Cryptanalyse de GSM

Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication*

Elad Barkan[1] Eli Biham[1] Nathan Keller[2]

[1] Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel

[2] Department of Mathematics

Technical Report CS-2003-05 - 2003

A3, A5, A8

There are three main types of cryptographic algorithms used in GSM:

- A5 is a **streamcipher** used for encryption
- A3 is an **authentication** algorithm
- A8 is the **key agreement** algorithm.

- The design of A3 and A8 is not specified in the specifications of GSM, only the external interface of these algorithms is specified.
- The exact design of the algorithm can be selected by the operators independently.
- However, many operators used the example, called COMP128, given in the GSM memorandum of understanding (MoU).

Cryptanalyse COMP128

Marc Briceno, Ian Goldberg, David Wagner

http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html

GSM Cloning

Here is some information on our GSM cloning results, starting at a very high level, and moving on eventually to detailed technical information, with data for the cryptographers and mathematicians at the end. This is joint work with Ian Goldberg (also of the ISAAC research group) and Marc Briceno (Director of the Smartcard Developers Association).

Briceno, Goldberg, Wagner

http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html

Important note added after publication: This article was released on April 13, 1998. This is the original version of that article (with no changes made other than this note), and is provided primarily for historical reasons. Please beware that some of our understanding about some details of the attack -- especially the possibility of over-the-air cloning -- has changed since when we wrote this note. We now feel that we understated the risk of over-the-air attacks in our initial announcement; based on new information, we have come to the conclusion that *over-the-air cloning must be considered a very real threat* which should not be ignored. Please see here for a more recent update.

A5/1 et A5/2

The description of A5 is part of the specifications of GSM, but it was never made public. There are two currently used versions of A5:

- A5/1 is the ``strong'' exportlimited version,
- A5/2 is the ``weak'' version that has no export limitations.

