DoS Attacks Exploiting Signaling in UMTS and IMS

[1]Georgios Kambourakis, [1]Constantinos Kolias, [1]Stefanos Gritzalis, [2]Jong Hyuk Park

[1]Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, GR-83200 Samos, Greece
Email: {gkamb, kkolias, sgritz}@aegean.gr

[2]Department of Computer Science and Engineering,
Seoul National University of Technology, Korea
Email: parkjonghyuk1@hotmail.com

Abstract – The Universal Mobile Telecommunication Standard (UMTS) is continuously evolving to meet the growing demand of modern mobile and Internet applications for high capacity and advanced features in security and quality of service. Although admittedly enhanced in terms of security when compared to 2G systems, UMTS still has weaknesses that can lead to security incidents. In this paper we investigate the vulnerabilities of the UMTS security architecture that can be exploited by a malicious individual to mount Denial of Service (DoS) attacks. Our focus is on signaling-oriented attacks above the physical layer. We describe and analyze several novel attacks that can be triggered against both core UMTS architecture as well as hybrid UMTS/WLAN realms. An additional contribution of this paper is the presentation of an extensive survey of similar attacks in UMTS and related protocol infrastructures such as IP Multimedia Subsystem (IMS). Finally, we offer some suggestions that would provide greater tolerance to the system against DoS attacks.

Keywords - UMTS; Denial of Service; Signaling; IMS; Security; EAP-AKA.

Note: This work is based in part on a paper presented at the ISA 2009, 3rd International Conference on Information Security and Assurance [1].

1. Introduction

Beyond doubt, handheld devices have changed the modern way of communication and information access. The increasing demand for high quality multimedia services along with the need for modern pervasive applications has given birth to the Universal Mobile Telecommunication System (UMTS). UMTS is the outcome of a collaborative effort of many international organizations gathered around the 3rd Generation Partnership Project (3GPP) consortium (http://www.3gpp.org/). Today, 3rd Generation (3G) mobile networks based on the UMTS standard are deployed in Europe and USA (3GPP2) with great success. Users of these networks benefit from the higher quality of voice and video calls, higher transfer rates, communication with the internet, and enjoy advance applications and value-added services such as e-commerce, e-banking etc. In the years to come, most people will use their handheld device to make wireless security-sensitive transactions like e-banking, stock trading, and shopping. Therefore, with the introduction of such new applications to the mobile world, security, now more than ever, is a crucial aspect. Nevertheless, the inherited weaknesses of the UMTS that derive mostly from its wireless nature and its compatibilities with Second Generation (2/2.5G) systems make it prone to a substantial number of security threats.

A Denial of Service (DoS) attack is the type of attack preformed by a malicious entity in order to render a service unavailable to its intended users. Numerous DoS incidents verify the catastrophic potential of this class of attacks, and several researchers characterize DoS attacks as the second most dangerous cyber crime threat after viruses. The methodology and target of a DoS attack may vary, spanning from simple packet injection, done by an amateur to suppress the quality of offered services, to professionally orchestrated distributed attacks able to paralyze entire network infrastructures. While this type of attacks has its roots on the Internet realm, its philosophy and purpose has derived to the GSM networks and lately to UMTS, since wireless communications offer a new challenging terrain for attackers.

The primary target of the designers of UMTS was to maintain maximum compatibility with the 2G systems. Additionally, its designers took into account the constraints in computational power of the various mobile devices, and for that reason, they adopted relatively lightweight security techniques such as symmetric encryption [2]. Thus, even though UMTS is characterized by many major security enhancements comparing to its 2G predecessor the GSM it still presents architectural weaknesses. These weaknesses render it vulnerable to several security threats against the confidentiality of its users and the integrity and availability of the offered services. Until now, the majority of research in UMTS has focused on ways to preserve the privacy and confidentiality of the end-users [3-5]. Although privacy and confidentiality are always of top priority in any wireless system, we believe that the availability of the services should not be neglected. Unfortunately, UMTS in its current form makes it quite easy for DoS attacks to be launched.

In this paper, we particularly concentrate on signaling–oriented DoS attacks that can affect both UMTS and UMTS/WLAN integrated systems. We mark down and analyze several novel attacks of this class and point out architectural and protocol vulnerabilities that can be exploited to unleash such situations. Also, where applicable, we give directions for possible improvements. Finally, we provide an overview of the literature for similar attacks launched against both the core UMTS as well as IP Multimedia Subsystem (IMS) components.

The remainder of the paper is structured as follows: the next section gives background information regarding UMTS security architecture. Section 3 points out certain UMTS system vulnerabilities and discusses requirements and methodologies that can be exploited by an aggressor to achieve DoS. Section 4 surveys previous work on the topic, while Section 5 offers our suggestions and gives pointers to future work. The last section concludes the paper.

2. UMTS security architecture

The UMTS security architecture defines a set of procedures that the user's mobile equipment as well as the network should execute in order to achieve increased message confidentiality and integrity during their communication. In the heart of the UMTS security architecture lies the user authentication mechanism known as Authentication and Key Agreement (AKA) [6]. This mechanism is somewhat similar to the authentication in GSM. The idea to use public keys in the process of authenticating the users, was abandoned, mainly due to backwards compatibility (with GSM) and for performance considerations. The authentication in UMTS is based on a 128-bit symmetric secret key, namely Ki, which is stored in the user’s tamper-resistant Universal Integrated Circuit Card (UICC) and in the corresponding Home Location Register (HLR) of the user’s Home Network (HN). The AKA scheme is a combination of the well known challenge response-protocol found in GSM and the authentication mechanism based on sequence number as defined by the ISO organization [7]. The network entities that take part in the user’s authentication procedure are:

- The User’s Equipment (UE) and more specifically the USIM application stored in the UICC.
- The Serving GPRS Support Node (SGSN) of the HN or the Serving Network (SN).
- The HLR of the user’s HN.

The authentication procedure in UMTS is mutual, which means that both the network is authenticated to the UE and the UE is authenticated to the network. After successful authentication the two ends agree on the use of two additional 128-bit symmetric keys. These keys are derived from the master key Ki and renewed every time the user is authenticated. The procedure typically initiates after the Mobile Station (MS) attaches to the network and sends its identity. Note, that the user can be identified either by a permanent ID, i.e., the International Mobile Subscriber Identity (IMSI) or, usually, a temporary one known as Temporary Mobile Subscriber Identity (TMSI). During the process, the user’s ID is forwarded from the Radio Access Network (RAN) sub-network to the core network, that is, the SGSN serving that particular area. In any case, the latter entity may send an Authentication Data Request message to the HLR of the user’s HN in order to acquire Authentication Vectors (AV) required to authenticate the user. This happens only in cases that no AV for that particular user is available locally in the SGSN. For instance, the user attaches for the first time to this SGSN or the available in the SGSN AVs for that user have been already consumed. Since the HLR possesses the master key (Ki) for each user *i* is capable of creating the corresponding AVs. The vectors are sent back to the SGSN in charge by making use of a control message known as Authentication Data Response. Each vector can be used only once except the case the SGSN does not receive an answer from the MS.

After the SGSN in charge acquires some AVs (they are sent usually in batch), it sends an Authentication Request to the user. The request contains two parameters: (a) a RAND which is a random number and, (b) the AUTN, i.e., the authentication token. These parameters are transferred in the tamper resistant environment of the UICC/USIM and stored for further processing.

