# Projet SFPN: Attaques sur le GSM

Aymeric BARBIN, Cyrille PRESSAT, Tayyib PATEL

UPMC

May 28, 2015

# Introduction

# Le GSM

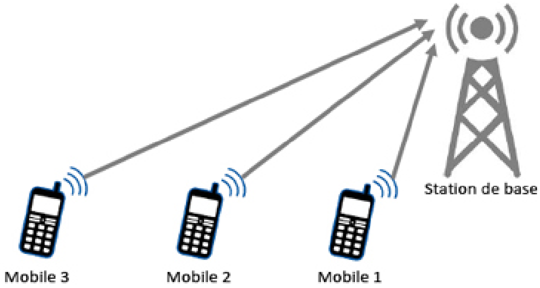



Figure: Global System for Mobile Communications

## Chiffrement symétrique

Chiffrement par flots

$$\text{Message} \oplus \text{SuiteChiffrante} = \text{Chiffré}$$

$$\text{Chiffré} \oplus \text{SuiteChiffrante} = \text{Message}$$

Utilise une famille de chiffrement A5

- A5/1
- A5/2
- A5/3

## Faiblesse

Attaque algébrique

$$\text{Message:}1011 \oplus \text{Variables:}XYZT=0$$

# Article de référence

Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication

Barkan,Biham,Keller

# LFSR

Un «clocking»:un décalage

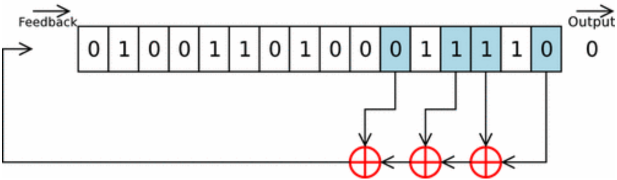



Figure: Linear Feedback Shift Register

# A5/2

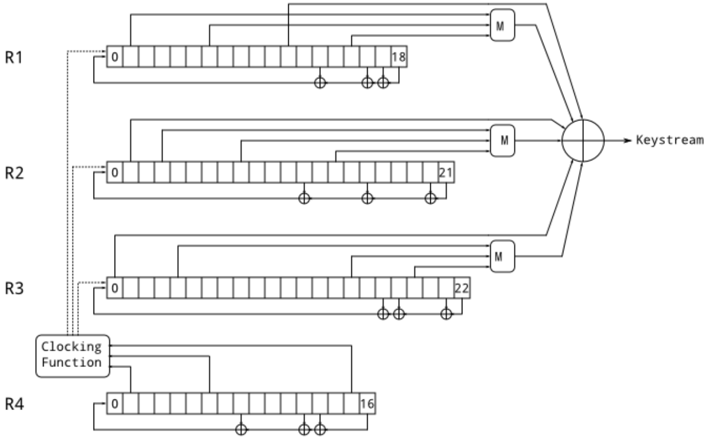
Fonctionnement:

- Une clé secrète (64 bits)
- Un IV(numero aléatoire, 22 bits)

$\rightarrow$ Une Keystream (240)

