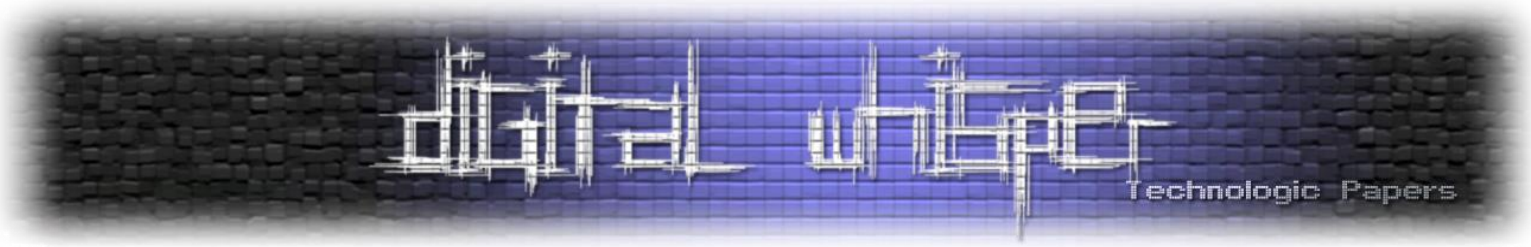


GSM SMS Decryption

מאת מיכאל טויטו

הקדמה

ברוב מכשירי הסלולר הנמצאים בשימוש קיים כרטיס חכם שנקרא כרטיס SIM, כרטיס זה הינו בקר לכל דבר ועניין ומכיל בתוכו CPU, מערכת הפעלה ואזור זיכרון, כרטיס ה-SIM הוא למעשה מודול זיהוי המנוי בין המכשיר לבין הרשת הסלולרית כך שעבור כל פעולה שדורשת אימות עם הרשת, ישנה אינטראקציה בין כרטיס ה-SIM, דרך המכשיר הסלולרי עם הרשת.

במאמר זה אגע בחלק קטן באינטראקציה הזו כדי להסביר כיצד ניתן לפענח חבילות תקשורת מוצפנת של הודעת SMS על ידי גישה לכרטיס ה-SIM, ההנחות שנתבסס עליהן הינן:

- הודעת ה-SMS מתקבלת כאשר המכשיר המקבל מחובר לרשת GSM (דור שני).
- התוקף מאזין לאנטנה בעת קבלת ה-SMS.
- האנטנה המואזנת אינה מבצעת CHANNEL HOPPING.
- לתוקף יש גישה לכרטיס ה-SIM ממש לאחר קבלת ה-SMS.

הסבר מפורט על ההנחות הללו וכיצד אפשר בעזרת מחקר נוסף להתגבר עליהן אספק בסוף המאמר.

תוכנות וציוד נדרשים

1. עבור קריאת הפרמטרים מה-SIM צריך קורא כרטיסים פשוט (של רב-קו עובד מצוין), ובנוסף תוכנה לתקשר עם הקורא, יש המון תוכנות, אני ממליץ על:

https://github.com/minghsu/usim_modifier_v3

2. כרטיס SDR כלשהו שמסוגל להאזין על תדרי GSM (לדעתי כמעט כולם מסוגלים), ואת התוכנות grgsm:

https://github.com/ptrkrysik/gr-gsm

מפתחות בכרטיס ה-SIM

כפי שציינתי בהקדמה כרטיס ה-SIM מהווה מנגנון Authentication עם הרשת, מכאן שהכרטיס מכיל בתוכו מפתחות הצפנה עבור מספר סוגי חיבורים. בנוסף לכרטיס ישנה היכולת להפעיל אלגוריתמים כדי לייצר מפתחות זמניים, כך שכל SESSION של שיחה או SMS יהיה מוצפן עם מפתח חד פעמי.

בכרטיס עצמו ישנם מספר סוגי מפתחות קבועים שמוטמעים בעת צריבת מערכת ההפעלה של הכרטיס, לדוגמא סוג אחד הינם שלושה מפתחות (**KIC**, **KID** ו-**KIK**) עבור עדכוני OTA (כתיבה וקריאה של קבצים או התקנה של אפליקציות ב-SIM ללא גישה פיזית לכרטיס).