The USIM is also aware of the Ki, and uses it along with the received parameters RAND and AUTN to perform a series of calculations similar to those that took place for the generation of the corresponding AV in the HN’s HLR. The outcome of this procedure enables USIM to verify that the AUTN parameter was indeed created by the HLR of the HN and also that it is fresh (i.e., it is not a message replay). In case that the above verifications have a positive outcome the RES (result) parameter is calculated and sent back to the corresponding SGSN by utilizing a User Authentication Response message. Upon that, the SGSN compares the received RES with the XRES (Expected Response) which is contained in the initial AV. If the two values match then the user is granted access to the network.

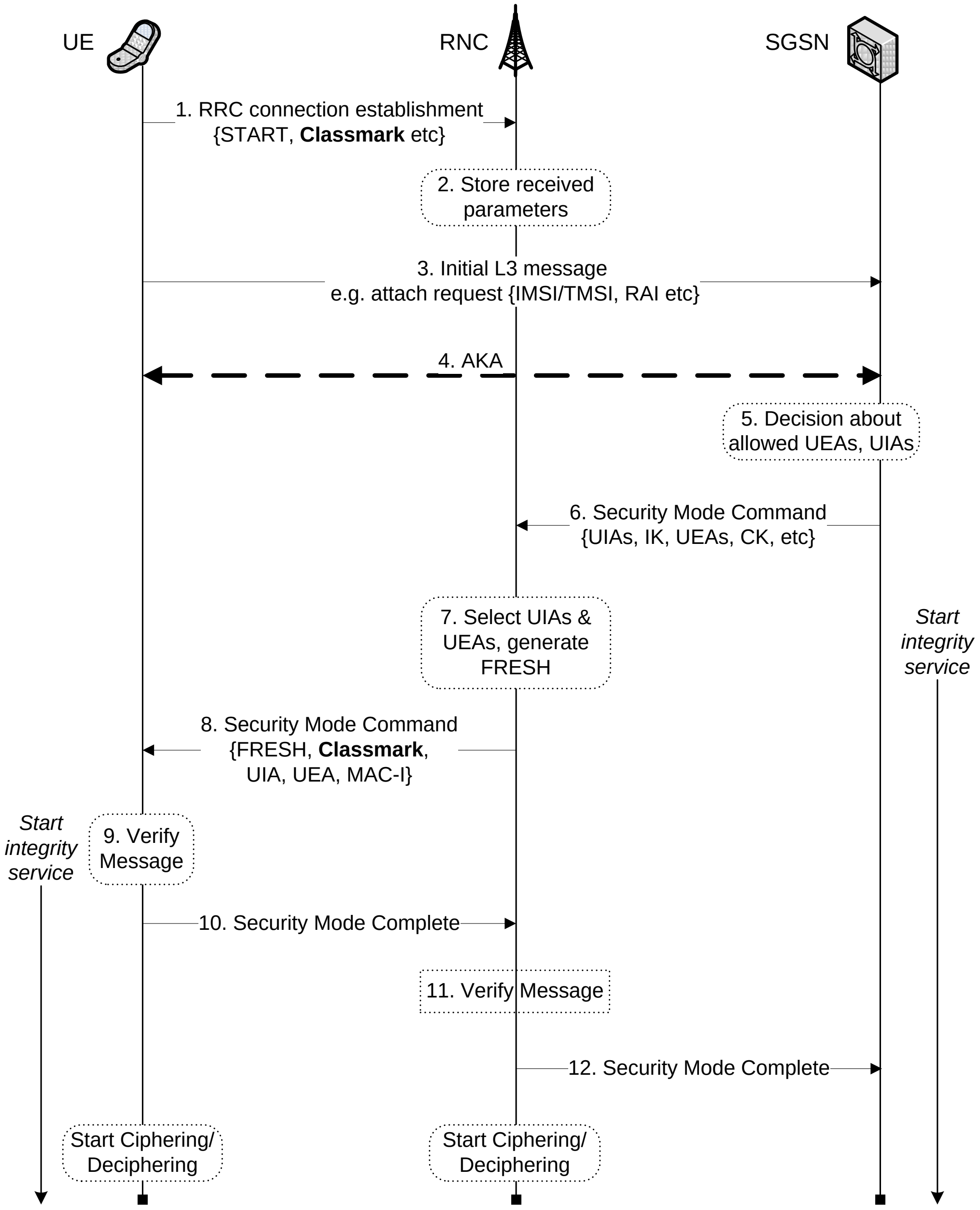
Moreover, as already mentioned, two other keys that will be used for confidentiality and data integrity are calculated by the USIM. Using a security mode command the same keys, which are contained in the initial AV, are transmitted by the SGSN to the corresponding Radio Network Controller (RNC). These keys are known as CK (Cipher key) and IK (Integrity Key). Note that while these keys are part of the corresponding AV and thus immediately available to the SGSN, the USIM has to calculate them by itself. An overview of the authentication sequence described above is depicted in Figure 1.

It is to be noted that this section presents only the fundamental information on UMTS security architecture required for comprehending the concepts described afterward. For a more detailed analysis the reader may refer to [6].

3. DoS attacks in UMTS

In this section we shall describe some vulnerabilities of the UMTS security architecture which can be exploited to launch DoS-type attacks. It is stressed that this paper considers only signaling-oriented DoS attacks expressed in the higher layers of the UMTS, not the physical one. Typically, an attacker would seek unprotected control messages which would attempt to modify in order to manipulate specific procedures or make them repeat. The expected outcome varies: from lower Quality of Service (QoS) that a specific user may experience to a massive denial of any underlying service. In the attacks described below we assume that the attacker carries some special equipment, such as a false Base Station (BS) or a specially modified UE with the help of which is able to act as a man-in-the-middle entity, intercept a valid UE-to-BS session, sniff and replay packets, analyze traffic, and spoof the data of UMTS frames. Also in some cases it is important for the attacker to build a database of valid intercepted IMSIs. Research on the field [3,8] proves that this is a relatively straightforward procedure and, in some cases, requires equipment which is easy to obtain or self-fabricate.

A very simple but primitive DoS attack unfolds as follows: An attacker with a false BS equipment moves close to its target victims. All users' mobile terminals will be deceived into connecting to the false BS if its signal is stronger than the legitimate BS. After the victim is connected to its fake equipment the attacker would simply drop every packet that is transmitted from and towards the UE. This could be described as a variation of a black hole attack and could be conceived as the higher layer equivalent of radio jamming. UMTS security architecture in its current form is not able to counteract this type of attacks [9]. On the other hand, an attacker would rarely adopt such methods to launch DoS attacks because: (a) the attack persists only when the attacker is active, (b) it affects only a small number of users, and (c) it cannot be directed to inflict specific targets (users) only, without affecting others as well, (d) it is easy to detect the attack and locate the malicious element in the network. For these reasons it is likely that an attacker would seek more intelligent ways of launching DoS attacks. Hereunder we shall elaborate on more sophisticated attacks.

