# GSM network and its privacy - the A5 stream cipher

Thomas Stockinger *

November 2005

*Abstract:* This paper shows the basic mechanisms of the GSM cellular network to protect security and privacy. The A5 stream cipher is described in detail in both variations A5/1 and A5/2, with a short introduction of the required A8 cipher and the similar A3 cipher. A summary of major cryptanalysises on the A5 cipher is presented, followed by ideas and personal opinions about the practical approach of attacks.

**Keywords:** GSM, A3, A8, A5, privacy, security, encryption, cipher

## 1 Introduction

The main concept of wireless communication is data transfer over-the-air – a media that cannot be shielded by means of hardware from eavesdropping and intrusion. Hence other ways have to be established to protect security and privacy, like cryptography. The GSM network is the biggest cellular network nowadays, and its specifications include such means: The A3 algorithm for authentication, A8 for key generation and A5 for data encryption. All these algorithms are relatively weak and therefore have successfully been attacked in the past. This paper presents an overview of cryptography in the GSM network and its hazards.

In section 2 a short introduction to the main components and operations of the GSM network is given, followed by an explanation of the authentication and key generation mechanisms in section 3: The A3 algorithm is used for authentication of the subscriber, whereas the very similar algorithm A8 generates the session key for the A5 encryption cipher. A5 has two variations: The "stronger" A5/1 and the "weaker" A5/2 which are fully described in section 3.3. Both variations have a common way of encrypting data by encoding plaintext/decoding cyphertext with an XOR operation with pseudo random bits. The algorithm to generate these pseudo random bits is slightly different in A5/1

*E-mail: Thomas.Stockinger@nop.at

and A5/2, though both use linear feedback shift registers and irregular clocking of these registers.

A5 has been cryptanalysed by experts and different weaknesses have been found which are stated in section 4, giving the practical side of those attacks as well as a short summary of major publications on A5 which can be found on the internet.

The paper concludes with an emphasis on the inadequacy of current cryptographical algorithms in the GSM network, but also considers the fact that the effort of successfully attacking the network is beyond the means of most people, thus reducing the overall-threat.

## 2 The GSM network

Mobile communication in general and mobile telephone systems in particular are a booming market. With over 200 participating countries the "Global System for Mobile Communcation" (GSM) is the largest mobile phone network. Since its first commercial operation in 1991 it has grown to over 1.5 billion customers worldwide. Starting 1982 the "Groupe Spécial Mobile"[1] designed GSM to be a cellular, digital network.

Compared to previous commercial (analog) wireless telephone networks, this cellular, digital design has advantages like better utilization of radio frequencies (using timeslots and channel-hopping) providing more capacity, the ability to hand over calls between base-stations during active communication, less power-usage with adaptive regulation depending on distance and signal quality which resulted in smaller handsets, digital error correction, improved audio-codecs[2] for better audio quality, and usage of Subscriber Identity Modules (SIM) for switching devices but keeping the subscriber's identity.

The following paragraphs present the most important components of a GSM network and their names with abbreviations which are later used in this paper:

A Mobile Station Equipment (MSE) can be any kind of device (in most cases a mobile phone or datacard) which communicates over-the-air with Base Transceiver Stations (BTS). Those BTSs are arranged around cells (hence the name "cellular network"). Through directional antennas the area around a BTS is sectorized so it can serve all adjacent cells from one location. Each BTS communicates with a Base Station Controller (BSC) and together they form the Base Station Subsystem (BSS). The BSC is attached to the Mobile services Switching Centre (MSC) which finally handles all management and communication (inside and outside) of the system. Three databases are linked to the MSC: The Home Location Register (HLR), the Authentication Centre (AUC), and the Visitors Location Register (VLR). Together they form the Network SubSystem (NSS).

Those databases store vital information: VLR is a temporary list of subscribers which are currently roaming inside the network. The AUC authenticates a subscriber (or in fact his SIM) when connecting to the network, and the HLR stores information to uniquely identify the subscriber and the assigned telephone number.

[1] Originally hosted by the "European Conference of Postal and Telecommunications Administrations" (CEPT), later transferred to the "European Telecommunications Standards Institute" (ETSI)

[2] An audio-codec is responsible for digitizing and compressing analog audio signals.

The company which runs a GSM network is called "operator" or "provider" - the customers are the "subscribers" [2, 1, 3].