# Structure interne

4 LFSR connectés entre eux:

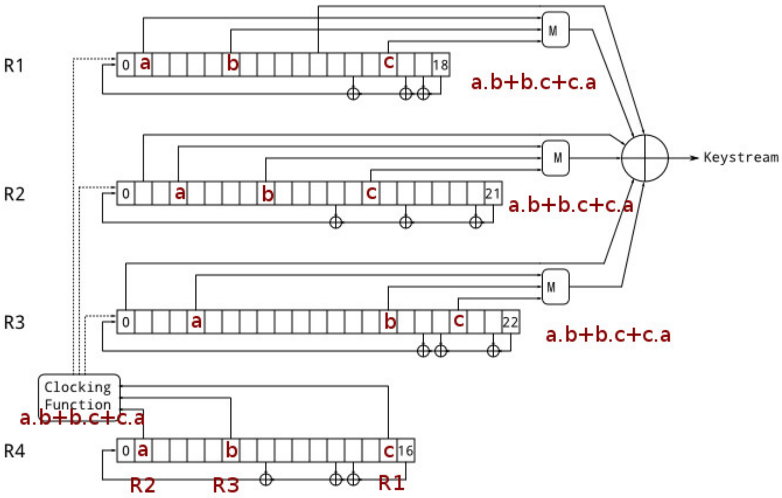


# Initialisation

1. Set $R1 = R2 = R3 = R4 = 0$.
2. For $i = 0$ to $63$
   - Clock all four registers.
   - $R1[0] \leftarrow R1[0] \oplus K_c[i]$; $R2[0] \leftarrow R2[0] \oplus K_c[i]$; $R3[0] \leftarrow R3[0] \oplus K_c[i]$; $R4[0] \leftarrow R4[0] \oplus K_c[i]$.
3. For $i = 0$ to $21$
   - Clock all four registers.
   - $R1[0] \leftarrow R1[0] \oplus f[i]$; $R2[0] \leftarrow R2[0] \oplus f[i]$; $R3[0] \leftarrow R3[0] \oplus f[i]$; $R4[0] \leftarrow R4[0] \oplus f[i]$.
4. Set the bits $R1[15] \leftarrow 1$, $R2[16] \leftarrow 1$, $R3[18] \leftarrow 1$, $R4[10] \leftarrow 1$.

Figure: Phase d'initialisation de A5/2.

# Phase de génération de la keyStream



# Résumé A5/2

**Phase 1:** Génération des registres (KeySetup)

Entrée: R1,R2,R3,R4, IV, clés secrète

Sortie: R1,R2,R3,R4 modifiés

**Phase 2:** Génération de la Keystream

Entrée: R1,R2,R3,R4

Sortie: Keystream

# Fonctionnement de l'attaque

- Attaque par recouvrement de clé
- Connaître l'état interne d' A5/2 $\Leftrightarrow$ Connaître la clé de session
- Attaque algébrique $\Leftrightarrow$ relation bits de sortie / bits registres d'entrée (codée sous MAGMA)

# Fonctionnement de l'attaque

Entrée:

- R4 (17 bits : $2^{17}$ possibilités, facilement récupérable)
- keyStream k récupérée sur le réseau

Sortie:

- Les 3 registres R1, R2 et R3 tels qu'ils se trouvaient juste après la KeySetup.

# Fonctionnement de l'attaque

## Génération d'équations

- Recherche de dépendances keyStream générée / registres utilisés
- R1' $= [x_1, ..., x_{19}]$, R2' $= [x_{20}, ..., x_{42}]$, R3' $= [x_{43}, ..., x_{64}]$
- Phase de génération de keyStream d'A5/2 à partir de R1', R2' R3' et R4
- KeyStream [1] : vecteur de polynômes de la forme

$$\sum_{i,j} x_j x_i \quad i, j \in 1, 64$$

[1] noté: kB

# Fonctionnement de l'attaque
## Génération d'équations

En particulier : kB[i] =k[i] $\forall i \in [1, len(k)]$[2]

Exemple:

$$\begin{pmatrix} x_1 + x_2.x_{34} \\ x_2.x_3 \\ x_3.x_5 \\ \vdots \\ x_{13}.x_{63} + x_6.x_{20} \\ x_{60}.x_{62} + x_1.x_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

[2] len(k)=len(kB)=taille de la keyStream considérée.

# Fonctionnement de l'attaque

## Génération d'équations

**ksolu= kB - k**

len(ksolu) équations pour 64 variables

Rajout de 64 équations triviales dans $F_2$:

$\forall x \in F_2$, $x^2$-x=0

# Fonctionnement de l'attaque

## Résolution des équations

Dans l'article: Linéarisation des équations

$$\begin{pmatrix} 1 & \dots & 0 & 1 & 0 & \dots \\ 0 & \dots & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} * \begin{pmatrix} x_1 \\ \vdots \\ x_{64} \\ x_2.x_{34} \\ x_2.x_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} x_1 + x_2.x_{34} \\ x_2.x_3 \\ \vdots \end{pmatrix}$$

# Fonctionnement de l'attaque

## Résolution des équations

Notre implémentation: Construction de la variété algèbrique[3] correspondant aux équations

