Project Report

CS 649 : NETWORK SECURITY

GSM and UMTS Security

Under Guidance of

Prof. Bernard Menezes

Submitted By
Vishal Prajapati (08305030)
Vishal Sevani (07405010)
Om Pal (07405702)
Sudhir Rana (05005002)

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay

CONTENTS

Abstract

Security requirements and services of a mobile communication system differ, due to the radio communication between the user and the base station, extensively from those of a fixed network. There is no physical link in the form of a (fixed) telephone line between the user and the local exchange, which could serve to "identify" the user for routing and charging purposes. Authentication by means of cryptographic procedures is thus required to stop impostors from taking on the identity of somebody else and "transferring" calls and charges. Eavesdropping on the radio path, intercepting data or tracing the whereabouts of a user by listening to signaling data are other serious threats.

Keywords and Abbreviations:

AuC Authentication Centre
BSS Base Station System
HLR Home Location Register
IC Integrated Circuit
IMSI International Mobile Subscriber Identity
Kc Ciphering key
Ki Individual subscriber authentication key
LAI Location Area Identity
ME Mobile Equipment
MS Mobile Station
MSC Mobile Switching Centre
PIN Personal Identification Number
PLMN Public Lands Mobile Network
PUK PIN Unblocking Key
R**AND** Random Number
SIM Subscriber Identity Module
SRES Signed Response
TMSI Temporary Mobile Subscriber Identity
VLR Visitor Location Register
UMTS Universal Mobile Telecommunications System
UTRA UMTS Terrestrial Radio Access
USIM Universal Subscriber Identity Module
AKA Authentication and key agreement
RNC Radio Network Controller,
SN **S**erving network

1. Introduction

Mobile communications are the fastest growing service industry in the world today, it is estimated that by the year 2005, wireless companies will have more than 1 billion subscribers around the world. Subscribers can, not only talk over their cellular phones but the can also have access to many other services that were limited to stationary workstations before, services such as internet access, chat services, online banking, data transfer, etc. are becoming very common in this days. But as services increase so do the security measures to be taken, more valuable and private information is sent through wireless networks everyday and this information has to be protected from mal intentioned people who may try to access it. With the older analog-based cellular telephone systems such as the Advanced Mobile Phone System (AMPS) and the Total Access Communication System (TACS), it is a relatively simple matter for the radio hobbyist to intercept cellular telephone conversations with a police scanner.

New technologies make it harder to third parties to have access to this data or voice communications. Global Standard Mobile GSM (originally Group Special Mobile) and UMTS Universal Mobile Telecommunications System are very good examples, of the cellular technologies available today.

2. GSM Overview

The GSM standard specifies the frequency bands of 890 to 915 MHz for the uplink band, and 935 to 960 MHz for the downlink band, with each band divided up into 200 kHz channels. Other features of the radio channel interface include adaptive time alignment, GMSK modulation, discontinuous transmission and reception, and slow frequency hopping. Discontinuous transmission and reception refers to the MS powering down during idle periods and serves the dual purpose of reducing co-channel interference and extending the portable unit's battery life. Slow frequency hopping is an additional feature of the GSM radio channel interface which helps to counter the effects of Rayleigh fading and co-channel interference. It uses TDMA as its access method.

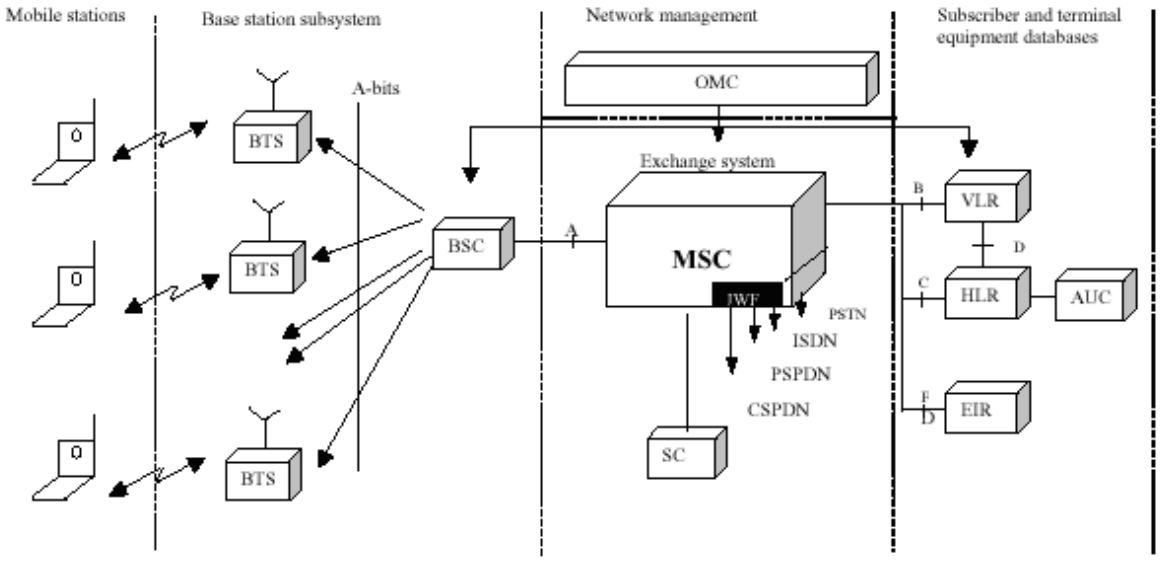


Figure 1: GSM Network

2.1. Architecture

The GSM network is divided in 4 sections :

- **Mobile Stations**

  The subscriber will use a mobile station to make and receive calls via the GSM network. The MS is composed of two distinct functional entities, the subscriber identity module (SIM), which is a removable smart card, and the mobile equipment.

- **Base Station Subsystem (BSS)**

  The MS communicates with the base transceiver station (BTS) via the radio interference. A BTS performs all the transmission and reception functions relating to the GSM.