LFSR

$$maj(a,b,c) = a \cdot b \oplus b \cdot c \oplus c \cdot a$$

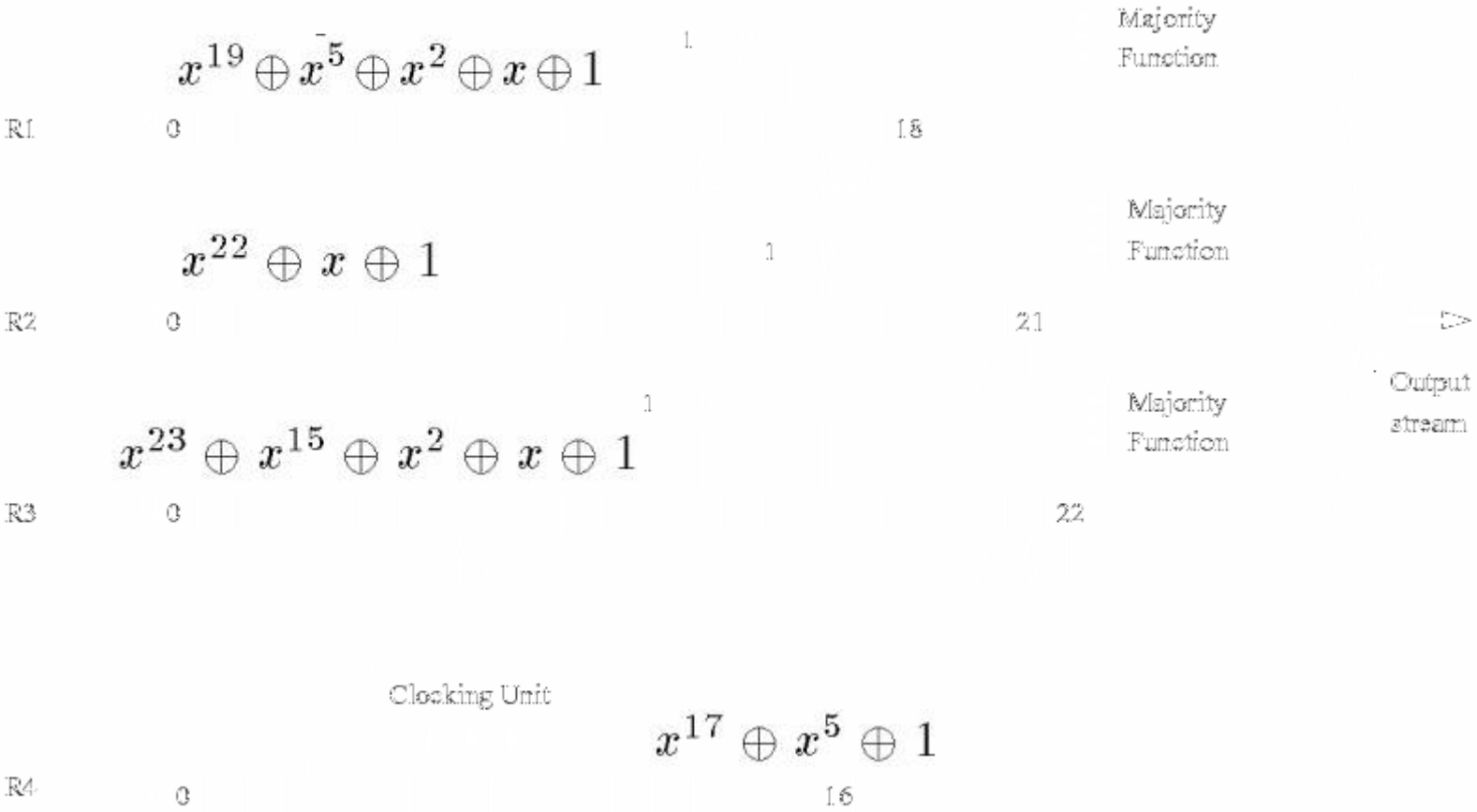
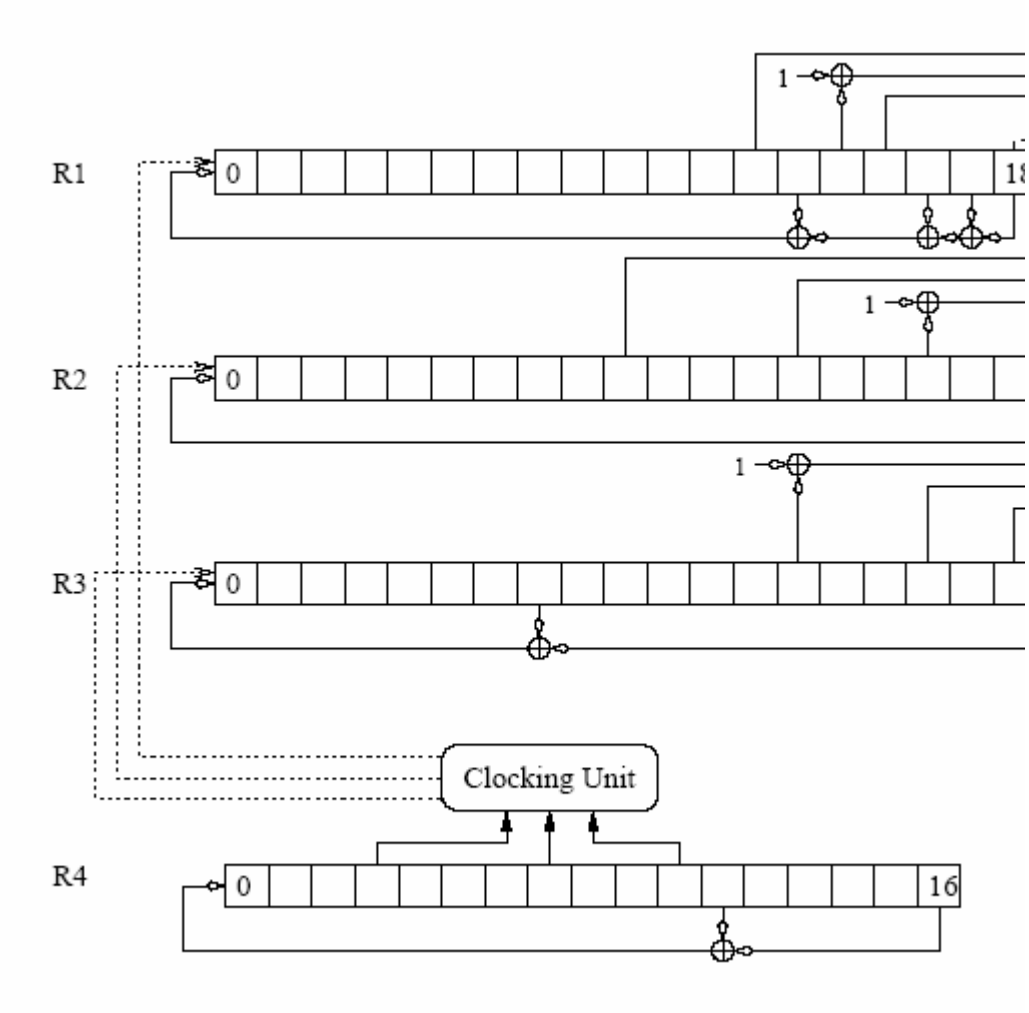