סוג אחר הינו מפתחות **ADM** ו-**PIN** המשמשים עבור קריאה וכתיבה או הרצה של תוכניות על ה-SIM בגישה פיזית, כלומר ע"י קורא כרטיסים ומפתח אחד שעליו ארחיב מעט הינו ה-**KI** המשמש לצרכי אימות.

מהו ה-KI ולמה הוא חשוב?

ה-**KI** הינו המפתח הכי חשוב בכרטיס ה-SIM מאחר ובכל פעם שמכשיר כלשהו עולה לרשת, מתבצע Authentication challenge בין הכרטיס אל ה-HSS/HLR (השרת 'המרכזי') של מפעיל הסלולר, שזהו המקום היחידי שמחזיק את מפתח ה-**KI** עבור כל מנוי.

במאמר זה אתמקד על המפתח החד פעמי **KC** אך לפני שאסביר עליו נצטרך להבין מה קורה בעת קבלת SMS ואיך כרטיס ה-SIM קשור לכל זה.

מפתח חד פעמי KC

כפי שציינתי לכרטיס ה-SIM ישנה היכולת להפעיל אלגוריתמי הצפנה כדי לייצר מפתח חד-פעמי, מפתח זה נקרא KC, והוא מיוצר על ידי פקודת **APDU** שנשלחת אל ה-SIM כאשר הקלט הינו מפתח ה-KI וה-**RAND** שנשלח מהרשת - מספר רנדומלי בגודל 16 בתים.

פרוטוקול APDU

כפי שציינתי כרטיס ה-SIM הינו בקר לכן יש אפשרות לכתוב ולקרוא ממערכת הקבצים שלו או לגרום לו 'להריץ תכנית' על ידי פקודות, הפרוטוקול שמשתמשים בו נקרא **APDU** והוא נמצא בשימוש בהרבה כרטיסים חכמים היום, ישנו תקן מסוים עבור הפקודות הללו וכמובן גם תקן עבור מערכת הקבצים בכרטיס ה-SIM (לפעמים ישנם חריגות מועטות במבנה הפקודות או בשמות במערכת הקבצים), ישנן הרבה הסברים באינטרנט על מבנה הפקודה לכן לא אתעכב על כך במסגרת מאמר זה.

קריאה של קובץ

דוגמא לסדרת פקודות **APDU** של קריאת קובץ:

```
00A40000025F3B - (SELECT THE DIRECTORY 5F3B)
00A40000024F20 - (SELECT THE FILE 4F20)
00B0000009     - (READ 9 BYTES OF DATA)
```

הפקודה הראשונה תבחר את התיקייה **GSM-ACCESS**, הפקודה השניה תבחר את הקובץ **KC** והפקודה השלישית תקרא 9 בתים מהקובץ הנ"ל (הקובץ הינו מסוג **transparent**).

שימו לב שהפקודות יכולות להשתנות מעט בין יצרני כרטיסים שונים, בעיקר שני הבתים הראשונים משמאל, אך ניתן למצוא את הווריאציות השונות במסמכי התקן של GlobalPlatform, כמו כן ישנם קבצים ופקודות שלא ניתן לגשת אליהם או להריץ אותם ללא הזנת מפתח, את כל הפרטים הללו ניתן לראות במסמכי התקנים של **ETSI/3GPP**.

הרצת תכנית

ניקח לדוגמא פקודה להרצת תוכנית המקבלת מספר רנדומלי באורך 16 בתים ומחשבת עליהם את אלגוריתם ההצפנה **A3** ואת **A8** ויחזיר SRES (4 Bytes) ו-KC (8 Bytes) בהתאמה:

```
RUN GSM ALGORITHM
88000010112233445566778899101112131415
```

כאשר **11223344556677889910111213141 5** הינו מספר רנדומלי כלשהו, נקרא לו **RAND**, שאפרט עליו בחלק הבא.

