilpasset til at køre en godkendelses- og nøgleoverensstemmelsesprocedure baseret på en UMTS AKA-protokol, hvilken anordning (100) er kendetegnet ved:

- en første komponent (102), der er tilpasset til at tilvejebringe mindst to parametre (106, 108), hvor den første parameter (106) omfatter eller er afledt af et sæt af krypteringsnøgler (110, 112), der er blevet beregnet af den mobile kommunikationsenhed (202, 302) ved at køre AKA-proceduren, og den anden parameter (108) omfatter eller er afledt af et token (116), hvilke mindst to tilvejebragte parametre har en forskellig værdi, hver gang AKA-proceduren initieres for den mobile kommunikationsenhed (202, 302); og
- en anden komponent (104), der er tilpasset til at køre en nøgleafledningsfunktion til at generere en krypteringsnøgle (120) baseret på de tilvejebragte parametre (106, 108);

hvor tokenet (116) en sammenkædning af det eksklusive ELLER af et sekvensnummer <SQN> og en anonymitetsnøgle <AK>, et godkendelses- og nøgleforvaltningsfelt <AMF> og en kode for meddelelsesgodkendelse <MAC>, hvor SQN indikerer antallet af gange AKA-proceduren er blevet initieret for den mobile kommunikationsenhed (202, 302), og hvor AK er en krypteringsnøgle frembragt af en nøglegenereringsfunktion f5 ved anvendelse af en vilkårlig udfordring i henhold til UMTS AKA-protokollen.

13. Anordning (100) ifølge krav 12, hvor sættet af krypteringsnøgler (110, 112), der indgår i den første parameter (106), eller hvoraf den første parameter (106) er afledt, omfatter eller er afledt af en talnøgle <CK> (110) og en integritetsnøgle <IK> (112) beregnet af den mobile kommunikationsenhed (202, 302) som en del af godkendelses- og nøgleoverensstemmelsesproceduren.

14. Anordning (100) ifølge et hvilket som helst af kravene 12 til 13, der endvidere er tilpasset til at anvende én eller flere yderligere nøgleafledningsfunktioner til at generere flere krypteringsnøgler baseret på den genererede krypteringsnøgle (120).