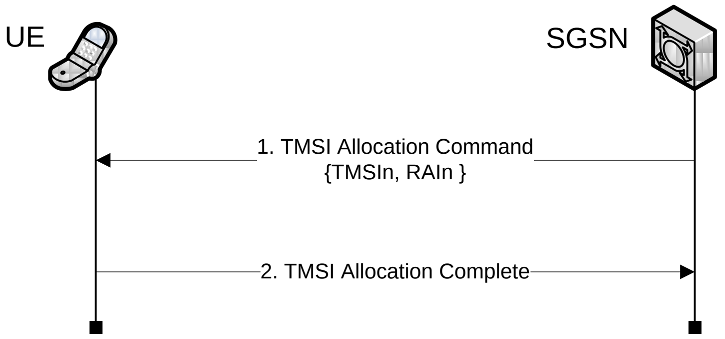

UE
RNC
SGSN
1. RRC connection establishment {START, **Classmark** etc}
2. Store received parameters
3. Initial L3 message e.g. attach request {IMSI/TMSI, RAI etc}
4. AKA
5. Decision about allowed UEAs, UIAs
6. Security Mode Command {UIAs, IK, UEAs, CK, etc}
7. Select UIAs & UEAs, generate FRESH
Start integrity service
8. Security Mode Command {FRESH, **Classmark**, UIA, UEA, MAC-I}
Start integrity service
9. Verify Message
10. Security Mode Complete
11. Verify Message
12. Security Mode Complete
Start Ciphering/ Deciphering
Start Ciphering/ Deciphering

UE = User Equipment, RNC = Radio Network Controller, SGSN = Serving GPRS Support Node, UIA = UMTS Integrity Algorithm, UEA = UMTS Encryption Algorithm, IK = Integrity Key, CK = Cipher Key, MAC-I = Message Authentication Code, RAI = Routing Area Identity

Fig. 1 Start of Security services in UMTS

3.1. Dropping ACK signal

The protection of IMSI is considered a very important issue in UMTS. Therefore, an effort has been made by the designers of the system in order for the IMSI to be transmitted and used as seldom as possible. Instead, temporary identities known as TMSIs are distributed to the users and thereafter are used for all signaling communication. TMSIs are assigned to users, right after the initiation of ciphering. Also new TMSIs are assigned every time a user roams to an area monitored by a different SGSN. Although, a TMSI is transmitted encrypted to the UE the SGSN does not associate the IMSI with the corresponding TMSI unless it receives a TMSI Allocation Complete message from the MS. If this message never reaches the intended SGSN then both the associations {IMSI, TMSI_{old}} and {IMSI, TMSI_{new}} are considered valid by the SGSN in charge for uplink communication and the UE is free to use any of them. Contrariwise, for the downlink, the IMSI must be used because the network has no means to know which one of TMSI_{new} or TMSI_{old} is valid at the UE side at this particular moment. Upon such an event, the SGSN will instantly instruct the MS to delete every available TMSI. In either of the two cases the network may initiate the normal TMSI allocation procedure. Of course, repeated failure of TMSI reallocation may be reported for further maintenance actions by the network provider.



UE = User Equipment, SGSN = Serving GPRS Support Node, TMSI = Temporary Mobile Subscriber Identity, RAI = Routing Area Identifier

Fig. 2 TMSI allocation procedure

Capitalizing on the aforementioned situation the aggressor might wish to position his equipment to a strategic location, for instance circumferential to a given network cell (where typically new TMSIs are assigned to subscribers entering the cell after a handoff). Then, he would monitor for TMSI Allocation Command messages and immediately drop any following TMSI Allocation Complete message as depicted in Figure 2. This would cause new TMSIs to be created repeatedly, which would be expressed as DoS to all the users entering the particular routing area. Although the creation of a new TMSI is a multistep procedure it cannot be considered resource demanding. So, extending this attack to become a flooding attack is considered rather difficult. Nevertheless this attack can be launched alone to offend small groups of users of the network or alternatively it could be used to expose and collect a large number of IMSIs and then exploit them to launch more dangerous and persistent attacks like the one described further down in section 3.3.

3.2. Modification of unprotected RRC messages

The Radio Resource Control (RRC) messages are considered vital for the smooth and normal operation of the UMTS system. Therefore, these signaling information messages are protected by integrity mechanisms, i.e., by applying a message authentication function. While this is true for most of the RRC messages exchanged between an MS and the corresponding RNC, many messages exist that are not integrity protected and therefore are vulnerable to manipulation. Table 1 presents some of the unprotected RRC messages. This might happen either because these messages are exchanged during the early stage of a connection - when the AKA procedure has not yet completed and thus an IK is not present - or for reasons of efficiency.

Modifying, dropping or substituting unprotected RRC messages is expected to cause general system instability, or at least commotion, which may lead to lower QoS or more probably DoS for the end-user. Theoretically, the ways and possibilities to stress the system with this method are countless. Let us consider the following example: an attacker would insert an RRC Connection Release message during a valid ongoing session. By acting the same way, an attacker could substitute a valid RRC Connection Setup Complete with a RRC Connection Reject message.

Handover to UTRAN Complete
Paging Type 1
Push Capacity Request
Physical Shared Channel Allocation
RRC Connection Request
RRC Connection Setup
RRC Connection Setup Complete
RRC Connection Reject
RRC Connection Release
System Information (Broadcast Information)
System Information Change Indication
Transport Format Combination Control (TM DCCH only)

Table 1. List of unprotected RRC messages

3.3. Modification of the initial security capabilities of MS

This is an extension of an attack already proposed in [8]. The attack is described in detail in section 4 but in short during its execution the attacker modifies a RRC Connection Request message in order to trigger the termination of the connection. The authors limit this attack to a simple user DoS scenario but if we take a closer look we realize that it forces the system to go through a number of (unnecessary) heavyweight operations. If the attacker has a large database of stolen IMSIs at hand he would be able to cause a much more serious damage compared to that caused by a single IMSI as the authors propose in [8]. By utilizing the proper equipment the attacker could create a very large number of simultaneous connection requests with bogus classmarks (a specific field contained in the RRC connection request message), thus causing steps 1 to 9 of Figure 1 to constantly repeat. Obviously, this would stress the system since a large number of heavyweight procedures both bandwidth and computationally intense would take place simultaneously, any of which would require a significant period of time for its completion.

Note that if the security capabilities of the MS is not repeated in step 8 of Figure 1 (Security Mode Command - selected algorithms, security capabilities protected with IK), the attacker is able to deceive the system, as message 1 carrying the security capabilities and integrity/encryption algorithms of MS is not integrity protected. An attacker could therefore request no or weak encryption on behalf of the victim MS (instead of its original security capabilities). In turn, the attacker would inform the MS of the choice of no (weak) encryption by the network in Step 12. Actually, this is the case for hybrid GSM/UMTS network setups as described in [4].

3.4. Modification of periodic authentication messages

Periodic local authentication in UMTS is a procedure destined to provide an additional security mechanism [6]. Potentially, it can provide some sort of integrity protection in the U-plane. According to this procedure the volume of data transmitted during the RRC connection (i.e., the COUNT-C value [6]) is periodically checked by both the RNC and the UE. The system makes use of two variables to keep track of the user data transmitted from the MS towards the network. The first one namely Count-C_{UE} tracks the volume of user data transmitted by the user equipment, while the other, known as Count-C_{RNC}, stores the volume of user data actually received by the corresponding RNC. The value of these variables is cross-checked at regular intervals upon initiation by the RNC in charge. If a significant inconsistency is found then the RNC may decide to abruptly release the connection assuming that someone is injecting or dropping messages on the connection. Assuming that the network provider supports this option, the RNC is constantly monitoring the COUNT-C_{RNC} value associated to each radio bearer and triggers the procedure whenever any of these values reaches a threshold.