# 3 Security and privacy on GSM

Nowadays we know how important security and privacy in communication systems is, especially when using wireless networks which transmit data over the air and therefore cannot be shielded from unwanted intruders and listeners. But during the design of the GSM network from 1982 to 1991 only a moderate level of security was finally specified, as described in the following. The digital communication of GSM allowed usage of cryptographic algorithms that directly en- and de-code digital data streams and are implemented by discrete hardware components.

GSM utilises cryptographic algorithms for three purposes: [4]

| Purpose | Algorithm | Variations |
|---|---|---|
| Authentication | A3 | COMP128 COMP128-2 COMP128-3 COMP128-4 |
| Key generation | A8 | COMP128 COMP128-2 COMP128-3 COMP128-4 |
| Encryption | A5 | A5/0 A5/1 A5/2 |

None of these algorithms has ever been officially published ("Security Through Obscurity"), though all were later either discovered through leaking of documents or reverse engineering of MSEs' firmware [9, 6]. Subsequent analysis by cryptographical experts revealed possible attacks which (almost) render these algorithms virtually useless [12, 13, 14, 15, 16]. But it also lead to the implementation of stronger ciphers which were again not officially released to the public for review.

## 3.1 A3 - Authentication

The A3 algorithm is used to generate a signed response which is sent from MSE to BTS to authenticate the identity of the MSE [4]:
The MSE retrieves the 32-bit Signed Response (SRES) by issuing a command to the SIM. This command includes the 128-bit random challenge (RAND) generated by the Home Location Register (HLR). The SIM uses the RAND, its 128-bit Individual Subscriber Authentication Key ($K_i$) and the A3 algorithm to calculate a 128-bit response which is returned to the MSE, then passed on to the BTS, MSC and finally verified by the AUC. Only the first 32 bits are used as SRES. A3 is completely implemented in the SmartCard, so $K_i$ never leaves the SIM.

Most GSM networks use a version of the COMP128 algorithm as implementation of A3. Because of a leaked document the first version of COMP128 was made public in 1997 [6] and shortly later was successfully attacked. With improved attacks it is possible to extract the $K_i$ of a COMP128-SIM in less than a minute nowadays, given physical access to the SIM and knowing the PIN. The extracted $K_i$ can then be used to break authentication security (e.g. clone someones SIM) [5]. Therefore GSM network providers have switched to COMP128-2, COMP128-3 and COMP128-4 (for 3G networks) algorithms which are secret and have not yet been subject to cryptanalysis.

## 3.2 A8 - Key generation

The key generation algorithm A8 is very similar to A3. In fact the same COMP128 algorithm [6] is used to create the 64-bit ciphering key ($K_c$) which is subsequently used in A5 [4]:
Taking the 128-bit RAND received from the Mobile Services Switching Center (MSC) and the 128-bit $K_i$ stored in the SIM as input, A8 calculates 128 bits of output. Figure 1 shows a schema of this data flow. On production systems only 54 bits are used as ciphering key $K_c$, the remaining 10 bits of $K_c$ are zeroed[3]. The same key $K_c$ stays active until the MSC decides to request a new one, which rarely happens and is therefore an issue concerning attacks.

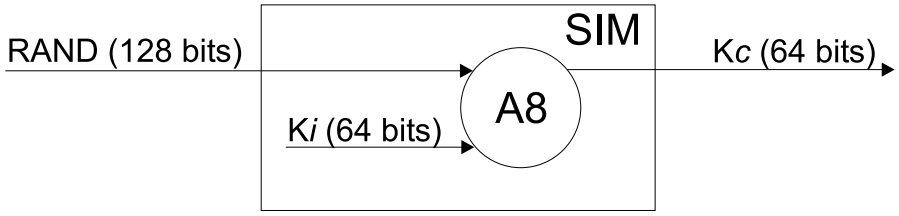


Figure 1: Data flow of A8

The COMP128 algorithm will not be further presented here. For details see the implementation written in C which was published by Marc Briceno, Ian Goldberg and David Wagner [6] in 1998 and also the presentation by Bill Brumley [5].