- **Network Management**

  Every BSS is connected to a Mobile services switching centre (MSC). The MSC is concerned with the routing of calls to and from the mobile users. The Home Location Center (HLR) is used to store information that is specific to each subscriber. Every GSM subscriber will have a record in the HLR.

- **Subscriber and terminal equipment database**

  The Authentication Center (AuC) is much related to the HLR. The AuC is solely used to store information that is concerned with GSM security features, i.e. user authentication and ratio path encryption. It will contain the subscriber's secret key Ki and the A3 and A8 security algorithms. The Visitor Location Register (VLR) is associated with one or more MSCs and it contains information relating to those subscribers that are currently registered within the MSC area(s) of its associated MSC.

2.2. Security features

The subscriber is uniquely identified by the International Mobile Subscriber Identity (IMSI). This information, along with the individual subscriber authentication key (Ki), constitutes sensitive identification credentials analogous to the Electronic Serial Number (ESN) in analog systems such as AMPS and TACS. Security in GSM consists of the following aspects:

- Authentication
- Signal and Data confidentiality
- Identity confidentiality

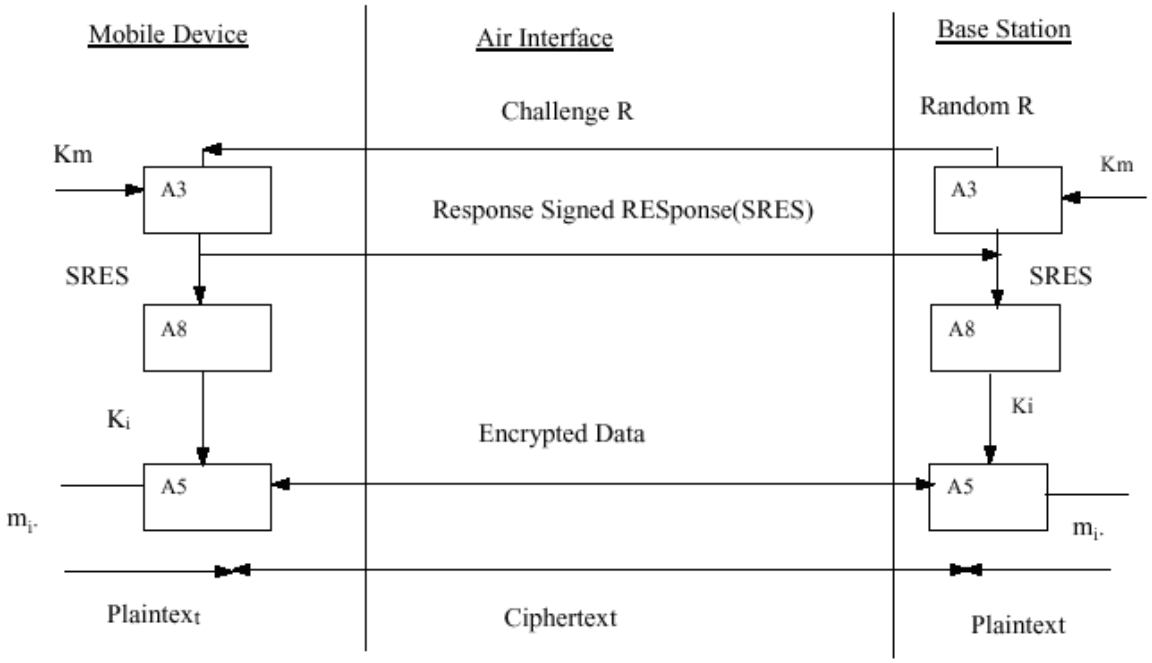


Figure 2: GSM Security Architecture

2.2.1. Authentication

The GSM Security Model is based on a shared secret between the subscriber's home network's HLR and the subscriber's SIM. The shared secret, called Ki, is a 128-bit key. When the MS first comes to the the area of a particular MSC, the MSC sends the Challenge of the first triple to the MS. The MS calculates a SRES with the A3 algorithm using the given Challenge and the Ki residing in the SIM. The MS then sends the SRES to the MSC, which can confirm that the SRES really corresponds to the Challenge sent by comparing the SRES from the MS and the SRES in the triple from the HLR. Thus, the MS has authenticated itself to the MSC.

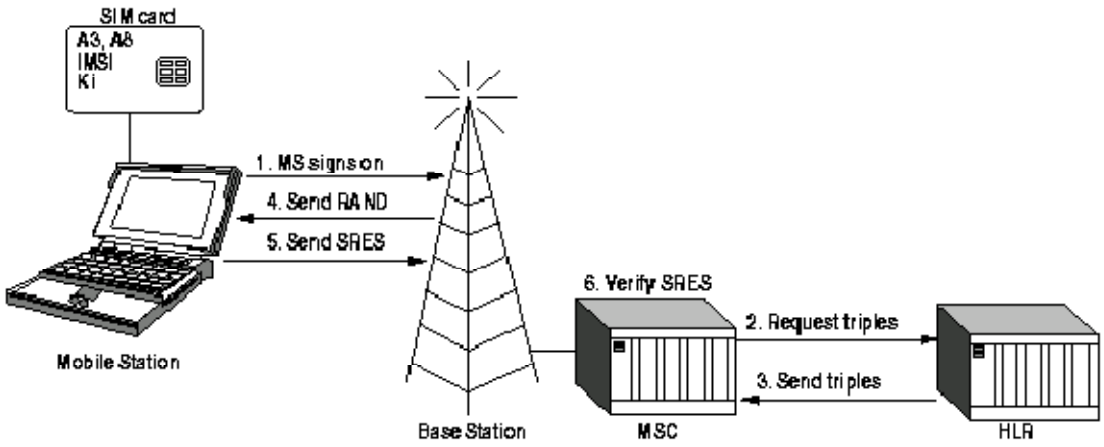


Figure 3: GSM Authentication

2.2.2. Signal and Data confidentiality

The SIM contains the ciphering key generating algorithm (A8) which is used to produce the 64-bit ciphering key (Kc). The ciphering key is computed by applying the same random number (RAND) used in the authentication process to the ciphering key generating algorithm (A8) with the individual subscriber authentication key (Ki). The ciphering key (Kc) is used to encrypt and decrypt the data between the MS and BS.