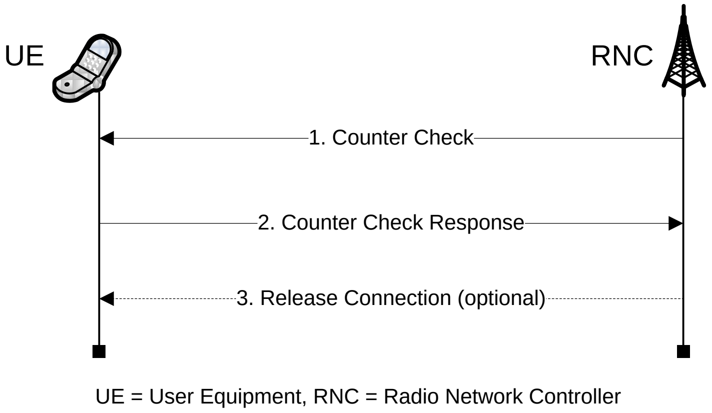

UE
RNC
1. Counter Check
2. Counter Check Response
3. Release Connection (optional)
UE = User Equipment, RNC = Radio Network Controller

Fig. 3 UMTS Periodic authentication procedure

If this threshold is reached, the RNC sends a Counter Check message which contains the most significant bits of Count-C of each active radio bearer. The user equipment compares the Count-C value(s) received from the RNC with its local value of any matching active radio bearer, computes the difference, if any, and constructs a Counter Check Response message containing all differences. If one or more of the values contained in the Counter Check Response message is greater than 0 (null) then the RNC may decide to send a Release Connection message. Otherwise the procedure completes successfully. The above procedure is depicted in Figure 3.

According to the UMTS specifications [6] all the messages described above are integrity protected. As a result, an attacker is not able to modify any of these messages (e.g., spoof the value(s) contained in a Counter Check Response message). If so, the system (RNC) will notice that the received message has been somehow tampered. If this happens, the specification defines that the RNC may release the connection. This would be interpreted as disconnecting the MS or waiting indefinitely for a valid Counter Check Response message to arrive. Without doubt, this issue is provider-specific which of course leaves room for possible errors or misconfigurations.

3.5. SQN Synchronization

During the AKA phase the MS receives a User Authentication Request message from SGSN via the RNC in charge. This message includes the pair {RAND, AUTN}. The AUTN is a sequence of 128 bits produced as the concatenation of the following fields: $\{SQN \oplus AK \| AMF \| MAC\}$. The SQN is a 48-bits sequence number which guarantees the freshness of the message and may be masked with the Anonymity Key (AK) before transmission. The MS (USIM) keeps track of the latest SQN_{HN} values received by storing them locally. When a new AKA procedure initiates the SQN_{HN} is increased to its next value. At least three policies exist for the SQN increment but generally it is expected to be $SQN_{HN} > SQN_{USIM}$. The validity of this relation is checked every time by the USIM. As soon as a AUTN is received, and if SQN concealment applies, the MS has to extract the SQN by XORing the first 48 bits of the AUTN, i.e., $\{SQN \oplus AK\}$, with the AK, where $AK = f5_K(RAND)$. Next, the MAC value contained in the AUTN will be checked against the result of the calculation $XMAC = f1_K\{SQN, AMF, RAND\}$ called the eXpected MAC. If XMAC equals MAC then the MS assumes that the authentication token is original. Otherwise, the procedure aborts and the MS sends a User Authentication Reject message back to the SGSN with an indication of the cause. In this case, the SGSN shall initiate an Authentication Failure Report procedure towards the HLR [6] and the latter may decide to initiate a new identification and authentication procedure towards the user. Assuming that MAC checking succeeds, the MS ensures that SQN is in the proper order. In other words, the MS initially attempts to find out if the received AUTN has been tampered with and then if it is fresh.

Resynchronization is a procedure done for synchronizing the value of SQN in both the MS and HLR. Since it involves the generation of new AVs (normally in batches) and their transmission from the HLR to the SGSN it is classified as a computational intense and network resource demanding procedure. Thus an attacker would wish this procedure to be executed simultaneously for large numbers of users, and if possible, repeatedly in order to mainly overstress the HLR.

However, the attacker cannot simply modify the SQN_{HN} value contained in the AUTN part of the Authentication Request message since it may be concealed with the AK (anyhow it is protected by MAC). Additionally, he is not able to spoof the whole {RAND || AUTN} hoping to subsequently modify the SQN_{HN} in order to make it large or small enough to trigger resynchronization. In fact, every attempt to spoof the SQN would lead to MAC verification failure in the MS and probably the whole procedure would be abandoned. Note this would still be perceived as DoS from a user point of view but it is limited to individual user level DoS rather than massive exhaustion of server resources. Instead of attempting to modify the Authentication Request message and hope for an Authentication Reject one to be sent as response the attacker can apply the following procedure. He would eavesdrop on connections and build a database of MS_i, $\{UserAuthenticationRequest_1, UserAuthenticationRequest_2 \ldots UserAuthenticationRequest_n\}$. After a period of time the attacker would repeatedly replay these messages towards the corresponding MSs. The MAC should be verified but the SQN would fail triggering the Synchronization Failure message to be sent and resynchronization procedure to be initiated towards the HLR. The correct timing the attacker chooses to unleash his attack is important. He may wait a significant amount of time in order for the SQN_{HN} to be sufficiently old or soon enough in order for the SQN_{HN} to be contained in the array of recently received SQNs in the USIM. But also in this case the possibility of flooding the HLR with Authentication Data Requests containing a Synchronization Failure indication is low to moderate. A more sophisticated approach would require several attackers acting simultaneously.

Assuming that each attacker controls several terminals (the acquisition of many different UICCs nowadays is not a problem) he can send quite a lot of Synchronization Failure messages immediately after receiving a User Authentication Request one.

One additional reason of unpredicted behavior and general system instability or commotion might be caused from a rogue User Authentication Reject message sent to SGSN during a valid session. An attacker can easily capture and store a User Authentication Reject message monitored at a previous time and replay it at will. Unlike Synchronization Failure messages which contain secure indications about the cause of the rejection (for instance the AUTS field [6]) the structure of this message is not well defined in the UMTS specification leaving it up to the provider. In this way, it is very easy for a HLR to proceed to user disconnection or to reinitiate the AKA procedure. Therefore, the integrity of Authentication Reject messages should be also protected by a message authentication code as in the case of Synchronization Failure.

What is more, Extensible Authentication Protocol (EAP)-AKA [10] authentication method used for WLAN/UMTS interworking also makes use of AV. As usual, if the received SQN_{HN} is in the incorrect range the MS should perform the SQN synchronization procedure. Here the situation is worse because the communication penalty in terms of network signaling also increases. This is because the cost for accessing HLR is expensive, especially when AAA, SGSN and HLR are located in different countries. That is, the AAA server in the visited domain must notify the HN and request fresh AV from the HN's HLR once more. Therefore, leaving aside the additional overhead caused to the involved entities, another penalty is the bandwidth consumption between the AAA server and the HLR. Note that the same situation applies for IMS which also consumes AV and makes use of several proxies during user registration and authentication.