## 3.3 A5 - Encryption

To protect privacy all over-the-air transmissions on a GSM network are encrypted with a stream cipher [23, para.11-13] known as A5. Four different variants of the algorithm exist: A5/0 is a no-operation cipher which does not encrypt data [16, p.13]. A5/1 is the standard version and was specified in the mid 1980's after a dispute between several NATO countries about the strength of the algorithm: Germany wanted it to be strong because of its long borders to Eastern Europe, but was later overruled by the other countries and a relatively simple design for A5/1 was specified [9]. A5/2 is a weakened version which was chosen to deal with export restrictions on strong ciphers [17, p.1]. A5/3 was later added for 3G networks (UMTS - successor to GSM) and is a totally new algorithm based on the clock cipher KASUMI by Mitsuru Matsui [18] who designed KASUMI (also named MISTY) to be resistant against differential and linear cryptanalysis [24].

Though the A5 algorithm is described in the specifications of GSM it has never been made public officially. Companies implementing GSM networks have to buy the

[3]This is most likely an implementation flaw, not a design flaw. This will again be referred to in section 4

GSM specifications from ETSI [1], most likely accompanied by strong non-disclosure-agreement contracts. Through leaking of documents a first draft of the algorithm was made public by Ross Anderson [9] in 1994 and fully discovered through reverse engineering of a mobile phone's firmware by Brienco citebgwa5 in 1998/1999, and even later confirmed by the GSM group to be correct [13]. Presumably all technical papers of cryptoanalysis (see section 4.1) refer to those sources when explaining the A5 algorithm, whereas printed literature (about the GSM network) seem to avoid this subject, probably because of possible copyright infringements.

A5 is a stream cipher[4]. It operates on 228-bit blocks called "frames" sent and received over the air every 4.6 milliseconds. 114 bits represent data sent from the MSE and the other 114 bits are data received by the MSE, both mainly containing digitized audio signals (after error correction [17, p.1]). Taking the session key $K_c$ produced by A8 and a frame counter[5] $F_n$, A5 generates 228 pseudo random bits (PRAND) which are XOR'ed with the plaintext frame resulting in 228 bits of ciphertext. A schema of this data flow is shown in figure 2.

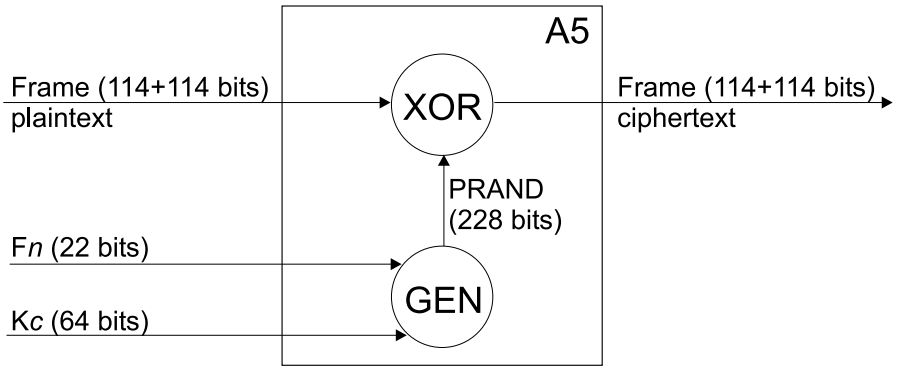


Figure 2: Data flow of A5

The most important part in A5 is generating the pseudo random bits (function GEN in figure 2). In A5/0, as a no-op cipher, the PRAND is generated by negating the input frame. Or in other words, the XOR function is left out, using the input frame as output.

### 3.3.1 A5/1

A5/1 implements PRAND generation by 3 linear feedback shift registers[6] (LFSRs) denoted as $R_1$, $R_2$ and $R_3$. In this case an LFSR feedback function is an XOR of all its

[4] "*In cryptography, a stream cipher is a symmetric cipher in which the plaintext digits are encrypted one at a time, and in which the transformation of successive digits varies during the encryption.*" – Wikipedia [7, top]

[5] The frame counter is initialized with 0 at conversation-start and incremented by 1 (mod $2^{22}$) with each frame sent.