תהליך התחברות לרשת GSM

כידוע ישנם הרבה אנטנות סלולר של מפעילות סלולר שונות, איך המכשיר יודע באיזה רשת לבחור?

בחירת אנטנה

בכרטיס ה-SIM ישנו קובץ **(EFplmn)** וכאשר מצב 'חיבור אוטומטי לרשת' פעיל, המכשיר הסלולרי יקרא את הקובץ ויבצע חיפוש, ולאחר מכן מתחיל ניסיונות חיבור לאנטנות השייכות למפעיל עם התעדוף הגבוה ביותר בקובץ, כעת נסביר כיצד מתבצע 'ניסיון חיבור' שכזה.

הליך האימות

כרטיס ה-SIM הוא בעל מספר ייחודי שמזהה אותו בכל רשת ה-GSM העולמית שנקרא IMSI, כדי לשמור על סודיות המספר הזה, בתחילת ה-SESSION עם הרשת כרטיס ה-SIM יקבל מהרשת מספר זמני שנקרא TMSI, מספר זה יחליף את תפקיד ה-IMSI במהלך ה-SESSION הנוכחי.

לאחר מכן, הכרטיס יקבל מהרשת מספר רנדומלי באורך 16 בתים - **RAND**, וכאן יקרו שתי פעולות, ע"י כך שהמכשיר יבצע את הרצת הפקודה שראינו קודם לכן: **RUN GSM ALGORITHM**, אשר מחזירה שני פרמטרים בהינתן ה-**KI** והמספר הרנדומלי, הכרטיס יחשב את מחרוזת האימות שתאומת ע"י הרשת - SRES, ובנוסף מפתח ההצפנה הזמני שנקרא KC ייכתב בקובץ שנמצא ב-SIM (הסיבה לכך היא של-SIM אין את היכולת החישובית להצפין DATA ב-REAL-TIME), מפתח זה ישמש את המכשיר והרשת עבור ההצפנה של ה-SESSION.

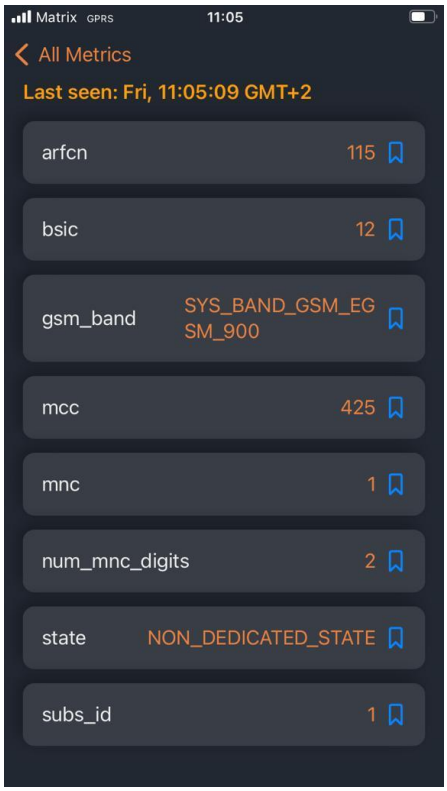
האזנה לאנטנת GSM

לכל אנטנת GSM משוייך טווח או מספר טווחים של תדרים עבור העלאה מידע ועבור הורדתו, ולכל מכשיר שמחובר לאנטנה ישנו תדר מסוים שממנו הוא יוריד מידע (DOWNLINK) ותדר אחר שאליו הוא יעלה (UPLINK) מידע. נקודה חשובה כאן היא שישנן אנטנות אשר משתמשות ב-CHANNEL HOPPING כלומר כל מכשיר יצטרך לקפוץ בין תדרים בזמן נתון, במצב כזה צריך להאזין על טווח תדרים ולדעת את סדר הקפיצה כדי לפענח את המידע שנקבל, אתן על זה מעט מידע בהמשך אך במאמר זה נניח כי האנטנה לא מבצעת CHANNEL HOPPING.