Fig. 1. The A5/2 internal structure

LFSR (2)



The clocking mechanism works as follows: R4 controls the clocking of R1, R2, and R3.

- When clocking of R1, R2, and R3 is to be performed, bits R4[3], R4[7], and R4[10] are the input of the clocking unit.
- The clocking unit performs a majority function on the bits.
- R1 is clocked if and only if R4[10] agrees with the majority.
- R2 is clocked if and only if R4[3] agrees with the majority.
- R3 is clocked if and only if R4[7] agrees with the majority.

After these clockings, R4 is clocked.

Initialisation

Initialisation à l'aide de la clef de chiffrement K_c (produite par A8), et du numéro de trame *f* :

- Set all LFSRs to 0 ($R1 = R2 = R3 = R4 = 0$).
- For $i := 0$ to 63 do
  1. Clock all 4 LFSRs.
  2. $R1[0] \leftarrow R1[0] \oplus K_c[i]$
  3. $R2[0] \leftarrow R2[0] \oplus K_c[i]$
  4. $R3[0] \leftarrow R3[0] \oplus K_c[i]$
  5. $R4[0] \leftarrow R4[0] \oplus K_c[i]$
- For $i := 0$ to 21 do
  1. Clock all 4 LFSRs.
  2. $R1[0] \leftarrow R1[0] \oplus f[i]$
  3. $R2[0] \leftarrow R2[0] \oplus f[i]$
  4. $R3[0] \leftarrow R3[0] \oplus f[i]$
  5. $R4[0] \leftarrow R4[0] \oplus f[i]$

Chiffrement du flot

The **keystream** generation is as follows:

1. Initialize the internal state with Kc and frame number.
2. Force the bits R1[15], R2[16], R3[18], R4[10] to be 1.
3. Run A5/2 for 99 clocks and ignore the output.
4. Run A5/2 for 228 clocks and use the output as keystream.

Attaque algébrique, à texte clair connu

Knowing the initial internal state of R1, R2, R3, R4, and the initial frame number, the session key can be retrieved using simple algebraic operations. This is mainly because the initialization process is **linear** in the session key and the initial frame number. Therefore, in the attack we focus on revealing the initial internal state of the registers.

Assume we know the initial state $R4_f$ of R4 at the first frame. An important observation is that R4 controls the clockings of the other registers. Since we know $R4_f$, then for each output bit we know the exact number of times that a register is clocked to produce that output bit. Each register has a linear feedback, therefore, once given the number of times a register is clocked, we can express every bit of its internal state as a **linear combination** of bits of the original internal state.