The synchronization attack described in this subsection is feasible mostly due to weaknesses spotted in the UMTS specification itself. At its current form the UMTS architecture in order to protect against reply attacks in AKA procedure leaves room for DoS attacks. A counter value indicating the number of failed authentication attempts at the MS side can be a valuable tool for both the MS (for avoiding frequent resynchronization attempts) and the HLR (for the same reason and for extracting conclusions and taking appropriate measures). Table 2 provides a more effective way in pseudo-code for dealing with falsified synchronization failure events at the MS side.

```
counter ← 0;
authenticationSuccess ← false;
while (authenticationSuccess = false)
begin
    counter++;
    waitFor(UserAuthenticationRequest);
    compute_AK();
    retrieve_SQN();
    compute_XMAC();

    if (counter < threshold) ! threshold should be a small integer
        if (XMAC <> MAC) OR (SQNRange() = false)
            drop_packet();
        else
            send(UserAuthenticationResponse); ! it contains RES
            authenticationSuccess ← true;
        end if
    else

        if (XMAC <> MAC)
            send(UserAuthenticationReject); ! include AUTS, counter values
            abandonProcedure();
        else if (SQNRange() = false)
            send(SynchronizationFailure); ! include AUTS, counter values
            abandonProcedure();
        else
            send(UserAuthenticationResponse); ! it contains RES
            authenticationSuccess ← true;
        end if
    end if
end
```

Table 2. Improved procedure for authentication function in the USIM

3.6. EAP-AKA originated DoS

In this section we consider some new attacks that originate from the way Extensible Authentication Protocol (EAP)-AKA operates [10]. Note that we do not consider attacks like the Negotiation and Protected Result Indications ones which have been already recognized in the corresponding RFC. Moreover, we left out other sort of attacks like the malformed message style attacks which could potentially also lead to DoS incidents as well.

3.6.1. Background

Extensible Authentication Protocol (EAP) [11] provides a universal authentication framework that is frequently used in wireless LANs, MANs, and cellular networks. EAP is not an authentication mechanism per se, but it offers a series of general functions and a negotiation process based on the preferred authentication mechanism between two parties. EAP-AKA [10] has been specified for achieving access control integration in hybrid UMTS/WLAN network realms. Four entities take part in a full EAP-AKA transaction: (a) the peer which corresponds to the MS (user), (b) the authenticator to which the peer attaches (usually a wireless access point that simply relays EAP messages to and from the EAP server), (c) the EAP server that is located on a backend authentication server using an AAA protocol, and (d) the HLR of the HN. As with other EAP schemes, an identity request/response message pair which contains the subscriber's identity should be exchanged first. Generally, the peer responds including either the cleartext user's permanent ID, or a temporary identity (pseudonym) if identity privacy applies [10]. After getting the peer’s identity, the EAP server acquires an AV for authenticating the subscriber by contacting the HLR in subscriber’s HN.

Next, the AKA protocol initiates by sending an EAP-Request/AKA-Challenge message. This message contains an AT_RAND (same as RAND), an AT_AUTN (same as AUTN), and an AT_MAC. The

AT_MAC attribute contains a message authentication code over the contents of the EAP packet; this is actually a HMAC-SHA1-128 keyed hash value. The peer runs the AKA algorithm as usual, i.e., verifies AUTN and AT_MAC and derives RES and session keys. If everything is correct the peer contacts back the EAP server and send it an EAP-Response/AKA-Challenge. This message contains the RES and the AT_MAC attribute to integrity protect the response. The EAP server makes its own verifications (RES, AT_MAC) and if no mistake is detected it sends an EAP-Success packet indicating that the authentication was successful. The EAP server may also include derived keying material in the message it sends to the authenticator (acting as RNC).

As pointed out in [10] EAP-AKA does not prohibit the use of the EAP Notifications. An EAP Notification can be used at any time in the EAP-AKA exchange. Also, note that EAP-AKA does not protect EAP notifications, (except in special cases called result indications). An attacker can take advantage of this fact by: (a) directly spoofing a notification message indicating an error, thus signifying to the involved parties the failure of the protocol, and (b) sending out misleading messages into fooling one party to transmit a notification message indicating error. Note that EAP-based signaling attacks (referred to as exception triggered DoS attacks) considering EAP-TLS method have been very recently exposed by [12]. Here, we convey them to UMTS and EAP-AKA context specifically. Note however the following attacks can be easily launched against EAP-SIM (for GSM/WLAN networks) and CDMA2000 (when using the EAP-AKA method).

Note that spoofing a notification message is quite straightforward. Specifically, according to [10] the most significant bit of the notification code, which is a 16-bit number, is called the Success bit (S bit). This bit specifies whether the notification designates failure. The notification code values with the S zeroed (code values 0...32767) are used on case of failure. By definition, the peer uses the client error code 0, i.e., "unable to process packet", while the server employs one of the general failure codes ("General failure after authentication" (0) or "General failure" (16384) depending on the phase of the EAP-AKA exchange). When receiving a notification code from this range the peer implies a failed EAP exchange and the server must issue an EAP-Failure packet. The Phase bit (P bit) is the second most significant bit of the notification code. It designates at which phase of the EAP-AKA exchange the notification can be used. For example, If the P bit is set to 1, the notification can only be issued prior to the EAP/AKA-Reauthentication round in re-authentication or before the EAP/AKA-Challenge round in full authentication indicating various failure cases. Simply put, if the P bit is set to 1, then the S bit must be 0.

3.6.2. Fooling the EAP-Server

Two special error messages have been specified for error cases that are related to the processing of the AT_AUTN parameter by the peer (see also section 3.5). Note, that in all but one of the following cases the server is forced into issuing an EAP-Failure causing the authentication exchange to terminate:

- The AT_AUTN parameter is not accepted by the peer. Then, it responds with an EAP-Response/AKA-Authentication-Reject notification.
- The sequence number contained in AUTN is not in the expected range when checked by the peer. Then, the peer responds with an EAP-Response/AKA-Synchronization-Failure notification including the AUTS and RAND causing the resource and bandwidth consuming resynchronization procedure to initiate (see also section 3.5).

If the peer detects any other error in a received EAP-AKA packet, it issues an EAP-Response/AKA-Client-Error message with error code 0. Specifically, this error code is used in various cases, e.g., “The peer encountered a malformed attribute”, “unrecognized or unexpected EAP-AKA Subtype in the EAP request”, “the peer is not able to parse the EAP request” etc. It is stressed that none of the aforementioned peer notifications / messages (EAP-Response/AKA-Authentication-Reject, EAP-Response/AKA-Synchronization-Failure, EAP-Response/AKA-Client-Error) is protected (authenticated) by an AT_MAC attribute. Therefore, these messages could be exploited by an attacker in several stages of the EAP-AKA process. For instance, the attacker could spoof an EAP-Response/AKA-Client-Error message and sent it to the EAP-Server in order to fool him into halting the protocol. Also, it could spoof an EAP-Response/AKA-Synchronization-Failure notification into forcing the server to trigger the costly resynchronization procedure.

3.6.3. Fooling the EAP-Peer

On the other hand, in case the EAP-server detects an error when processing a received EAP-AKA response, it must respond using an EAP-Request/AKA-Notification packet with an AT_NOTIFICATION code that implies failure (see section 3.6.1). Some of the error cases forcing the server to send an EAP-Request/AKA-Notification are: “The server is not able to parse the peer's EAP response”, “The server encounters a malformed attribute, a non-recognized non-skippable attribute, or a duplicate attribute”, “Unrecognized or unexpected EAP-AKA Subtype in the EAP Response” etc [10]. As with peer notifications EAP-Request/AKA-Notification packet is not protected and can be exploited by an attacker into fooling the client to tear down the protocol session.