כדי להאזין לתדר שממנו המכשיר מוריד מידע מהרשת, נצטרך למצוא את האנטנה שעליה מחובר המכשיר בעת קבלת ה-SMS ולמצוא את תדר ההורדה של אנטנה זו, באופן כללי תיעדוף המכשיר יהיה לאנטנה עם הקליטה הכי טובה המשויכת לחברה המפעילה (של ה-SIM) לכן ניתן לבצע סריקה עם grgsm_scanner

ולמצוא את אנטנות ה-GSM עם הקליטה החזקה ביותר, אך לשם הנוחות נוציא את התדר מהמכשיר ע"י קוד טכנאי.

בהדגמה אשתמש באייפון אך ניתן למצוא קוד טכנאי עבור כל מכשיר על ידי חיפוש פשוט באינטרנט, אז ראשית, נעבור לרשת GSM (דור שני), ואז נבצע קוד טכנאי ***#3001#12345*** ותחת: **SERVING CELL INFO** נראה מהו ה-ARFCN:



ARFCN הינו מספר סידורי המשויך לתדר ונמיר אותו לתדר בקישור הבא:

https://www.cellmapper.net/arfcn

תוכניות ה-grgsm שולחות אל כתובת ה-LOOPBACK את החבילות שהן מקבלות ולכן נפתח Wireshark ונאזין על ה-LOOPBACK כשנרצה לראות את החבילות. כדי לבדוק האם האנטנה המואזנת מבצעת CHANNEL HOPPING כאשר 958.0 הינו התדר נריץ:

```
michael@Matrix:~$ grgsm_livemon -f 958.0M
```

נביט על חבילה מסוג **System information Type 1**:

```
GSM CCCH - System Information Type 1
  L2 Pseudo Length
  .... 0110 = Protocol discriminator: Radio Resources Management messages (0x6)
  Message Type: System Information Type 1
  Cell Channel Description
    00.. 000. = Format Identifier: bit map 0 (0x00)
    List of ARFCNs = 115
  RACH Control Parameters
  SI 1 Rest Octets
```

בשדה **List of ARFCNs** מופיע ARFCN אחד, יתרה מזאת נביט על חבילה מסוג **Immediate Assignment**:

1649	226.553988654	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1650	226.619297401	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1651	226.628531219	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1652	226.641169584	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1653	226.684563952	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1654	226.697587950	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1655	226.705726325	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1656	226.754214838	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1657	226.762960294	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1658	226.812158515	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) System Information Type 3
1659	226.821286415	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1660	226.834596687	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1661	226.878173632	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1662	226.891828817	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1663	226.899953623	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1664	226.948665182	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Immediate Assignment
1665	226.956202462	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Immediate Assignment
1666	226.971549978	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1667	227.012595357	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) Paging Request Type 1
1668	227.028070281	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) System Information Type 2
1669	227.037352975	127.0.0.1	127.0.0.1	GSMTAP	81 (CCCH) (RR) System Information Type 2quater

```
▾ GSM CCCH - Immediate Assignment
  ▸ L2 Pseudo Length
  ▸ .... 0110 = Protocol discriminator: Radio Resources Management messages (0x6)
    Message Type: Immediate Assignment
  ▸ Page Mode
  ▸ Dedicated mode or TBF
  ▾ Packet Channel Description
      0000 1... = Channel Type: 1
      .... .110 = Timeslot: 6
      100. .... = Training Sequence: 4
      .... .0.. = Spare: 0x00
      .... ..00  0111 0011 = Single channel ARFCN: 115
```