Attaque algébrique (2)

The output of A5/2 is an XOR of the last bits of R1, R2, and R3, and three majority functions of bits of R1, R2, R3. Therefore, the resulting function is **quadratic** with variables which are the bits in the **initial** state of these registers.

We take advantage of this low algebraic degree of the output. The goal in the next paragraphs is to express every bit of the whole output of the cipher (consisting of several frames) as a quadratic multivariate function in the initial state. Then, we construct an **overdefined** system of quadratic equations which expresses the keystream generation process and finally we solve it.

Attaque algébrique (3)

We can summarize this attack as follows: we try all the 2^{16} possible values for $R4_f$, and for each such value we solve the linearized system of equations that describe the output. The solution of the equations gives us a suggestion for the internal state of R1, R2, and R3. Together with R4, we have a suggestion for the full internal state. Most of the $2^{16} - 1$ wrong states are identified due to inconsistencies in the Gauss elimination. If more than one consistent internal state remains, these suggestions are verified by trial encryptions.

Attaque « à texte chiffré seul »

In this section we convert the attack of Section 3 on A5/2 to a ciphertext-only attack. We observe that error-correction codes are employed in GSM before encryption. Thus, the plaintext has a highly structured redundancy.

There are several kinds of error-correction methods that are used in GSM, and different error-correction schemes are used for different data channels. We focus on control channels, and specifically on the error-correction codes of the Slow Associated Control Channel (SACCH). We note that this error-correction code is the one used during the initialization of a conversation. Therefore, it suffices to focus on this code. Using this error-correction code we mount a ciphertext-only attack that recovers the key. However, we stress that the ideas of our attack can be applied to other error-correction codes as well.

In the SACCH, the message to be coded with error-correction codes has a fixed size of 184 bits. The result is a 456-bit long message. The 456 bits of the message are then interleaved, and divided to four frames. These frames are then encrypted and transmitted.

Code correcteur

The coding operation and the interleaving operation can be modeled together as one 456×184 matrix over $GF(2)$, which we denote by G. The message to be coded is regarded as a 184-bit binary vector, P. The result of the coding-interleaving operation is: $M = G \cdot P$. The resulting vector M consists of 4 frames.

Since G is a 456×184 binary matrix, there are $456 - 184 = 272$ equations that describe the kernel of the inverse transformation (and the dimension of the kernel is not larger than 272 due to the properties of the matrix G). In other words, for any vector M, $M = G \cdot P$, there are 272 linearly independent equations on its elements. Let K_G be a matrix that describes these 272 linear equations, i.e., $K_G \cdot M = 0$ for any such M.

Code correcteur (2)

We denote the output sequence bits of A5/2 for a duration of 4 frames by $k = k_j||k_{j+1}||k_{j+2}||k_{j+3}$, where $||$ is the concatenation operator. The ciphertext C is computed by $C = M \oplus k$. We use the same 272 equations on C, namely:

$$K_G \cdot C = K_G \cdot (M \oplus k) = K_G \cdot M \oplus K_G \cdot k = 0 \oplus K_G \cdot k = K_G \cdot k.$$

Since the ciphertext C is known, we actually get linear equations over elements of k. Note that the equations we get are independent of P — they depend only on k. We substitute each bit in k with its description as linear terms over V_f (see Section 3), and thus get equations on variables of V_f. Each 456-bit coding block, provides 272 equations. The rest of the details of the attack and its time complexity are similar to the case in the previous section, but in this attack we

etc.

Attaque de « l’homme du milieu »

This man-in-the-middle attack is performed as follows:

• When authentication is performed (in the initialization of a conversation), the network sends an authentication request to the attacker, and the attacker sends it to the victim.

•The victim computes SRES, and returns it to the attacker, which sends it back to the network. Now the attacker is ``authenticated'' to the network.