2.2.3. Identity confidentiality

To ensure subscriber identity confidentiality, the Temporary Mobile Subscriber Identity (TMSI) is used. The TMSI is sent to the mobile station after the authentication and encryption procedures have taken place. The mobile station responds by confirming reception of the TMSI.

3. Encryption algorithms

GSM algorithms were developed in secret, official descriptions were not published to the public. Most of the information available come from leaked documents and cryptanalysis projects.

GSM specifications define 3 algorithms:

- A3 the authentication algorithm.
- A8 the key generation algorithm.
- A5 the encryption algorithm.

3.1 The COPM128 algorithm

The algorithm is used in GSM networks for the purpose of authenticating the Mobile Station to the Base Transceiver Station and to initialize the encryption standard A5 which is used to encrypt the air to air connection. The GSM specification only mentions COMP128 as an example of how to implement the A3 (authentication) and A8 (key derivation) algorithms using a single function. The algorithm is still not available to the public but it has been reverse engineered by Marc Briceno, Ian Goldberg, and David Wagner in 1998.

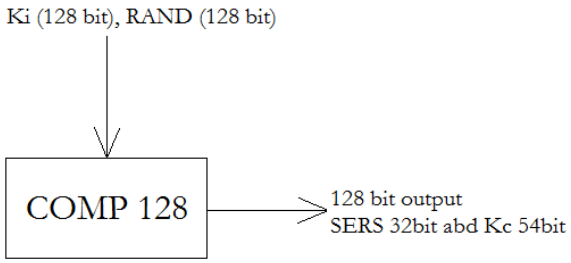


Figure 4: COMP128 Algorithm

Basic functionality

- Input: 16 Byte random value (challenge)
- 16 Byte secret key of the mobile station
- Output: 12 byte
- -32 bit used for authentication
- -54 + 10 bit used for A5 initialization

Order of events

1. The random and key value are concatenated to the input x.
2. The input is hashed (8 times) which reduces it from 32 bytes to 16 bytes.
3. After each but the last hashing, the result value x' is permutated.
4. The result of the permutation is used as the random input for the next round.
5. After 8 passes the hash value is used as the algorithm´s output without permuting it.

3.1.1 COPM128 Weaknesses

In April 1998, the Smartcard Developer Association (SDA) together with two U.C. Berkeley researchers claimed to have cracked the COMP128 algorithm stored on the SIM. By sending large number of challenges to the authorization module, they were able to deduce the K_I within several hours. They also discovered that K_C uses only 54 bits of the 64 bits. The remaining 10 bits are replaced by zeros, which makes the cipher key purposefully weaker. They feel this is due to

government interference. A weaker ciphering key, could potentially allow governments to monitor conversations.

The COMP128 algorithm was cracked exploiting a weakness in the butterfly structure; only the random number of the input can be variable. A collision on the stage 2 of the hash function occurs. It will propagate entirely to the output, so it will be detectable. To launch the attack one has to vary the I + 16 and I + 24 bytes of the COMP128 input and fix the remaining input bytes. The birthday paradox guarantees that the collision will occur rapidly and the colliding bytes are i, i + 8, i+ 16 and i + 24. The attack requires 2^17.5 queries to recover the hole 128bit key.

3.2. The A5 Algorithm

There exists several implementations of this algorithm though the most commonly used ones are: - A5/0 used by countries under UN Sanctions, comes with no encryption.

- A5/1 is the strongest version and is used in Western Europe and America.
- A5/2 is a weaker version used mainly in Asia.

As with A8 and A3, this algorithm was secretly developed but some unofficial descriptions of the algorithms can be found in the internet. The A5 structure is shown in Figure 5

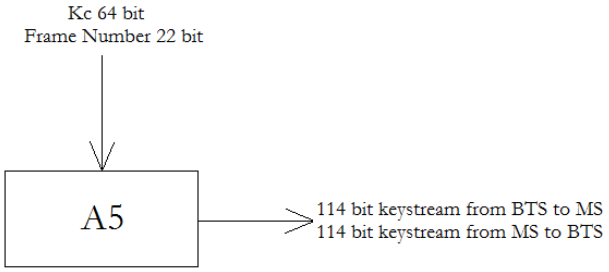


Figure 5: Keystream generation for MS to BTS and BTS to MS

The stream cipher is initialized all over again for every frame sent. The stream cipher is initialized with the session key, Kc, and the number of the frame being de/encrypted. The same Kc is used throughout the call, but the frame number (a22-bit number) frame number changes during the call, thus generating a unique keystream for every frame.

3.2.1 The A5/1 Algorithm description

The A5 algorithm used in European countries consists of three LSFRs of different lengths. The LSFRs are initialized with Kc, and the frame number. The Kc (64-bit) is first loaded into the register bit by bit. The LSB of the key is XORred into each of the LSFRs. The registers are then all clocked (the majority clocking rule is disabled). All 64 bits of the key are loaded into the registers the same way. The 22-bit frame number is also loaded into the register in the same way except that the majority clocking rule applies from now on. After the registers have been initialized with the Kc and the current frame number, they are clocked one hundred times and the generated keystream bits are discarded. This is done in order to mix the frame number and keying material together. Now 228 bits of keystream output are generated. The first 114 bits are used to encrypt the frame from MS to BTS and the next 114 bits are used to encrypt the frame from BTS to MS. After this, the A5 algorithm is initialized again with the same Kc and the number of the next frame.

3.2.2. A5/1 Weaknesses

A5 /1 is a very strong encryption algorithm, the best published attacks to it require 2 ^40 and 2^45 steps which makes it vulnerable to hardware-based attacks of organizations but not to software based

attacks. Its main weakness is that its key is the output of the A8 algorithm which has already been cracked. The actual size of its key is not 64 but 54, because the last 10 bits are set to 0, which makes it much weaker.