Utilisation d'une fonction en boîte noire de MAGMA : Variety

[3] Ensemble de racines communes d'un nombre fini de polynômes en plusieurs variabes

# Tests sur l'attaque

Résolution impossible avec une seule keyStream récupérée (version étudiante de MAGMA).
→ Introduction de 2 paramètres: nbo et nb

- nbo: taille de la concaténation des keyStreams considérées
- nb: nombre de variables des registres qu'on fixe

| Nbo | Nb | Time(s) | Mem. Size(Mo) |
|---|---|---|---|
| 15 | 6 | 4109,70 | - |
| 15 | 8 | 264,9 | 2357 |
| 15 | 10 | 114,64 | 1843 |
| 15 | 20 | 1,92 | 33 |
| 30 | 1 | 11,58 | 224 |
| 30 | 3 | 0,34 | 52,7 |
| 30 | 5 | 0,3 | 49,3 |
| 45 | 0 | 0,280 | 62,38 |

**Entrée:** Message C chiffré récupéré sur le réseau

**Sortie:** Etat interne des registres après le KeySetup

# Fonctionnement de l'attaque

## chiffrement + correction

Correction

$$M = G.P \oplus g \tag{1}$$

- P: message initial (184 bits)
- G: matrice de correction d'erreur (184x456)
- g: vecteur de correction des erreurs de transmission (ici fixé à 0 pour simplifier)
- M: message corrigé (456 bits)

Chiffrement

$$C = M \oplus k$$

# Fonctionnement de l'attaque
## matrice de parité

G de taille 456x184

**456-272=272 équations décrivent le noyau de la transformation inverse.**

Notons $H$ la matrice de taille 272x456 représentant ces équations.

$$\rightarrow H.M = 0 \;\; \forall M \in \{0,1\}^{456}$$

$H$ : "Matrice de Parité"

# Fonctionnement de l'attaque
## Construction du système linéaire

$$H.M = 0 \tag{2}$$

$$C = M \oplus k \tag{3}$$

(2) et (3) $\rightarrow$ (4)

$$H.C = H.(M \oplus k) = H.M \oplus H.k = H.k \tag{4}$$

Système linéaire à résoudre:

$$H.C - H.k = 0 \tag{5}$$

# Fonctionnement de l'attaque

## Ajout d'équations

$\rightarrow$ 272 équations linéaires pour 456 variables

Insuffisant !

Solution: concatèner plusieurs systèmes linéaires.

$$H.C1 - H.k \;||\; H.C2 - H.k = 0 \tag{6}$$

# Fonctionnement de l'attaque

## utilisation de l'attaque n°1

- Remplacer les 456 variables de k par les polynômes générés dans l'attaque n°1
- Résoudre $H.C1 - H.k \ ||...|| \ H.C - H.k = 0$
- Résultat: état des registres avant la $2^{\text{ème}}$ A5/2

# Fonctionnement de l'attaque
## Etape Finale

**Etape finale:** Retrouver la KeyStream avec A5/2 !

# Résumé de attaque et vérification

- Génération d'un vecteur aléatoire k_alea
- Génération de messages chiffrés: $C_i = M_i \oplus k$
- Résolution du système:

$$H.C_1 - H.k \mid\mid ... \mid\mid H.C_n - H.k = 0$$

$\rightarrow$ Etat interne des registres après le KeySetup

- A5/2 + registres $\rightarrow$ KeyStream
- Vérification: $k = k\_alea$

# Complémentarité des deux attaques

La 2$^{\text{ième}}$ attaque utilise la 1$^{\text{ière}}$ attaque pour retrouver l'état des registres

La 1$^{\text{ière}}$ attaque n'utilise qu'une KeyStream en clair pour retrouver l'état internet des registres

# Conclusion

A5/2 "cassable" par des attaques assez simples à mettre en place
A éviter si on veut assurer la confidentialité des communications

## Ce que le projet nous a apporté...

- Apprentissage de Magma
- Premiers pas dans le monde de la recherche
- Mise en application directe des notions vues en cours

# Remerciements

M. Ludovic Perret