[6] An LFSR uses some of its bits as input for a feedback function storing its result in the LSB and produces 1 bit output (the MSB) when it is clocked [8]

input bits[7], meaning that when the register is clocked, its input bits are XOR'ed and the result is stored in the rightmost bit. The registers are defined as follows [10, 11]:

| Register | Length | Characteristic polynomial | Clocking bit[8] | Input bits index |
|---|---|---|---|---|
| $R_1$ | 19 | $x^{19} + x^5 + x^2 + x + 1$ | $R_1[8] = C_1$ | 13, 16, 17, 18 |
| $R_2$ | 22 | $x^{22} + x + 1$ | $R_2[10] = C_2$ | 20, 21 |
| $R_3$ | 23 | $x^{23} + x^{15} + x^2 + x + 1$ | $R_3[10] = C_3$ | 7, 20, 21, 22 |

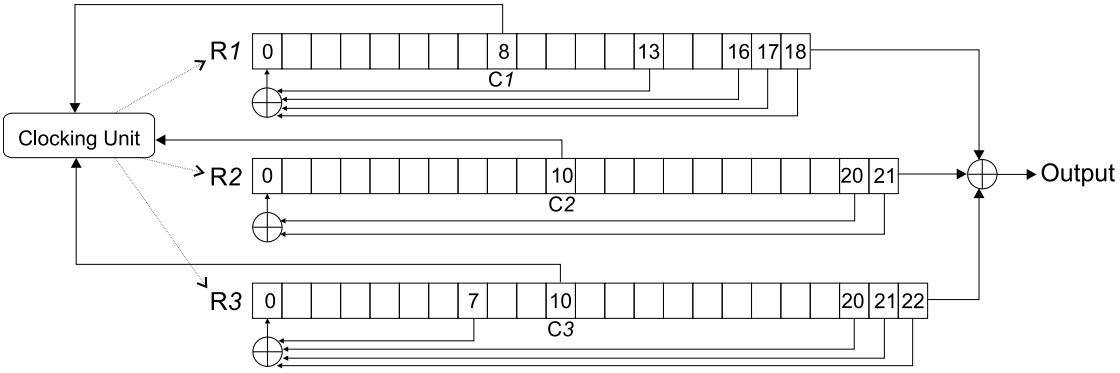


Figure 3: Schema of A5/1 PRAND generation

Steps to generate PRAND with A5/1 (see figure 3):

- All registers are zeroed.
- The registers are clocked for 64 cycles. In each cycle $i$ ($0 \leq i \leq 63$) the bit $K_c[i]$ is XOR'ed with the input bits of the register and stored in the LSB of the same register.
- The registers are clocked for 22 cycles. In each cycle $i$ ($0 \leq i \leq 21$) the bit $F_n[i]$ is XOR'ed with the input bits of the register and stored in the LSB of the same register.
- The next 100 cycles are run to diffuse $K_c$ and $F_n$ into the registers, discarding the output. An irregular clocking is applied: Whether a register is clocked or not is determined during each cycle by the *Clocking Unit* calculating the majority of all 3 clocking bits $C_{1-3}$ - if the majority matches the clocking bit, the register is clocked:

[7] In most publications referred to as "tap bit" or "tapped bits", whereas the simple name "input bits" is preferred in this paper.

[8] $R_x[y]$ denotes the bit on index $y$ of register $R_x$ where $R_x[0]$ is the LSB (leftmost bit). This is the common notation in [11, 12, 13, 14, 15, 16]

| $C_1$ | $C_2$ | $C_3$ | Majority | $R_1$ | $R_2$ | $R_3$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | clock | clock | clock |
| 1 | 0 | 0 | 0 | | clock | clock |
| 0 | 1 | 0 | 0 | clock | | clock |
| 1 | 1 | 0 | 1 | clock | clock | |
| 0 | 0 | 1 | 0 | clock | clock | |
| 1 | 0 | 1 | 1 | clock | | clock |
| 0 | 1 | 1 | 1 | | clock | clock |
| 1 | 1 | 1 | 1 | clock | clock | clock |

The registers are now initialized with values and are ready to produce output. This is called the *initial state*.

- The next 228 cycles are again carried out with the same irregular clocking as in the previous step. In each cycle $i$ ($0 \leq i \leq 227$) the MSBs of all 3 registers are XOR'ed and the result is used as bit $i$ of the PRAND.

The resulting PRAND is then ready to be used on the frame as shown in 2. For the next frame, a new PRAND is generated, and so on.