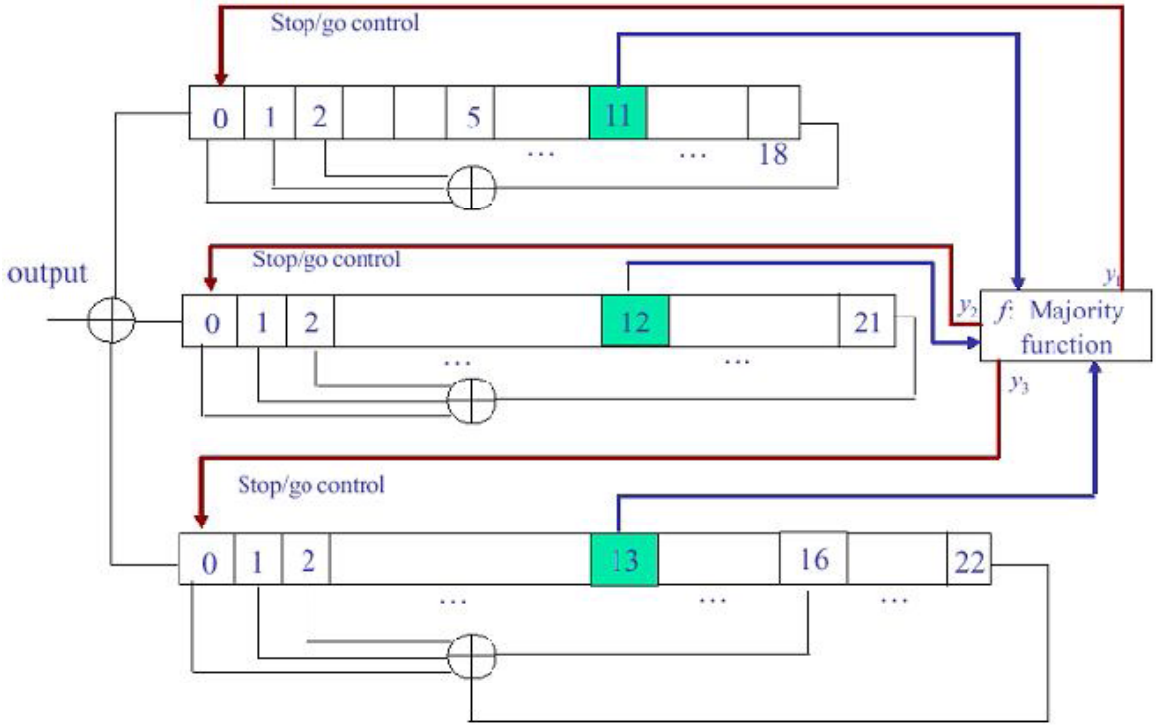


Figure 6: A5/1 Cipher stream cipher

4. UMTS Security Mechanisms

The UMTS access security standards, in particular the new authentication mechanism, are based on research work conducted by the European Union funded USECA project.

4.1 Enhancements in UMTS vs. GSM

- Mutual Authentication
  - provides enhanced protection against false base station attacks by allowing the mobile to authenticate the network
- Data Integrity
  - provides enhanced protection against false base station attacks by allowing the mobile to check the authenticity of certain signalling messages
- Network to Network Security
  - Secure communication between serving networks. MAPSEC or IPsec can be used
- Wider Security Scope
  - Security is based within the RNC rather than the base station
- Flexibility
  - Security features can be extended and enhanced as required by new threats and services
- Longer Key Length
  - Key length is 128 as against 64 bits in GSM

4.2 Entity authentication

UMTS provides mutual authentication between the UMTS subscriber, represented by a smart card application known as the USIM (Universal Subscriber Identity Module), and the network in the following sense:

- Subscriber authentication: the serving network corroborates the identity of the subscriber.
- Network authentication: the subscriber corroborates that he is connected to a serving network that is authorized, by the subscriber's home network, to provide him with services; this includes the guarantee that this authorization is recent.

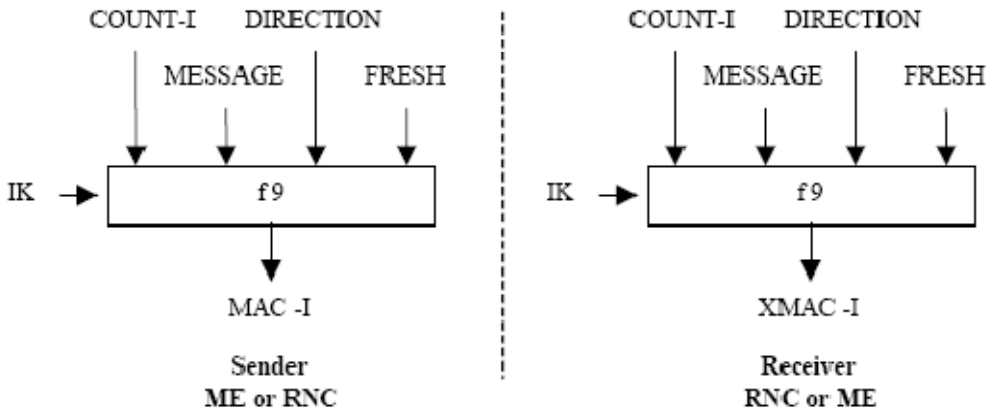


Figure 7: Input and Output parameters of the algorithm

4.3 Signalling data integrity and origin authentication

The following security features are provided with respect to integrity of data on the network access link:

- Integrity algorithm agreement: the mobile station (MS) and the serving network (SN) can securely negotiate the integrity algorithm that they use.
- Integrity key agreement: the MS and the SN agree on an integrity key that they may use subsequently; this is realized as part of the protocol which also provides entity authentication.
- Data integrity and origin authentication of signalling data: the receiving entity (MS or SN) is able to verify that signalling data has not been modified in an unauthorized way since it was sent by the sending entity (SN or MS) and that the data origin of the signalling data received is indeed the one claimed.