15. Brugerudstyr (202) kendetegnet ved anordningen (100) ifølge et hvilket som helst af kravene 12 til 14.

16. System omfattende brugerudstyret (202, 302) ifølge krav 15 og en netværksenhed (304).

17. System ifølge krav 16, hvor netværksenheden (304) er beregnet til anvendelse i et SAE/LTE-netværk.

DRAWINGS

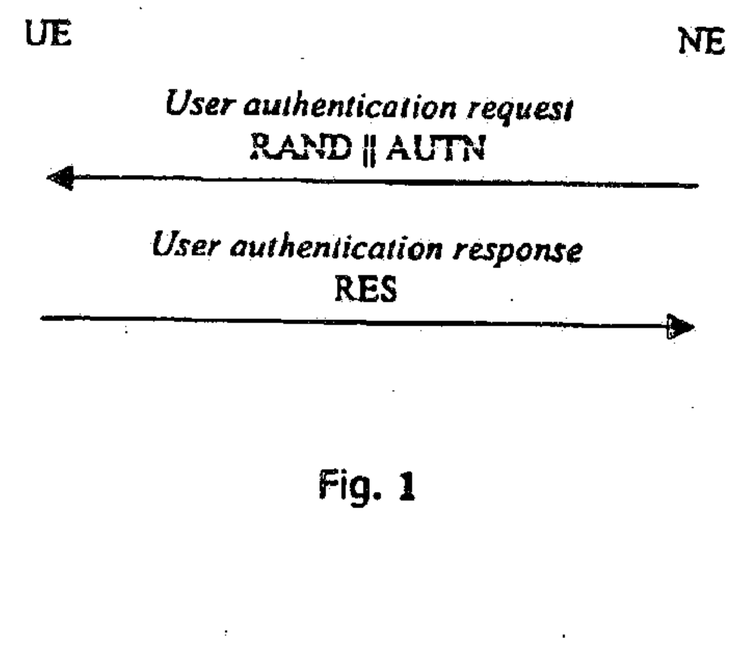


Fig. 1

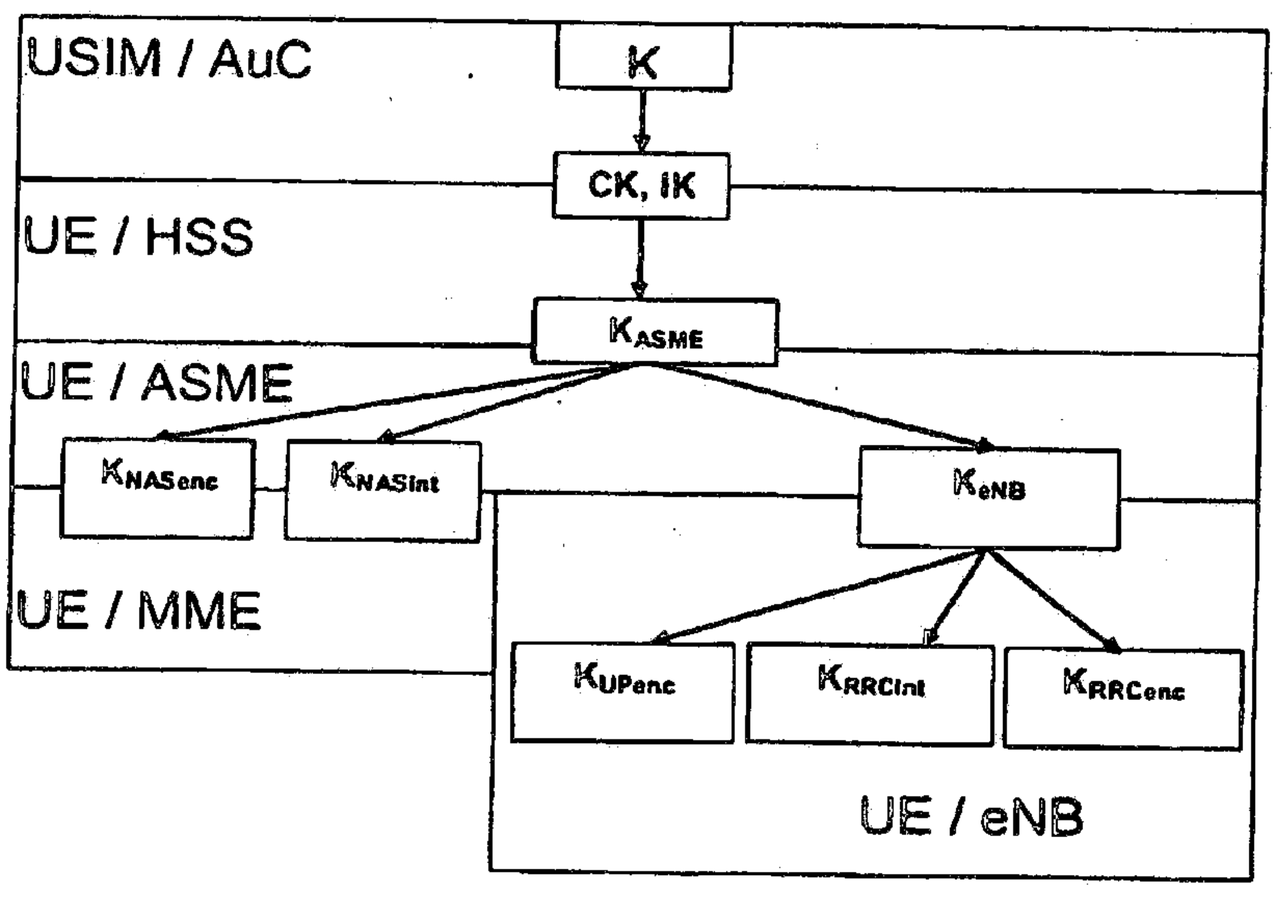


Fig. 2

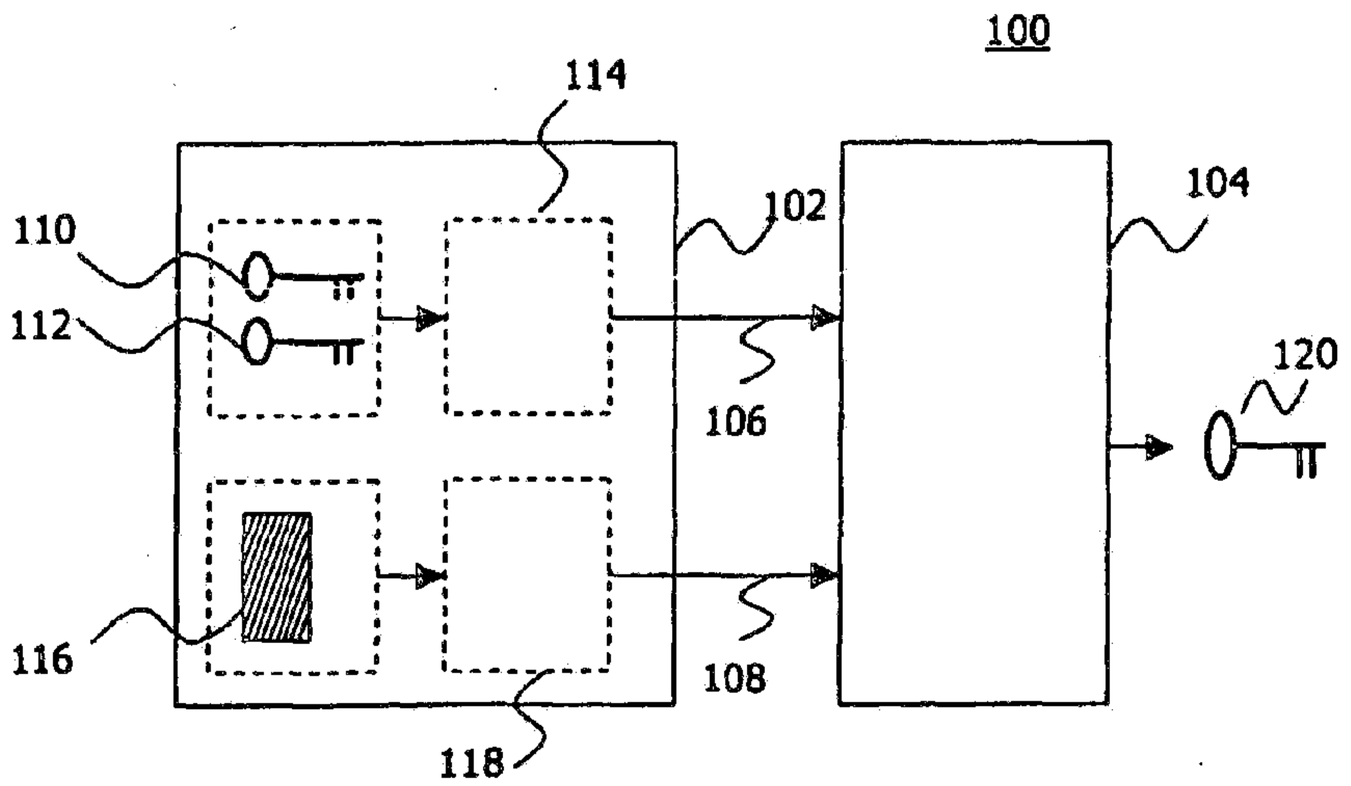