• Next, the network asks the customer to start encrypting with A5/1 or A5/3. In our attack, since the attacker **impersonates** the customer, the network actually asks the attacker to start encrypting with A5/1 or A5/3. The attacker does not have the key, yet, and therefore, is not able to start the encryption. The attacker needs the key before he is asked to use it.

« Homme du milieu » (2)

•To achieve it, the attacker asks the victim to encrypt with A5/2 just after the victim returned the SRES, and before the attacker returns the authentication information to the network.

• This request sounds to the victim as a legitimate request, since the victim sees the attacker as the network.

• Then, the attacker employs cryptanalysis of A5/2 to retrieve the encryption key of the A5/2 that is used by the victim.

• Only then, the attacker sends the authentication information (SRES) to the network.

• The key only depends on RAND, that means that the key recovered through the A5/2 attack is the same key to be used when A5/1 is used or even when 64bit A5/3 is used! Now the attacker can encrypt/decrypt with A5/1 or A5/3 using this key.

Classmark

A second possible attack, which can be relatively easily spotted (and prevented) by the network, is a **classmark** attack.

• During initialization of conversation, the mobile phone sends his ciphering capabilities to the network (this information is called classmark). Most mobile phones currently support A5/1, A5/2, and A5/0 (no encryption), but this may change from phone to phone, and can change in the future.

• The attacker changes (for example using a man-in-the-middle attack) the classmark information that the mobile phone sends, in a way that the network thinks that the mobile phone can only support A5/2, and A5/0.

• The network then **defaults to A5/2**, and thus allowing the attacker to listen to the conversation.

• This attack takes advantage of the following **protocol flaw** (faille): the classmark information is not protected.

Authentification fraîche

• Many networks initiate the authentication procedure rarely, and use the key created in the last authentication.

• An attacker can discover this key by impersonating the network to the victim mobile phone. Then the attacker initiates a radiosession with the victim, and asks the victim mobile phone to start encrypting using A5/2. The attacker performs the attack, recovers the key, and ends the radio session.

• The owner of the mobile phone and the network have no indication of the attack.

• The **leveraging** (effet de levier) in the first and last attacks relies on the fact that the same key is loaded to A5/2 and A5/1 and even to 64bit A5/3 (in case A5/3 is used in GSM, according to GSM standards). We note that although A5/3 can be used with key lengths of 64-128 bits, the GSM standard allows the use of only 64bit A5/3.

Attaque reportée

Call Wire-Tapping

• The most naive scenario that one might anticipate is **eavesdropping** conversations (« écouter aux portes », écouter une conversation privée)

.

• In another possible wire-**tapping** (mise sur écoute) attack the attacker records the encrypted conversation. The attacker must make sure that he knows the RAND value that created the used key (the RAND is sent unencrypted).

• At a later time, when it is convenient for the attacker, the attacker impersonates the network to the victim. Then the attacker initiates a GSM radiosession, asks the victim to perform authentication with the above RAND, and recovers the session key used in the recorded conversation.

• Once the attacker has the key he simply decrypts the (old) conversation and can listen to its contents.

•This attack has the advantage of transmitting only in the time that is convenient for the attacker. Possibly even years after the recording of the conversation, or when the victim is in another country, or in a convenient place for the attacker.

Détournement d’appel

Call Hijacking (**Call Theft --- Dynamic Cloning**)

• While a GSM network can perform authentication at the initiation of the call, encryption is the means of GSM for preventing impersonation at later stages of the conversation. The underlying assumption is that an imposter do not have Kc , and thus cannot conduct encrypted communications.

• Once an attacker has the encryption keys, he can **cut the victim off** the conversation, and impersonate the victim to the other party. Therefore, hijacking the conversation after authentication is possible.

• Hijacking can occur during early callsetup, even before the victim's phone begins to ring. The operator can hardly suspect there is an attack. The only **clue** (indice) of an attack is a moment of some increased electromagnetic interference.

• The victim's phone does not ring, and the victim has no indication that he is a victim. At least until his monthly bill arrives.