### 3.3.2 A5/2

A5/2 also uses 3 LFSRs and a fourth LFSR $R_4$ is introduced:

| Register | Length | Characteristic polynomial | Clocking bits | Input bits index |
|---|---|---|---|---|
| $R_1$ | 19 | $x^{19} + x^5 + x^2 + x + 1$ | none | 13, 16, 17, 18 |
| $R_2$ | 22 | $x^{22} + x + 1$ | none | 20, 21 |
| $R_3$ | 23 | $x^{23} + x^{15} + x^2 + x + 1$ | none | 7, 20, 21, 22 |
| $R_4$ | 17 | $x^{17} + x^5 + 1$ | $R_4[3]$, $R_4[7]$, $R_4[10]$ | 11, 16 |

A5/2 works similar to A5/1 by also using XOR as LFSR feedback function and an irregular clocking. Differences between A5/2 and A5/1 are:

- A fourth register.
- A slightly different initialization phase.
- The input bits for the *Clocking Unit* are taken from $R_4$.
- The output is an XOR of
  1. the MSB of $R_1$, $R_2$ and $R_3$
  2. the majority of 3 bits of each register with one of these 3 bits negated:

| Register | MSB | Majority bits | Negated majority bit |
|---|---|---|---|
| $R_1$ | $R_1[18]$ | $R_1[12]$, $R_1[15]$ | $R_1[14]$ |
| $R_2$ | $R_2[21]$ | $R_2[9]$, $R_2[13]$ | $R_2[16]$ |
| $R_3$ | $R_3[22]$ | $R_3[16]$, $R_3[18]$ | $R_3[13]$ |

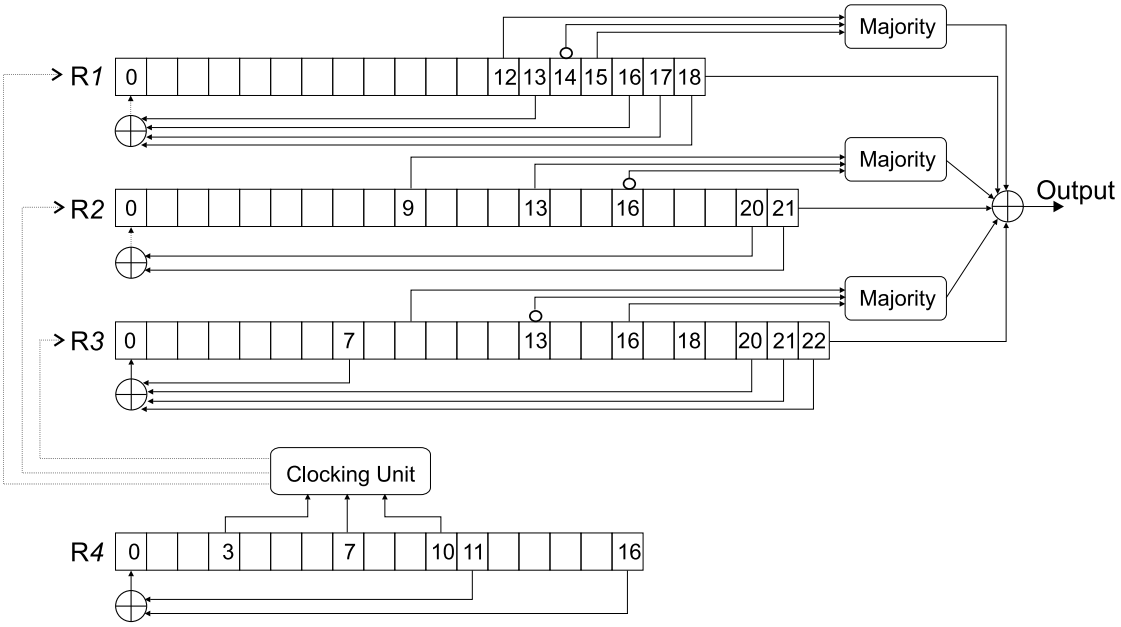


Figure 4: Schema of A5/2 PRAND generation

Steps to generate PRAND with A5/2 (see figure 4):