The use of the integrity feature for signalling data is mandatory. This security feature has no equivalent in GSM. It provides protection against false base station attacks as the origin of signalling messages required to set up a communication with a mobile can now be authenticated by the mobile. The input parameters to the algorithm are:

- The integrity key IK, which is 128 bits long
- An integrity sequence number (COUNT-I) and a random value generated by the radio network controller (FRESH). COUNT-I and FRESH are each 32 bits long. Together, they provide replay protection.
- A direction identifier (DIRECTION) to prevent so-called reflection attacks

4.4 Mutual authentication and key agreement between user and network

The design of the authentication and key agreement (AKA) protocol for UMTS reflects the results of an analysis of the threats and risks in GSM. It was guided by the principle that the compatibility with GSM should be maximized and the migration from GSM to UMTS, and the handover between GSM and UMTS access networks, should be made as easy as possible. In particular, the changes to the GSM core network should be minimized.

The main changes with respect to the GSM authentication and key agreement protocol are:

- The challenge is protected against replay by a sequence number and it is also 'signed' (integrity-protected). This means that old authentication data intercepted by an attacker cannot be re-used.
- The AKA generates an integrity key in addition to a ciphering key. This integrity key is used to protect the integrity of the signalling data between the MS and the RNC.

4.5 Protection against false base station attacks

While designing UMTS security mechanisms, emphasis was laid on providing protection against false base station attacks, where attacker masquerades as base station to the user. No security was provided for these attacks in GSM, as previously cost involved in carrying out such attacks made this impossibility. The advantages that the attacker can gain by carrying out these attacks are,

1) By suppressing encryption between the target user and the intruder: An attacker with a modified BS entices the user to camp on his false BS and when the service is initiated, the intruder does not enable encryption.
2) By suppressing encryption between the target user and the true network: In this case, during call setup the ciphering capabilities of the MS are modified by the intruder and it appears to the network that there is genuine mismatch of the ciphering and authentication algorithms. After this the network may

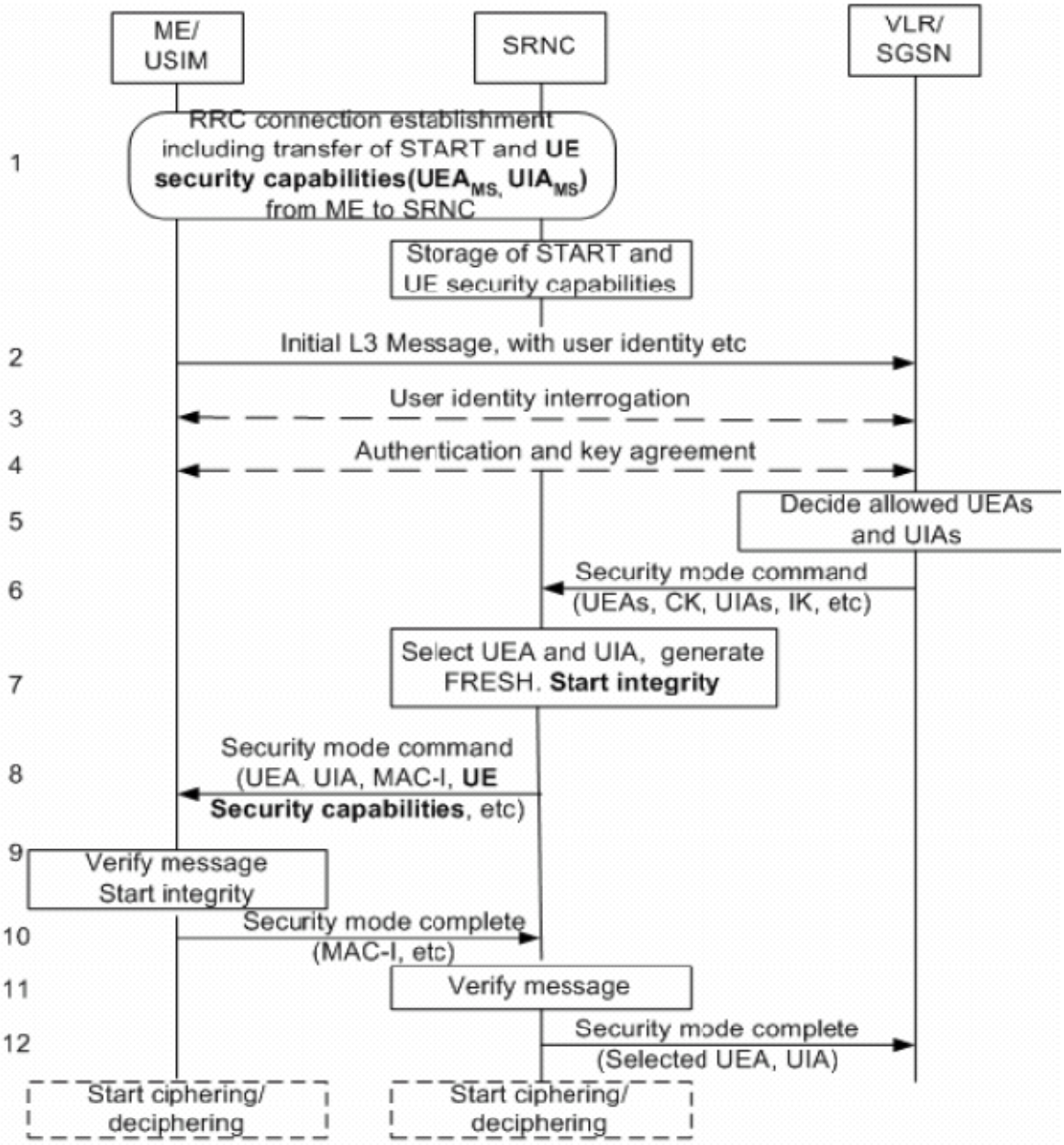
decide to establish an un-enciphered connection: The intruder cuts the connection and impersonates the network to the target user.