Fig. 3

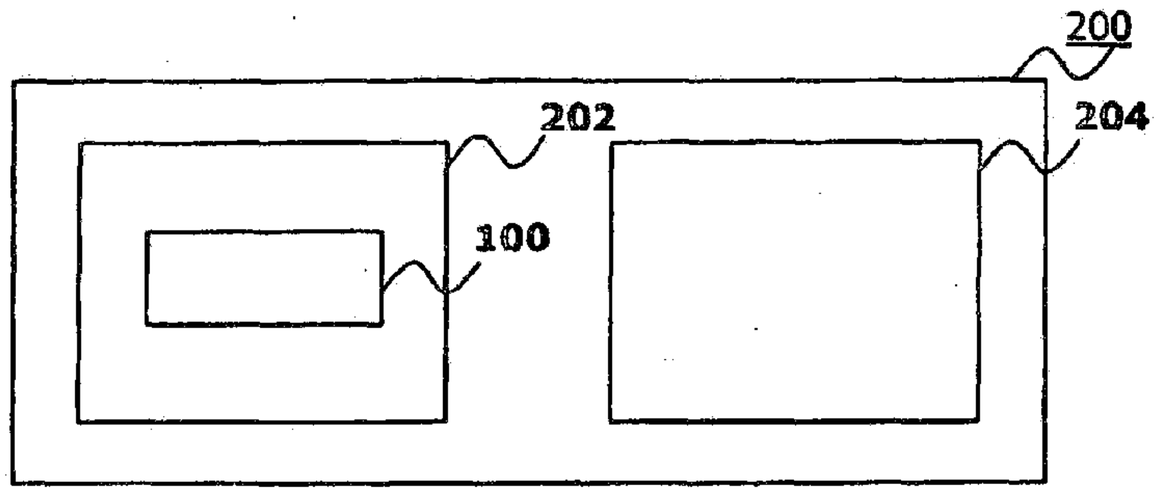


Fig. 4

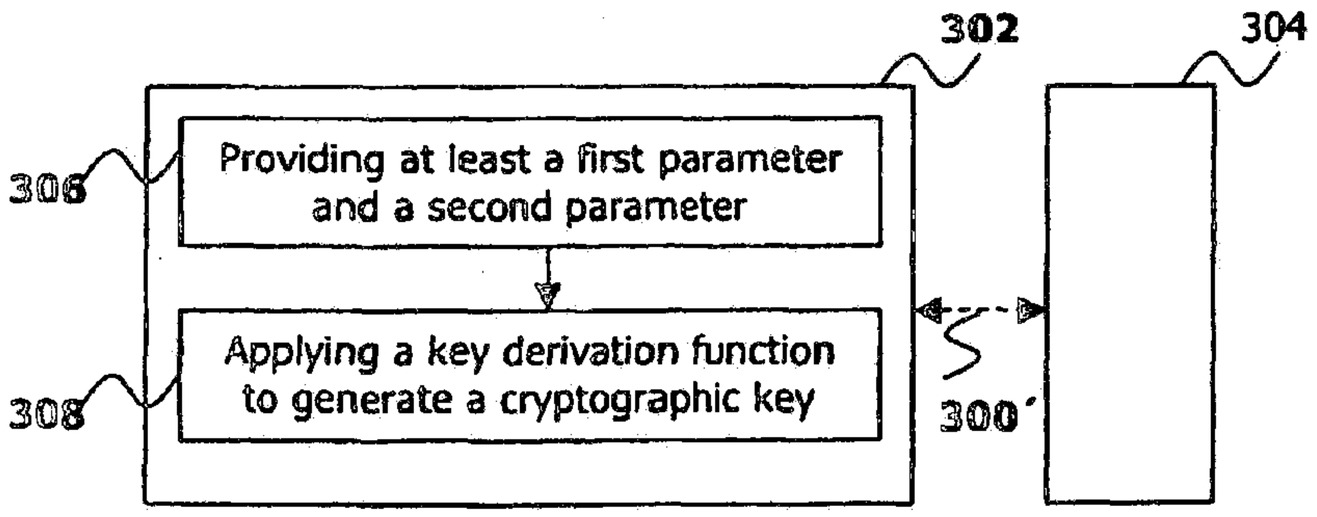
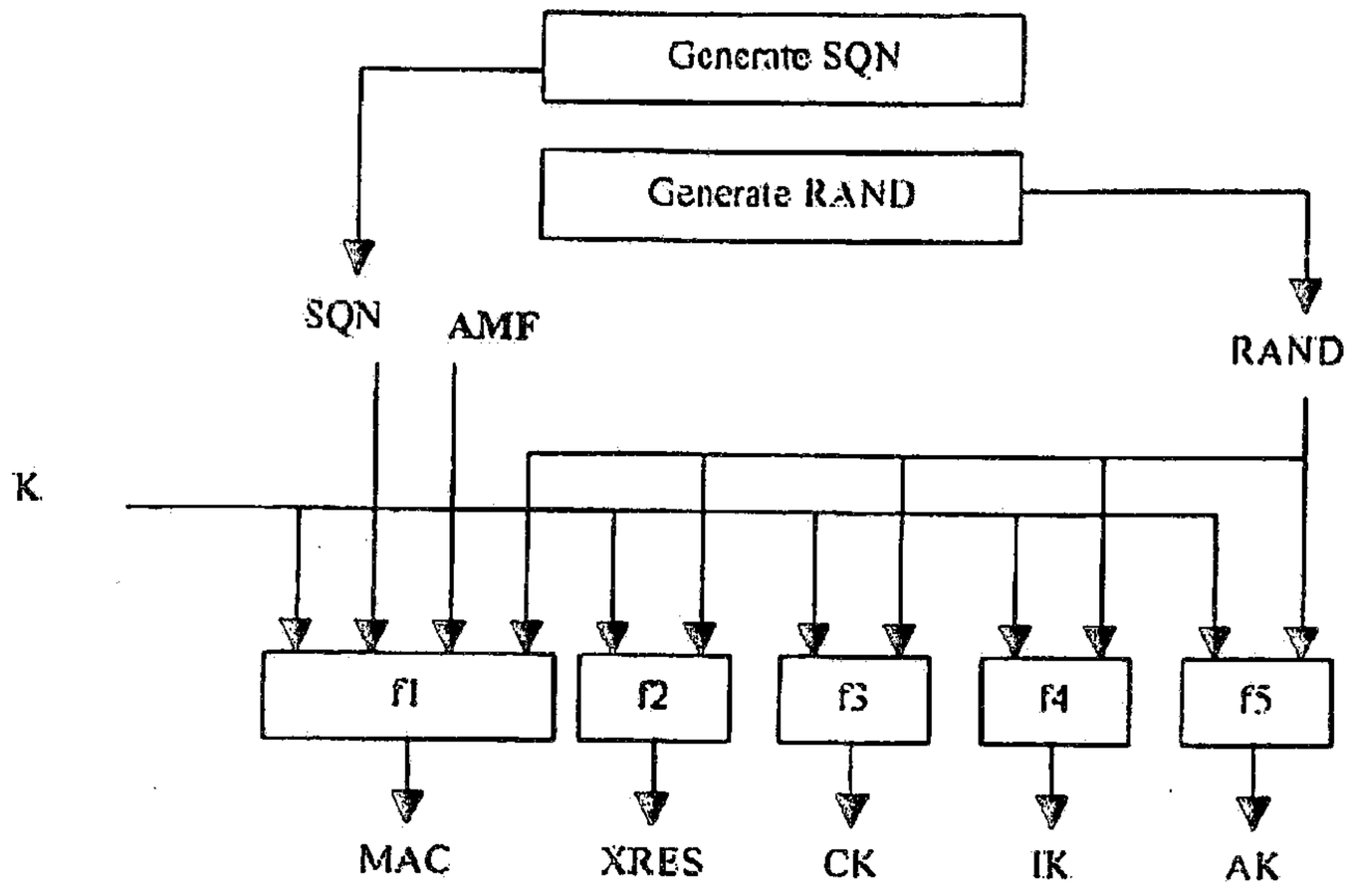