3.6.4. An Example

In this subsection we describe more analytically such an attack. The EAP-AKA RFC describes two possible ways the peer can behave when receiving a permanent ID request by the server (AT_PERMANENT_ID_REQ) during an identity request/response message pair exchange. Note that the server uses the AT_PERMANENT_ID_REQ attribute in an EAP-Request/AKA-Identity message to request the peer to send its permanent identity. This could be happen because the server does not support fast re-authentication or identity privacy, or the server is not able to map the received pseudonym identity of the subscriber to a permanent identity. However, a received AT_PERMANENT_ID_REQ is not always coming from a legitimate network entity. For instance, an attacker may send such a message to the peer, trying to derive the true identity of the subscriber. Thus, if the peer is not disposed to expose its permanent identity (e.g., if it believes that the network should be able to recognize the pseudonym), then it transmits an EAP-Response/AKA-Client-Error packet with the error code 0, and the authentication exchange terminates. It is also stressed that the AT_MAC attribute cannot be used in the very first EAP-AKA messages during the AKA-Identity phase, since no keying material is available to the parties yet. So, a man-in-the-middle attacker can act in three ways:

- Spoof an EAP-Request/AKA-identity message that includes the AT_PERMANENT_ID_REQ attribute and hope that the peer would be unwilling into disclosing his permanent identity.
- Directly spoof an EAP-Response/AKA-Client-Error packet using the error code 0 and send it towards the server after the server has sent an identity request/response message that includes the AT_PERMANENT_ID_REQ attribute.

- Fabricate an EAP-Response/AKA-Identity message making it to contain an inapprehensible peer identity. This is possible because in case the server used AT_PERMANENT_ID_REQ, and the AT_IDENTITY attribute carrying the identity of the peer in the EAP-Response/AKA-Identity message does not contain a valid permanent identity, then the server transmits an EAP-Request/AKA-Notification packet with AT_NOTIFICATION code "General failure" (16384). This would terminate the EAP exchange.

3.6.5. Flooding the HLR

The EAP-AKA server acquires authentication vectors from the HLR residing in the HN. Thus an insider, i.e., a malicious peer, may produce a lot of EAP-AKA protocol requests to mount a DoS attack against the HLR. Therefore, the server must consider this threat and deploy the necessary countermeasures, e.g., by setting boundaries to the traffic that a given EAP-AKA server generates when communicating with the HLR.

4. Previous Work

In this section we survey in sort previous work. Apart from signaling attacks that apply in core UMTS our analysis includes similar attacks that target the IP Multimedia Subsystem (IMS) infrastructure as well.

4.1. Signaling attacks on Core UMTS

Protection against DoS type attacks is an active research topic but until now, most of the research is focused on Wireless networks such as WLANs, Wireless Sensor Networks (WSN) or 2G mobile networks such as GSM. Many different attacks have been proposed that unfold at different layers like physical, MAC, network or application. For instance, a jamming attack [13] is unleashed in the physical layer and can prove very hazardous since it cannot be addressed by adopting a more sophisticated network security architecture design or by cryptographic techniques. In [14] the authors explore the feasibility and effectiveness of jamming attacks in wireless networks and propose detection schemes. Other examples include: for the data link layer exhaustion attacks [15], and unfairness attacks [16], for the network layer black hole attacks [17,18] and smurf attacks [19], and for the transport layer flooding attacks [20], and desynchronization attacks [19].

Recently, as 3G starts to meet wider acceptance from the consumers and attracts more and more the attention of attackers, further research interest is targeted around the identification and treatment of DoS type attacks for this type of networks. In Khan et al., [8] among other types of attacks, investigate the feasibility of a DoS attack by taking advantage of a particular flow spotted in the UMTS security architecture. Their proposed attack involves the modification of the RRC connection Request Message and more specifically the field which defines the UE security capabilities. This message is not integrity protected since the AKA procedure takes place at a later stage and the MS and SGSN do not share a common IK yet. Any modification of this message will go unnoticed until eventually the AKA procedure completes and the Security Mode Command message is sent to the MS. This message includes the user’s equipment security capabilities as received from the RCC Connection Request message in order to be verified by the UE. In case of mismatch the connection will terminate, but during the process sufficient resources will have been already consumed at both sides.

Lee et al. [21] introduce a novel DoS attack specific for the 3G wireless networks which they identify with the term “signaling attack”. Unlike traditional DoS attacks that unfold in the data plane this one targets and attempt to overload the signaling plane. The signaling attack is implemented by sending low volume (for instance 40 byte packets) bursts at a specific time interval such that as soon as Radio Access Bearer (RAB) is torn down due to a period of inactivity a new packet burst that originates from the attacker forces for a new RAB establishment. This triggering of radio channel allocations/revocations is associated with a large number of signaling operations; more specifically 15 signaling messages are being processed by the RNC for the establishment of a synchronized RAB and 12 messages for its release. The results of this attack are: (a) congestion of the RNC-BS with setup/release messages, (b) consumption of resources of the RNC processor, (c) potentially consumption of the battery of the MS. The attack can prove to be very dangerous since it does not require many resources from the attacker point of view (by using a cable modem with 1Mbps uplink bandwidth the attacker can simultaneously attack 160K MSs) and it can evade detection by traditional IDSs. In the same work the authors propose a technique for detecting and repelling this attack.

Generic DoS protection techniques targeting 3G networks have also been proposed. Lei et al. [22] propose a version of the well known puzzle technique [23] specifically adapted for the UMTS networks. The idea behind this technique is that the client must commit some of its resources before the server commits its own, by solving a cryptographic puzzle. The puzzles should be easy for the server to verify so that the server can do this process massively; while at the same time be computationally inefficient for the client for large numbers. This makes it very difficult for an attacker to unleash a large scale attack and cause overloading of the servers by acting as a legitimate user of the network. The authors in [24] consider a hybrid UMTS/WLAN network architecture and propose a generic protection scheme to resist DoS attacks by utilizing the Authorized Anonymous ID (AAI) method.

4.2. Signaling attacks on IMS

The IMS introduced with 3GPP Rel. 5 provides a powerful framework for deployment of Next Generation Networks (NGN). IMS is based on Session Initiation Protocol (SIP) [25] and the Internet Protocol (IP). IMS connectivity to the Internet makes it by definition vulnerable to DoS attacks. A competent attacker having sufficient resources is able to temporarily paralyze any Internet host, and thus any IMS application server or proxy. This could result in low quality of service offered to legitimate IMS subscribers. Modern, powerful and configurable MSs that are susceptible to compromise is another serious consideration that could lead to DoS. SIP is also a source of many DoS attacks to IMS as described further down.

Until now, several works address signaling oriented DoS attacks in IMS [26-28]. Note that almost every attack which is described in the following is directly taken or consists variation of a classical Internet or SIP attack. According to [27] such kind of attacks can be roughly classified into two major categories: (a) the time-dependent and (b) time-independent ones. The first category includes attacks that require a time interval to influence adversely or damage the target, e.g., a flooding attack. The second category affects instantly the target as soon as it is triggered successfully.

TCP SYN and* TCP/ACKs *Flooding: This well known category of attacks achieves its goals by creating half-open connections at the victim. Such a situation occurs when the server sends a SYN-ACK message, but never receives an ACK message from the client. Specifically, the attacker sends a spoofed SYN packet with an unreachable source IP address. Upon reception, the victim will respond with a SYN-ACK message, but the network is not able to route it. Therefore, the victim never receives an ACK message responding to the initial SYN-ACK message. Even worse, the memory reserved by the pending connection can be freed only after the TCP connection have been timeout. A similar attack to TCP SYN is the TCP/ACKs flood one. The attacker sends packets to arbitrarily selected destination IP addresses and spoofs the source address of the packets to be the victim's address. To maximize the attack consequences the attacker can use an arsenal of bots referred to as distributed TCP/SYN flood attack.