3) By forcing the use of a compromised cipher key: The attacker with a modified BS/MS and a compromised authentication vector entices the user to setup a call while camped on his false BS/MS. The attacker then forces the use of a compromised cipher key. One of the primary features in UMTS which safeguards against these attacks is Mutual Authentication as stated above and also integrity protection of signalling data. Briefly stating the exact mechanism followed by UMTS to prevent against these attacks is as follows,

At each new signalling connection establishment between MS and VLR/SGSN, the User Equipment (UE) security capabilities i.e. the USIM Encryption Algorithms (UEAs) and the USIM Integrity Algorithms (UIAs) are transferred from ME to Serving Radio Network Controller (SRNC). After the initial connection establishment there may be an optional user identity request and AKA. The SRNC decides which algorithms to use by selecting the highest preference algorithm from the list of allowed algorithms that matches any of the algorithms supported by the MS. The SRNC generates (an integrity protected) RRC message, security mode command (Fig. 8, Message 8), which includes the UE security capabilities. The UE after verifies that security capabilities transferred in this message are indeed what it had sent to the network, thereby ensuring that it is talking to the valid base station.

4.6 Network Domain Security

GSM does not specify any security mechanisms to be followed for communication between different network elements such as HLR, VLR etc., making it easy for attacker to eavesdrop on these links. To safeguard against this vulnerability UMTS has made it mandatory the use of either MAPSEC or IPSec, so that communication on these links is well protected. The IPSec protocol to be followed for this is similar to that used in wired links, and its use has dominated that of MAPSEC as the focus is shifting towards all IP communication.



FRESH: A network side nonce
ME: Mobile equipment
UE: User equipment

START
MAC-I:
SRNC: Serving Radio Network Controller

UIA: UMTS Integrity algorithm
UEA: UMTS Encryption algorithm

Figure 8: Security mode set-up procedure

5. Personal views

UMTS Security Mechanisms

One of major mistakes done while designing GSM security mechanisms, is that primarily the encryption algorithms were sought to be kept secret and were not submitted for review by peer researchers. Therefore once the algorithms were leaked out, the vulnerabilities were identified easily, leading to the need for redesign these algorithms, whereas in UMTS the algorithms were thoroughly evaluated for any vulnerability by thorough review by researchers, which had led to more robust design of the algorithms.

Also, UMTS has further enhanced security mechanisms by specifying network domain security mechanisms wherein the communication between different network elements is made more secure. Also mechanisms are provided to safeguard against false base station attacks which are nonexistent in UMTS. So, overall it can be stated that the algorithms designed for UMTS are much more robust as compared to GSM as they were submitted for peer review and also as key size is 128 bits (more than double that in GSM) and can be continued to be used for considerable time in future. Also the use of network domain security and integrity protection of signaling data, make these networks much more reliable and ideal for use in today's world when mobile computing is attaining such significance.

6. Conclusion

We discussed about security mechanism GSM and loop hole for attackers in GSM mechanism. After that we talked about UMTS mechanism and its capabilities over GSM. The access security mechanisms in UMTS now protect against the false base station attacks which is not protected in GSM. The confidentiality algorithm is stronger than its GSM predecessor. The integrity mechanism works independent of confidentiality protection and provides protection against active attacks.

UMTS builds upon security mechanisms of GSM, and in addition provides following enhancements,

- Encryption terminates at the radio network controller
- Mutual authentication and integrity protection of critical signalling procedures to give greater protection against false base station attacks
- Longer key lengths (128-bit)
- Network Domain Security using MAPSEC or IPSec

7. References

- **UMTS security**, Boman, K. Horn, G. Howard, P. Niemi, V. Electronics & Communication Engineering Journal, Oct 2002, Volume: 14, Issue:5, pp. 191-204
- "**Evaluation of UMTS security architecture and services**", A. Bais, W. Penzhorn, P. Palensky, Proceedings of the 4th IEEE International Conference on Industrial Informatics, p. 6, Singapore, 2006
- **UMTS Security**, Valtteri Niemi, Kaisa Nyberg, published by John Wiley and Sons, 2003
- **GSM-Security: a Survey and Evaluation of the Current Situation**, Paul Yousef, Master's thesis, Linkoping Institute of Technology, March 2004
- **GSM: Security, Services, and the SIM** Klaus Vedder, LNCS 1528, pp. 224-240, Springer-Verlag 1998
- **Instant ciphertext-only cryptanalysis of GSM encrypted communication**, Elad Barkan, Eli Biham, Nathan Keller, Advances in Cryptology – CRYPTO 2003