Fig. 5

300



AUTN = (SQN xOR AK) || AMF || MAC
AV = RAND || XRES || CK || IK || AUTN

Fig. 6

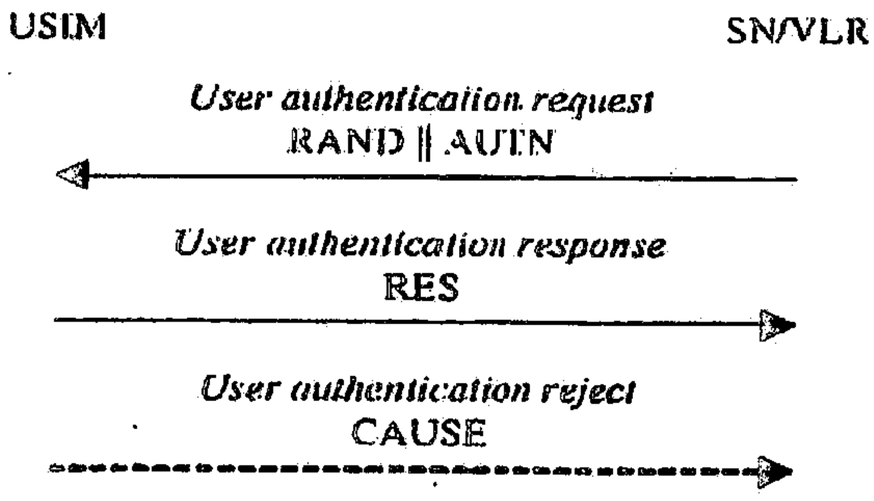
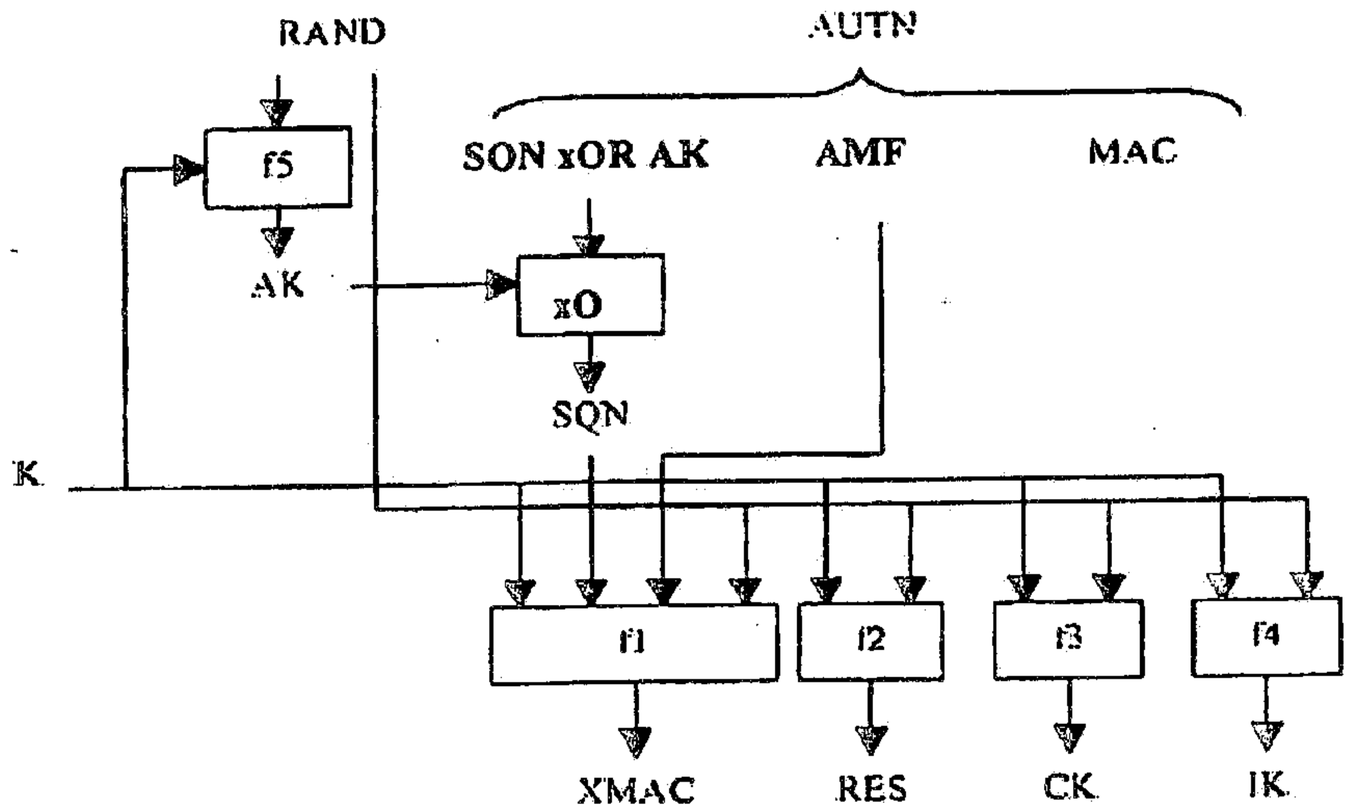