- All registers are zeroed.
- The registers are clocked for 64 cycles. In each cycle $i$ ($0 \leq i \leq 63$) the bit $K_c[i]$ is XOR'ed with the LSB of the register and stored in the LSB of the same register.
- The registers are clocked for 22 cycles. In each cycle $i$ ($0 \leq i \leq 21$) the bit $F_n[i]$ is XOR'ed with the LSB of the register and stored in the LSB of the same register.
- Set bits $R_1[15]$, $R_2[16]$, $R_3[18]$, $R_4[10]$ to 1.
- The next 99 cycles are run to diffuse $K_c$ and $F_n$ into the registers, discarding the output. Irregular clocking is applied: Whether a register is clocked or not is determined during each cycle by the *Clocking Unit* calculating the majority of all 3 clocking bits $R_4[3]$, $R_4[7]$, $R_4[10]$ - if the majority matches the clocking bit, the corresponding register is clocked:

| $R_4[3]$ | $R_4[7]$ | $R_4[10]$ | Majority | $R_1$ | $R_2$ | $R_3$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | clock | clock | clock |
| 1 | 0 | 0 | 0 | clock | | clock |
| 0 | 1 | 0 | 0 | clock | clock | |
| 1 | 1 | 0 | 1 | | clock | clock |
| 0 | 0 | 1 | 0 | | clock | clock |
| 1 | 0 | 1 | 1 | clock | clock | |
| 0 | 1 | 1 | 1 | clock | | clock |
| 1 | 1 | 1 | 1 | clock | clock | clock |

Register $R_4$ is always clocked last in every cycle.

- The next 228 cycles are again carried out with the same irregular clocking as in the pevious step. Each cycle $i$ ($0 \leq i \leq 227$) the MSBs of all 3 registers are XOR'ed and the result is used as bit $i$ of the PRAND.

# 4 Attacks on A5

Soon after the A5 algorithm was made public and reviewed by experts it was evident that the encryption can be attacked and broken by (relatively) easy methods. Classic cryptanalysis [24] revealed that the generated key ($K_c$) was the main flaw of the design:

- The algorithm (A5/1, A5/2) was not designed with modern cryptographical knowledge. Its maximal exhaustive search complexity is (only) $2^{64}$.
- It is generated only once after the MSE registers with the network and stays active for all communication, until the MSC requests a new one or the MSE deregisters [4, 17].
- It is artificially shortened in deployed systems by almost 16% when zeroing 10 bits, reducing the search to $2^{54}$.
- Encryption is applied *after* error correction.

## 4.1 Attacks in theory

The early, imprecise description of A5/1 was first cryptanalized by Golić [12] introducing a basic divide-and-conquer attack that recovers the unknown initial state from a known keystream sequence by exploiting the simple clocking rule. With an average of $2^{40}$ trial encryptions (instead of $2^{64}$ searches with full key length) this attack could in fact be performed on a high-end workstation taking weeks, or in shorter time by specialized hardware. Golić then enhanced his analisys with a time-memory tradeoff attack based on the birthday paradox.

After a more precise description of A5/1 was available and confirmed to be correct, Biryukov, Shamir and Wagner [13] published a cryptanalysis based on the previous work of Golić. They introduced two related attacks which made it possible to decrypt an A5/1

data stream in real time. Both attacks require between $2^{42}$ and $2^{48}$ preprocessing steps and between 146GB and 292GB of data storage. After the preprocessing phase has been performed once, the biased birthday attack requires only around $2^{15}$ frames of data (two minutes of a voice conversation) and one second of processing time to calculate the key $K_c$ which is used during this and most likely also during the next conversations. The random subgraph attack works similar, requiring only 2 seconds of data (roughly $2^9$ frames), but several minutes of after-processing.

These previous attacks on A5/1 have a complexity which is exponential with the length of shift registers. By increasing the length these attacks can no longer be preformed in a reasonable time-frame. Ekdahl and Johansson [15] then introduced a cryptanalysis based on correlation attacks which is (almost) independent of shift registers length. Its complexity correlates with the number of cycles the registers are clocked before the *initial state* is reached and before actually producing output bits. With the current number of 100 cycles this attack breaks A5/1 in a few minutes with only 2 to 5 minutes of conversation data as input.

The A5/2 algorithm (the "weaker" variant) was reverse engineered by Goldberg, Wagner and Green and immediately cryptanalyzed, revealing a possible known-plaintext attack. It requires the difference of two given (plaintext) frames which are roughly 6 seconds apart. The average computation cost is about $2^{16}$ dot products of 114-bit vectors. But this attack also requires bit $R_4[11]$ to be zero in the initial state of A5/2 and therefore statistically fails every second case [16, p.2].