SIP Message Flooding Attack: SIP is a text-based signaling protocol, so there are quite a few types of SIP message flooding attack. The most significant ones are: Invite Flooding, Register Flooding, and Register Response Flooding. According to the first one the aggressor sends a large number of SIP Invite messages with a spoofed source IP address to the victim making it to consume recourses for processing the incoming messages. The Register flooding attack works similarly to the previous attack but it utilizes the Register SIP method instead of the Invite one. According to the latter attack the aggressor sends a large amount of Register messages with wrong credentials to the SIP proxy (called Call Session Control Function (CSCF) in IMS terminology) in order to overwhelm it.

SIP parser attacks: As SIP is a text-based protocol with a highly degree of freedom, an efficient parser is needed which only parses messages up to the point the information is required. However, even a perfectly valid SIP message can be constructed in a way to hamper proper parsing. For instance, an attacker can create unnecessarily long messages in a simple way by adding additional headers (such as informative header fields) in conjunction with a large message-body. Instead of only depleting processor power, longer message also increase network utilization and memory usage.

SQL injection: Contrariwise to all the aforementioned attacks this one is time independent. Message tampering attacks in SIP applications is quite easy due to the text-based nature of SIP messages. SQL injection has been already proved its effectiveness in the Internet but normally it can be used against any application that constructs and executes SQL statements on-the-fly. The primary target of this attack is malicious data modification, but can also be used for paralyzing database services in SIP proxies. As discussed within the SNOCER project (http://www.snocer .org) the attacker can also utilize HTTP messaging to launch SQL injection attacks because the IMS application server integrates the HTTP Servlet container.

SIP Time Independent Message Flows Attack: While the SIP protocol specification describes methods to end or terminate session, redirect a call, cancel an invitation etc it does not specify any custom-tailored security mechanisms. Thus, it is possible for aggressors to capitalize on any vulnerability in the SIP methods and cause DoS to IMS services. The main reason that an attacker is able to launch attacks by employing these messages is that current SIP specifications do not mandate authentication for all of the aforementioned methods. The most interesting attacks in this category are [29]:

- *The BYE attack*: The BYE request is used to terminate an established SIP session. An attacker may utilize the BYE request to tear down the session. To launch this attack, the attacker needs to learn all necessary session parameters (e.g., Session-ID, RTP Port etc.). This can be realized either by sniffing the network traffic or performing a man-in-the-middle attack to inject a BYE request into the session.
- *The CANCEL attack*: The CANCEL request, as its name implies, is used to cancel a previous request sent by a client. More specifically, it asks the corresponding server to cease processing the request

and generate an error response designating that request. The attacker may utilize the CANCEL method to cancel an INVITE request generated by a legitimate user to cause DoS.

- *The Re-INVITE attack*: Once a dialog-session has been established by initial SIP messaging, subsequent requests can be sent that attempt to modify the parameters of the dialog-session (e.g., address or port modification). Thus, any unauthorized modification with a forged Re-INVITE of a dialog-session by a potential attacker may cause a DoS.
- *The UPDATE attack*: The SIP UPDATE method provides the end-users with various capabilities, such as muting or placing on hold incoming calls, identification of QoS service, and negotiation for other session attributes like Re-INVITE. So, similar to the RE-INVITE attack, an attacker may send a forged UPDATE message in order to modify the initial session parameters to cause a DoS change of parameters like QoS or initial addresses and ports.

DNS Attacks: A rather simple way to disturb SIP server operation is to include un-resolvable host names in a SIP message. A SIP message can contain Uniform Resource Identifiers (URI) in varying header fields, including Via, Route, Record-Route, Contact, and Request-URI. A SIP server encountering an un-resolvable address in a header field (e.g., *Via: unintelligible.domain.org*) has to wait for the resolver reply to continue operation. Simulation demonstrates that a SIP server can be blocked for up to 5 secs through one simple message [30].

Operator-oriented Threats: As IMS operators will struggle in the near future to increase their market share by attracting more and more users they may become a threat to each other. That is, they will probably attempt to make the subscribers to change operator and enroll in their network. So, they may try to disturb (or steal bandwidth from) the competitors' network and services. For instance, subscribers could experience network unavailability, bad quality of their video calls, or other QoS problems making them to ultimately decide to change operator.

Apart from well known solutions like firewalls, other well-accredited mechanisms to mitigate IMS specific attacks are the TLS or IPsec protocols to provide authentication, integrity and confidentiality by securing SIP messages in an IMS Service Delivery Platform. Generic Bootstrapping Architecture (GBA) [31] and Generic Authentication Architecture (GAA) [32] can also be employed for authenticating users before accessing services [26]. The deployment of an IDS for IMS Application Servers is also under consideration. Such an IDS would detect and prevent attacks which cannot be detected and tackled by standard security mechanisms as in the case of malicious insiders.

5. Discussion and Future work

The detection of the attacks discussed in this paper is rather infeasible by the generic signature based Intrusion Detection System (IDS) such as the well known Snort (http://www.snort.org/). Instead, any detection mechanism should fully comprehend the employed protocols and be able to recognize the symptoms of each particular attack. Preferably, it should be capable of reacting upon detection, e.g., by banning network access to misbehaving peers. Generally, we can identify two main symptoms when such attacks unfold. First off, the protocol under execution always ends suddenly and usually abnormally or makes loops. Secondly, there may exist many and possibly conflicting or paradoxical response messages with different contents during the same stage of the protocol's state machine. This symptom can exclude the most common exceptions that stem from normal or occasional situations, e.g., a message is lost on the way or a received message is found malformed. Having these symptoms occurring repeatedly within a certain time span is an indication that a signaling attack takes place. In fact, only a stateful IDS would be able to detect such an attack, supposing that is capable of receiving and parsing every stage of each protocol in use. Therefore, the implementation of such a stateful IDS is not exactly a straightforward task. Actually, finding a generic solution in terms of an IDS for all protocols discussed here is a real challenge.

In any case, for core UMTS, the number of signaling messages that do not afford an integrity service must be limited. Signaling takes place at three different layers, i.e., RRC, Radio Link Control (RLC) and Medium Access Control (MAC). However, RRC layer signaling is the most sensitive one thus its integrity is protected by using the IK. On the other hand, RLC and MAC signaling is protected by means of encryption. Therefore, threats to signaling do exist especially for messages preceding the AKA procedure. In this context, an integrity mechanism should exist to protect all message exchanges before the IK is in place. As discussed in section 3.2 all RRC messages should also be integrity protected; otherwise the attacker is equipped with the simplest means to launch a simple but effective DoS attack.

Our ongoing and future work concentrates on two issues. First, find an alternative way to provide an integrity mechanism for protecting the network against flooding attacks. Our intension is not to replace or patch the standard UMTS integrity protection but to provide a simple method to safeguard signaling before AKA execution. In this direction we are examining some variations of the client puzzle scheme [23, 33-35]. At the same time, we are also working on the kind of actions, in terms of protocols, that should be executed when malicious traffic injects into the network, e.g., the received messages systematically do not pass the underlying integrity controls. This would lead to an IDS engine able to recognize and counteract upon such attacks.

6. Conclusions

3G system, and particularly UMTS, offers an answer to the weaknesses of GSM security and adds security features for new 3G radio access networks and services. MS-to-network mutual authentication, stronger confidentiality provided in the U-plane, and the protection of signaling messages integrity seem to fix certain 2G security issues towards making mobile communications safer, trustworthy, and thus, more attractive to consumers. Nevertheless, this might be not enough against competent attackers since several flaws are documented in the literature so far. After surveying related work, we introduce several additional flaws that can be relatively easy exploited by malicious individuals to launch hazardous DoS attacks. We show that such unwanted situations are possible not only to core UMTS but in UMTS/WLAN integrated environments as well. The inner workings of such an attack capitalize mostly on weaknesses found in signaling to achieve its goals. So, giving the fact that attackers become more and more creative there is an urgent need for more effective and carefully designed DoS countermeasures. This will allow the systems to deliver smooth and quality services to their subscribers.