Appendix

```
/* An implementation of the GSM A3A8 algorithm.  (Specifically, COMP128.)
 */

/* Copyright 1998, Marc Briceno, Ian Goldberg, and David Wagner.
 * All rights reserved.
 */

/*
 * For expository purposes only.  Coded in C merely because C is a much
 * more precise, concise form of expression for these purposes.  See Judge
 * Patel if you have any problems with this...
 * Of course, it's only authentication, so it should be exportable for the
 * usual boring reasons.
 */

typedef unsigned char Byte;

#include
/* #define TEST */

/*
 * rand[0..15]: the challenge from the base station
 * key[0..15]: the SIM's A3/A8 long-term key Ki
 * simoutput[0..11]: what you'd get back if you fed rand and key to a real
 * SIM.
 *
 *   The GSM spec states that simoutput[0..3] is SRES,
 *   and simoutput[4..11] is Kc (the A5 session key).
 *   (See GSM 11.11, Section 8.16.  See also the leaked document
 *   referenced below.)
 *   Note that Kc is bits 74..127 of the COMP128 output, followed by 10
 *   zeros.
 *   In other words, A5 is keyed with only 54 bits of entropy. This
 *   represents a deliberate weakening of the key used for voice privacy
 *   by a factor of over 1000.
 *
 * Verified with a Pacific Bell Schlumberger SIM.  Your mileage may vary.
 *
 * Marc Briceno , Ian Goldberg ,
 * and David Wagner
 */

void A3A8(/* in */ Byte rand[16], /* in */ Byte key[16],
          /* out */ Byte simoutput[12]);

/* The compression tables. */
static const Byte table_0[512] = {
        102,177,186,162,  2,156,112, 75, 55, 25,  8, 12,251,193,246,188,
        109,213,151, 53, 42, 79,191,115,233,242,164,223,209,148,108,161,
        252, 37,244, 47, 64,211,  6,237,185,160,139,113, 76,138, 59, 70,
         67, 26, 13,157, 63,179,221, 30,214, 36,166, 69,152,124,207,116,
        247,194, 41, 84, 71,  1, 49, 14, 95, 35,169, 21, 96, 78,215,225,
        182,243, 28, 92,201,118,  4, 74,248,128, 17, 11,146,132,245, 48,
        149, 90,120, 39, 87,230,106,232,175, 19,126,190,202,141,137,176,
        250, 27,101, 40,219,227, 58, 20, 51,178, 98,216,140, 22, 32,121,
         61,103,203, 72, 29,110, 85,212,180,204,150,183, 15, 66,172,196,
         56,197,158,  0,100, 45,153,  7,144,222,163,167, 60,135,210,231,
        174,165, 38,249,224, 34,220,229,217,208,241, 68,206,189,125,255,
        239, 54,168, 89,123,122, 73,145,117,234,143, 99,129,200,192, 82,
        104,170,136,235, 93, 81,205,173,236, 94,105, 52, 46,228,198,  5,
         57,254, 97,155,142,133,199,171,187, 50, 65,181,127,107,147,226,
        184,218,131, 33, 77, 86, 31, 44, 88, 62,238, 18, 24, 43,154, 23,
         80,159,134,111,  9,114,  3, 91, 16,130, 83, 10,195,240,253,119,
        177,102,162,186,156,  2, 75,112, 25, 55, 12,  8,193,251,188,246,
        213,109, 53,151, 79, 42,115,191,242,233,223,164,148,209,161,108,
         37,252, 47,244,211, 64,237,  6,160,185,113,139,138, 76, 70, 59,
         26, 67,157, 13,179, 63, 30,221, 36,214, 69,166,124,152,116,207,
        194,247, 84, 41,  1, 71, 14, 49, 35, 95, 21,169, 78, 96,225,215,
        243,182, 92, 28,118,201, 74,  4,128,248, 11, 17,132,146, 48,245,
         90,149, 39,120,230, 87,232,106, 19,175,190,126,141,202,176,137,
         27,250, 40,101,227,219, 20, 58,178, 51,216, 98, 22,140,121, 32,
        103, 61, 72,203,110, 29,212, 85,204,180,183,150, 66, 15,196,172,
        197, 56,  0,158, 45,100,  7,153,222,144,167,163,135, 60,231,210,
```

```
        165,174,249, 38, 34,224,229,220,208,217, 68,241,189,206,255,125,
         54,239, 89,168,122,123,145, 73,234,117, 99,143,200,129, 82,192,
        170,104,235,136, 81, 93,173,205, 94,236, 52,105,228, 46,  5,198,
        254, 57,155, 97,133,142,171,199, 50,187,181, 65,107,127,226,147,
        218,184, 33,131, 86, 77, 44, 31, 62, 88, 18,238, 43, 24, 23,154,
        159, 80,111,134,114,  9, 91,  3,130, 16, 10, 83,240,195,119,253
    }, table_1[256] = {
         19, 11, 80,114, 43,  1, 69, 94, 39, 18,127,117, 97,  3, 85, 43,
         27,124, 70, 83, 47, 71, 63, 10, 47, 89, 79,  4, 14, 59, 11,  5,
         35,107,103, 68, 21, 86, 36, 91, 85,126, 32, 50,109, 94,120,  6,
         53, 79, 28, 45, 99, 95, 41, 34, 88, 68, 93, 55,110,125,105, 20,
         90, 80, 76, 96, 23, 60, 89, 64,121, 56, 14, 74,101,  8, 19, 78,
         76, 66,104, 46,111, 50, 32,  3, 39,  0, 58, 25, 92, 22, 18, 51,
         57, 65,119,116, 22,109,  7, 86, 59, 93, 62,110, 78, 99, 77, 67,
         12,113, 87, 98,102,  5, 88, 33, 38, 56, 23,  8, 75, 45, 13, 75,
         95, 63, 28, 49,123,120, 20,112, 44, 30, 15, 98,106,  2,103, 29,
         82,107, 42,124, 24, 30, 41, 16,108,100,117, 40, 73, 40,  7,114,
         82,115, 36,112, 12,102,100, 84, 92, 48, 72, 97,  9, 54, 55, 74,
        113,123, 17, 26, 53, 58,  4,  9, 69,122, 21,118, 42, 60, 27, 73,
        118,125, 34, 15, 65,115, 84, 64, 62, 81, 70,  1, 24,111,121, 83,
        104, 81, 49,127, 48,105, 31, 10,  6, 91, 87, 37, 16, 54,116,126,
         31, 38, 13,  0, 72,106, 77, 61, 26, 67, 46, 29, 96, 37, 61, 52,
        101, 17, 44,108, 71, 52, 66, 57, 33, 51, 25, 90,  2,119,122, 35
    }, table_2[128] = {
         52, 50, 44,  6, 21, 49, 41, 59, 39, 51, 25, 32, 51, 47, 52, 43,
         37,  4, 40, 34, 61, 12, 28,  4, 58, 23,  8, 15, 12, 22,  9, 18,
         55, 10, 33, 35, 50,  1, 43,  3, 57, 13, 62, 14,  7, 42, 44, 59,
         62, 57, 27,  6,  8, 31, 26, 54, 41, 22, 45, 20, 39,  3, 16, 56,
         48,  2, 21, 28, 36, 42, 60, 33, 34, 18,  0, 11, 24, 10, 17, 61,
         29, 14, 45, 26, 55, 46, 11, 17, 54, 46,  9, 24, 30, 60, 32,  0,
         20, 38,  2, 30, 58, 35,  1, 16, 56, 40, 23, 48, 13, 19, 19, 27,