Petrović and Fúster-Sabater [14] later introduced an attack on A5/2 which is of algebraic nature. By solving a system of linearized equations which respresents the output bits it can predict the following output of A5/2 with only a few hundred known ciphertext bits. Opposed to previous attacks the key $K_c$ is *not* compromised.

The instant ciphertext-only cryptanalysis of A5/2 by Barkan, Biham and Keller [16] is a very practical approach which requires only a few encrypted frames from a conversation to find $K_c$ with a time complexity of about $2^{16}$ dot products. They first present a new known-plaintext attack based on solving an optimized linearized system of equations and a relatively small preprocessing phase. This attack is then converted to a ciphertext-only attack by taking advantage of the error correction codes implemented in GSM data communications which offers highly structured redundancy. Finally they introduce new ideas where man-in-the-middle attacks are actively forcing a MSE to use the weak A5/2 algorithm and then uses above cryptanalysis to retrive $K_c$ (which is the same key used in A5/1 or even A5/3 and thus may be used to attack later conversations of these ciphers). Another man-in-the-middle attack intercepts registering of the MSE to the network where capabilities of the MSE are reported to the network. By faking these data the network can be persuaded to only use A5/2 or even the no-op A5/0.

Barkan, Biham and Keller's work also includes an incomplete explanation of a passive ciphertext-only cryptanalysis of A5/1 and possible attack scenarios of passive and active attacks.

### 4.2 Practical approach of attacks

As there is proof that the encryption of GSM communication is relatively easy to attack in theory, one must not forget that a considerably amount of hardware is necessary to actually intercept GSM communications[9]. The hardware must at least consist of a radio receiver device which is capable of receiving and decoding digital data that is exchanged over-the-air. This is not an easy task considering the technics used by a GSM network as outlined in section 2. Secondly, the hardware has to implement a device for decrypting the digital A5 data stream by one of the attacks presented in section 4.1.

Hypothetically a simple GSM mobile phone already has all these capabilities (except the decrypting of an unknown A5 stream), so it might be possible to use such a phone for eavesdropping. Nevertheless a huge amount of know-how, time and money is needed. On the other hand, commercially available test equipment is available which "pretends" to be a cell phone and therefore could be used to intercept communication [17, p.2]

The more practical approach to e.g. listen to a cellphone conversation is by tapping the line where it is not encrypted[10]: Either on the cellphone's side, or right after it "leaves the air" inside the BSS or even the NSS. Governments generally have this option, but some are still trying to keep a hand on cryptographical encrypted communication [22].

Interestingly the majority of all GSM users seems not to be alarmed by the fact that their telephone conversations can be listened to or even hijacked: When the German operator E-Plus switched their network to be non-encrypted in 1999 for a short time [19], some people of the "Chaos Computer Club" deemed that to be a major scandal predicting a great commercial loss for E-Plus. Actually nothing happened and customers were barely interested. Not amazingly, because the everyday-users either rely on their privacy given by law or have nothing to hide anyway. Probably the biggest threat for such users is a nosy neighbour listening to their conversations, which is very improbable considering the necessary amount of hardware and knowledge as explained above.

## 5 Conclusion

*Every chain is only as strong as its weakest link.*

The huge market of GSM users is a big target which attracting all kinds of people who want to intercept mobile communications: Federal agencies, spys, criminals, nosy neighbours etc. The ETSI specification of the GSM network, mainly intended as recommendations for interoperability between different operators, adds methods to protect integrity and privacy of customers and also operators on a best-effort basis. But concluding from present knowledge companies that implemented the GSM infrastructure failed to interpret and translate those recommendations by using the provided examples directly in production-systems [16, p.2] and relying on restricted algorithms instead of

[9] In the early days of mobile telephony there was no encryption – eavesdropping of analog phones was possible through a simple and cheap radio-scanner.

[10] This so-called *wiretapping* is restricted by law in most countries and used by federal agencies only after authorisation by court [21].

inventing new, more secure methods. When those algorithms have eventually been published for peer-review and academically proven to be inadequate, new variations of the same theme are supposed to re-establish privacy [20]. No matter if it is very unlikely that GSM communications can be intercepted over-the-air, it must be noted that there are always other ways for organisations and individuals with power, money or authorization to do so, as long as no end-to-end encryption[11] with stronger ciphers is used.