References

[1] G. Kambourakis, C. Kolias, S. Gritzalis, J. H. Park, "Signaling-oriented DoS Attacks in UMTS Networks", Proceedings of the ISA 2009 3rd International Conference on Information Security and Assurance, pp. 280-289, 2009, Seoul, Korea, LNCS 5576, Springer.

[2] Kazumi Algorithm Specification, ETSI TS 135 202 V7.0.0, http://www.etsi.org/website/document /algorithms/ts_135202v070000p.pdf, accessed on 13/01/2008.

[3] C. Tang, D.O. Wu, “Mobile Privacy in Wireless Networks-Revisited”, IEEE transactions on the wireless communications, vol. 7, no. 3, pp. 1035-1042, March 2008.

[4] U. Meyer, S. Wetzel “A Man-in-the-Middle Attack on UMTS”, in proc. of WiSe’04, Oct 2004, Philadelphia, Pennsylvania, USA, pp. 90 - 97.

[5] Yi-Bing Lin, Ming-Feng Chang, Meng-Ta Hsu, and Lin-Yi Wu, “One-Pass GPRS and IMS Authentication Procedure for UMTS”, IEEE Journal on selected areas in communications, Vol. 23(6), June 2005.

[6] 3GPP Technical Specification, 3GPP, 3G Security; Security architecture (Release 8), TS 33.102 V8.3.0, June 2009.

[7] ISO/IEC 9798-4 (1999). Information Technology; Security Techniques; Entity Authentication Part 4: Mechanisms using a cryptographic check function, 1999.

[8] M. Khan, A. Ahmed, A.R. Cheema, “Vulnerabilities of UMTS Access Domain Security Architecture”, 9th ACIS International Conference on Software Engineering, Networking, and Parallel/Distributed Computing, pp. 350-355, Phuket, Thailand, Aug. 2008.

[9] 3GPP TR 33.900 (1.2.0), “A Guide to 3G Security”, Jan., 2000.

[10] J. Arkko, H. Haverinen, “Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)”, IETF RFC 4187, Jan. 2006.

[11] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Ed., “Extensible Authentication Protocol (EAP)”, RFC 3748, June 2004.

[12] Y. Zhao, S. Vemuri, J. Chen, Y. Chen, H. Zhou and Z. (Judy) Fu, “Exception Triggered DoS Attacks on Wireless Networks”, in proceedings of International Conference on Dependable Systems and Networks (DSN-2009), Portugal, IEEE.

[13] W. Xu, W. Trappe, Y. Zhang, T. Wood, “The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks”, proc. 6th ACM MobiHoc, 2005, pp. 46-57, ACM.

[14] W. Xu, K. Ma, W. Trappe, Y. Zhang, “Jamming sensor networks: attack and defense strategies”, IEEE Network, Vol. 20(3), pp. 41-47, May-June 2006, IEEE.

[15] D.C. Nash, T.L. Martin, D.S. Ha, M.S. Hsiao, “Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices”, 3rd IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05), pp.141-145, 2005.

[16] Y. Zhou, D. Wu, S. Nettles, “Analyzing and preventing MAC-layer denial of service attacks for stock 802.11 systems”, In Workshop on Broadband Wireless Services and Applications (BROADNETS), Oct. 2004.

[17] S. Ramaswamy, H. Fu, M. Sreekantaradhya, J. Dixon, and K. Nygard “Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks”, in proc. of 2003 International Conference on Wireless Networks (ICWN'03), pp. 570-575. Las Vegas, Nevada, USA, 2003.

[18] M. Al-Shurman, S.M. Yoo, S. Park, “Black hole attack in mobile ad hoc networks”, in proc. of the 42nd ACM annual Southeast regional conference, pp. 96 – 97, 2004, ACM.

[19] A.D. Wood, J.A. Stankovic, “Denial of service in sensor networks”, IEEE Computer, Vol. 35(10), Oct. 2002, pp. 54 – 62, IEEE.

[20] H. Wang, D. Zhang, K.G. Shin, “Detecting SYN flooding attacks”, in proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol 3. (2002) 1530–1539, IEEE.

[21] P.P.C. Lee, T. Bu, T. Woo, “On the detection of signaling DoS attacks on 3G/WiMax Wireless Networks”, Computer Networks, Accepted for publication, 2009, Elsevier.

[22] Y. Lei, S. Pierre, A. Quintero, "Client Puzzles Based on Quasi Partial Collisions Against DoS Attacks in UMTS", 64th IEEE Vehicular Technology Conference, pp. 1-5, 2006, IEEE.

[23] X. Wang and M. Reiter, “Defending against denial-of-service attacks with puzzle auctions,” in Proc. IEEE Security and Privacy, pp. 78–92, 2003.

[24] H. Qu, Q. Cheng, E. Yaprak, L. Y. Wang, “Towards actively defending from DoS Arracks in UMTS-WLAN”, Ubiquitous Computing and Communication Journal (UbiCC), Vol. 3(3), July 2008.

[25] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, ”SIP: Session Initiation Protocol”, RFC 3261, June 2002.

[26] M. Sher, T. Magedanz, “Protecting IP Multimedia Subsystem (IMS) Service Delivery Platform from Time Independent Attacks”, 3rd International Symposium on Information Assurance and Security (IAS 2007), pp. 171-176, IEEE.

[27] M. Sher, S. Wu, T. Magedanz, “Security Threats and Solutions for Application Server of IP Multimedia Subsystem (IMS-AS)”, IEEE/IST MonAM2006 - Workshop on Monitoring, Attack Detection and Mitigation, Tuebingen, Germany, 2006.

[28] S. Wahl, K. Rieck, P. Laskov, P. Domschitz, and K.-R. Müller, “Securing IMS Against Novel Threats”, Bell Labs Technical Journal 14(1), pp. 243–258 (2009), Wiley.

[29] D. Geneiatakis, A. Dagiouklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, S. Ehlert, D. Sisalem, "Survey of Security Vulnerabilities in Session Initiation Protocol", IEEE Communications Surveys and Tutorials, Vol. 8(3), pp. 68-81, 2006, IEEE.

[30] D. Sisalem, J. Kuthan, S. Ehlert, “Denial of Service Attacks Targeting a SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms”, IEEE Network, pp. 26-31, September/October 2006, IEEE.

[31] 3GPP, Generic Bootstrapping Architecture (GBA) (Rel. 7), TS 33.220 V7, 2005.

[32] 3GPP, “Generic Authentication Architecture (GAA); Access to Network Application Functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) (Rel.7)”, TS 33.222 V7, 2005.

[33] B. Waters, A. Juels, J. Halderman, and E. Felten, “New client puzzle outsourcing techniques for DoS resistance,” in Proc. Computer and Communications Security, pp. 246–256, 2004.

[34] W. Feng, E. Kaiser, W. Feng, and A. Luu, “The design and implementation of network puzzles,” in proc. of INFOCOM, pp. 2372-2382, 2005, IEEE.

[35] V. Gligor, “Guaranteeing access in spite of service-flooding attacks,” in Proc. of 11th Security Protocols Workshop, 2003.