Fig. 7



Verify MAC = XMAC

Verify SQN > SQN_{HE}

Fig. 8

Fig. 9

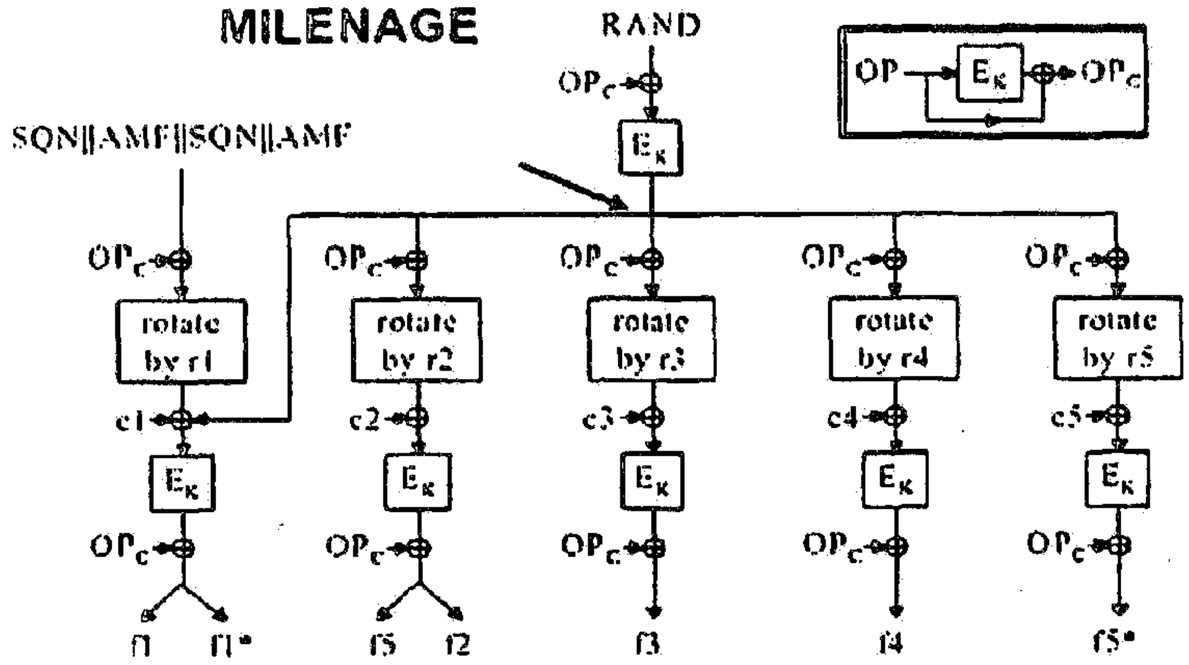
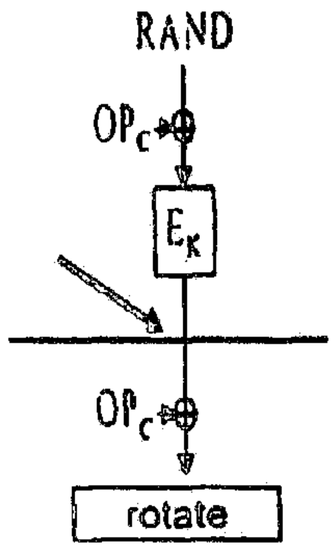

MILENAGE
RAND
OP
E_K
OP_C
SQN||AMF||SQN||AMF
rotate by r1
rotate by r2
rotate by r3
rotate by r4
rotate by r5
c1
c2
c3
c4
c5
f1
f1*
f5
f2
f3
f4
f5*

Fig. 10


RAND
OP_C
E_K
rotate