## References


[1] ETSI: *European Telecommunications Standards Institute,* `http://www.etsi.org/` (viewed 2005)

[2] Lawrence Harte: *Introduction to GSM: Physical Channels, Logical Channels, Network, and Operation,* Althos (Nov 2004)

[3] Wikipedia: *Global System for Mobile Communications,* `http://en.wikipedia.org/wiki/GSM` (viewed 2005)

[4] Will Spencer: *GSM Security Website,* FAQ, `http://www.gsm-security.net/gsm-security-faq.shtml` (viewed 2005)

[5] Billy Brumley: *T-79.514 Special Course on Cryptology, A3/A8 & COMP128,* `http://www.tcs.hut.fi/Studies/T-79.514/slides/S5.Brumley-comp128.pdf` (Nov 2004)

[6] Marc Briceno, Ian Goldberg, David Wagner: *An implementation of the GSM A3A8 algorithm,* `http://www.iol.ie/~kooltek/a3a8.txt` (1998)

[7] Wikipedia, Matt Crypto et al.: *Stream cipher,* `http://en.wikipedia.org/wiki/Stream_cipher` (viewed 2005)

[8] Wikipedia: *Linear feedback shift register,* `http://en.wikipedia.org/wiki/LFSR` (viewed 2005)

[9] Ross Anderson: *The GSM cipher,* `http://groups.google.com/group/sci.crypt/msg/ba76615fef32ba32` (1994)

[10] Marc Briceno, Ian Goldberg, David Wagner: *A pedagogical implementation of A5/1,* `http://www.gsm-security.net/papers/a51.shtml` (1998)

[11] Wikipedia, Matt Crypto et al.: *A5/1,* `http://en.wikipedia.org/wiki/A5/1` (viewed 2005)

[12] Jovan Dj. Golić: *Cryptanalysis of Alleged A5 Stream Cipher,* Springer-Verlag (1998)


[11] From one cellular device to another including every line between that could be tapped.

[13] Alex Biryukov, Adi Shamir, David Wagner: *Real Time Cryptanalysis of A5/1 on a PC,*
http://cryptome.org/a51-bsw.htm (2000)

[14] Slobodan Petrović, Amparo Fúster-Sabater: *Cryptanalysis of the A5/2 algorithm,*
http://www.gsm-security.net/papers/a52.pdf (2000)

[15] Patrik Ekdahl, Thomas Johansson: *Another Attack on A5/1,* IEEE Transactions on Information Theory,
http://www.it.lth.se/patrik/papers/a5full.pdf (Jun 2001)

[16] Elad Barkan, Eli Biham, Nathan Keller: *Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication,*
http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2003/CS/CS-2003-05.ps.gz (2003)

[17] Greg Rose: *A precis of the new attacks on GSM encryption,* QUALCOMM Australia,
http://www.qualcomm.com.au/PublicationsDocs/GSM_Attacks.pdf (Sep 2003)

[18] Mitsuru Matsui, Toshito Tokita: *MISTY, KASUMI and Camellia Cipher Algorithm Development,* Mitsubishi Electric ADVANCE Vol100,
http://global.mitsubishielectric.com/pdf/advance/vol100/03Vol100_TR2.pdf (Dec 2002)

[19] Johannes Endres: *Unverschlüsselte E-Plus-Gespräche,* c't Magazin 16/1999 p. 34 (1999)

[20] ETSI: *GSM calls even more secure - thanks to new A5/3 Algorithm,* ETSI news release
http://www.etsi.org/pressroom/previous/2002/3algorithm.htm (July 2002)

[21] Answers.com: *Telephone tapping,*
http://www.answers.com/topic/wiretapping (viewed 2005)

[22] Dirk Fox: *Lauschordnung - Regierungen wehren sich gegen allzu private Kommunikation,* c't Magazin 17/1995 p. 72 (1995)

[23] Reinhard Wobst: *Harte Nüsse – Verschlüsselungsverfahren und ihre Anwendungen,* c't Magazin 17/2003 p. 200 (2003)

[24] Reinhard Wobst: *Durch die Hintertür – Methoden der Kryptanalyse,* c't Magazin 22/2003 p. 204 (2003)