         31, 53, 47, 38, 63, 15, 49,  5, 37, 53, 25, 36, 63, 29,  5,  7
    }, table_3[64] = {
          1,  5, 29,  6, 25,  1, 18, 23, 17, 19,  0,  9, 24, 25,  6, 31,
         28, 20, 24, 30,  4, 27,  3, 13, 15, 16, 14, 18,  4,  3,  8,  9,
         20,  0, 12, 26, 21,  8, 28,  2, 29,  2, 15,  7, 11, 22, 14, 10,
         17, 21, 12, 30, 26, 27, 16, 31, 11,  7, 13, 23, 10,  5, 22, 19
    }, table_4[32] = {
         15, 12, 10,  4,  1, 14, 11,  7,  5,  0, 14,  7,  1,  2, 13,  8,
         10,  3,  4,  9,  6,  0,  3,  2,  5,  6,  8,  9, 11, 13, 15, 12
    }, *table[5] = { table_0, table_1, table_2, table_3, table_4 };

/*
 * This code derived from a leaked document from the GSM standards.
 * Some missing pieces were filled in by reverse-engineering a working SIM.
 * We have verified that this is the correct COMP128 algorithm.
 *
 * The first page of the document identifies it as
 *       _Technical Information: GSM System Security Study_.
 *       10-1617-01, 10th June 1988.
 * The bottom of the title page is marked
 *       Racal Research Ltd.
 *       Worton Drive, Worton Grange Industrial Estate,
 *       Reading, Berks. RG2 0SB, England.
 *       Telephone: Reading (0734) 868601   Telex: 847152
 * The relevant bits are in Part I, Section 20 (pages 66--67).  Enjoy!
 *
 * Note: There are three typos in the spec (discovered by
 * reverse-engineering).
 * First, "z = (2 * x[n] + x[n]) mod 2^(9-j)" should clearly read
 * "z = (2 * x[m] + x[n]) mod 2^(9-j)".
 * Second, the "k" loop in the "Form bits from bytes" section is severely
 * botched: the k index should run only from 0 to 3, and clearly the range
 * on "the (8-k)th bit of byte j" is also off (should be 0..7, not 1..8,
 * to be consistent with the subsequent section).
 * Third, SRES is taken from the first 8 nibbles of x[], not the last 8 as
 * claimed in the document.  (And the document doesn't specify how Kc is
 * derived, but that was also easily discovered with reverse engineering.)
 * All of these typos have been corrected in the following code.
 */

void A3A8(/* in */ Byte rand[16], /* in */ Byte key[16],
        /* out */ Byte simoutput[12])
{
        Byte x[32], bit[128];
        int i, j, k, l, m, n, y, z, next_bit;
```

```
        /* ( Load RAND into last 16 bytes of input ) */
        for (i=16; i<32; i++)
                x[i] = rand[i-16];

        /* ( Loop eight times ) */
        for (i=1; i<9; i++) {
                /* ( Load key into first 16 bytes of input ) */
                for (j=0; j<16; j++)
                        x[j] = key[j];
                /* ( Perform substitutions ) */
                for (j=0; j<5; j++)
                        for (k=0; k<(1<<(1<<(4-j)); l++) {
                                        m = l + k*(1<<(5-j));
                                        n = m + (1<<(4-j));
                                        y = (x[m]+2*x[n]) % (1<<(9-j));
                                        z = (2*x[m]+x[n]) % (1<<(9-j));
                                        x[m] = table[j][y];
                                        x[n] = table[j][z];
                                }
                /* ( Form bits from bytes ) */
                for (j=0; j<32; j++)
                        for (k=0; k<4; k++)
                                bit[4*j+k] = (x[j]>>(3-k)) & 1;
                /* ( Permutation but not on the last loop ) */
                if (i < 8)
                        for (j=0; j<16; j++) {
                                x[j+16] = 0;
                                for (k=0; k<8; k++) {
                                        next_bit = ((8*j + k)*17) % 128;
                                        x[j+16] |= bit[next_bit] << (7-k);
                                }
                        }
        }

        /*
         * ( At this stage the vector x[] consists of 32 nibbles.
         *   The first 8 of these are taken as the output SRES. )
         */

        /* The remainder of the code is not given explicitly in the
         * standard, but was derived by reverse-engineering.
         */

        for (i=0; i<4; i++)
                simoutput[i] = (x[2*i]<<4) | x[2*i+1];
        for (i=0; i<6; i++)
                simoutput[4+i] = (x[2*i+18]<<6) | (x[2*i+18+1]<<2)
                                | (x[2*i+18+2]>>2);
        simoutput[4+6] = (x[2*6+18]<<6) | (x[2*6+18+1]<<2);
        simoutput[4+7] = 0;
}


#ifdef TEST
int hextoint(char x)
{
        x = toupper(x);
        if (x >= 'A' && x <= 'F')
                return x-'A'+10;
        else if (x >= '0' && x <= '9')
                return x-'0';
        fprintf(stderr, "bad input.\n");
        exit(1);
}

int main(int argc, char **argv)
{
        Byte key[16], rand[16], simoutput[12];
        int i;

        if (argc != 3 || strlen(argv[1]) != 34 || strlen(argv[2]) != 34
                        || strncmp(argv[1], "0x", 2) != 0
                        || strncmp(argv[2], "0x", 2) != 0) {
                fprintf(stderr, "Usage: %s 0x 0x\n", argv[0]);
                exit(1);
        }
```

```
        for (i=0; i<16; i++)
                key[i] = (hextoint(argv[1][2*i+2])<<4)
                        | hextoint(argv[1][2*i+3]);
        for (i=0; i<16; i++)
                rand[i] = (hextoint(argv[2][2*i+2])<<4)
                         | hextoint(argv[2][2*i+3]);
        A3A8(key, rand, simoutput);
        printf("simoutput: ");
        for (i=0; i<12; i++)
                printf("%02X", simoutput[i]);
        printf("\n");
        return 0;
}
#endif
```