מכאן נסיק שאכן ישנו שימוש ב-ARFCN יחיד. דוגמא לאנטנה שמבצעת CHANNEL HOPPING חבילת **Immediate Assignment**:

```
▾ GSM TAP Header, ARFCN: 123 (Downlink), TS: 0, Channel: CCCH (2)
    Version: 2
    Header Length: 16 bytes
    Payload Type: GSM Um (MS<->BTS) (1)
    Time Slot: 0
    ..00 0000 0111 1011 = ARFCN: 123
    .0.. .... .... .... = Uplink: 0
    Signal Level (dBm): -53
    Signal/Noise Ratio (dB): 0
    GSM Frame Number: 704122
    Channel Type: CCCH (2)
    Antenna Number: 30
    Sub-Slot: 2
▾ GSM CCCH - Immediate Assignment
  ▸ L2 Pseudo Length
  ▸ .... 0110 = Protocol discriminator: Radio Resources Management messages (0x6)
    Message Type: Immediate Assignment
  ▸ Page Mode
  ▸ Dedicated mode or TBF
  ▾ Channel Description
      0100 0... = SDCCH/8 + SACCH/C8 or CBCH (SDCCH/8): 8
      Subchannel: 0
      .... .001 = Timeslot: 1
      100. .... = Training Sequence: 4
      ...1 .... = Hopping Channel: Yes
      Hopping channel MAIO: 0
      HSN: 22
```

וחבילת **System Information Type 1**:

```
GSM CCCH - System Information Type 1
  L2 Pseudo Length
  .... 0110 = Protocol discriminator: Radio Resources Management messages (0x6)
  Message Type: System Information Type 1
  Cell Channel Description
     00.. 010. = Format Identifier: bit map 0 (0x02)
     List of ARFCNs = 123 119
```

ניתן לראות שישנה קפיצה בין שני תדרים, שאר הפרמטרים ב-CHANNEL DESCRIPTION יכולים לסייע בפענוח חבילות שנשלחות באמצעות CHANNEL HOPPING אך לא אכנס לזה במסגרת מאמר זה.

לכידת המידע

לאחר שיש בידנו את האנטנה המבוקשת, נאזין באמצעות ה-SDR על התדר ונלכוד את החבילות בעזרת הכלי grgsm_capture:

```
michael@Matrix:~$ grgsm_capture -f 958.0M call.cfile
```

שימו לב שייתכן ותאלצו לשנות מעט את פרמטרים בפקודה כגון sample_rate, לכן רצוי לקרוא מעט על הפקודה הזו כדי לקבל את המידע בשלמותו.

כעת כשיש בידינו את החבילות בתוך call.cfile נרצה לפענח אותם, לשם כך נחלץ את המפתח וה-TMSI מה-SIM ע"י APDU.

שליפת המידע מה-SIM

ראשית נשלוף את ה-TMSI מה-SIM כדי שנוכל להפעיל FILTER על החבילות הרלוונטיות:

```
PIN1 Enabled: False, PIN1 Verified: False, ADM Key Verified: False

Type 'exit' to exit, 'plugin' for summary of suppotred plugins.

USIM modifier$ send 00A40000026F7E

, 61 21

USIM modifier$ send 00b0000009

8A 58 72 2E 24 F5 10 28 E7, 90 00

USIM modifier$
```

כעת נחלץ את ה-KC שהיה בשימוש בעת קבלת ה-SMS, ע"י הפקודות שראינו קודם:

```
USIM modifier$ send 00A40000025F3B

  , 61 2B

USIM modifier$ send 00A40000024F20

  , 61 21

USIM modifier$ send 00b0000009

  6F 8B C4 06 7E 18 D7 3A 06, 90 00
```

פענוח המידע

כאשר מקבלים SMS מתבצעות מספר פעולות בין הרשת למכשיר:

1. האנטנה **(BTS)** שולחת **Paging Request** למכשיר
2. המכשיר שולח **Channel Request** אל ה-**BTS**
3. ה-**BTS** שולח **Immediate Assignment** אל המכשיר
4. התקשורת ממשיכה ב-**Dedicated Channel**

מבלי להיכנס ליותר מידי פרטים ה-**SDCCH** (קיצור של **Stand-alone Dedicated Control Channel**) הינו CHANNEL המשמש עבור 'החלפת סיגנלים' בין המכשיר אל הרשת, והוא בעל 8 TIMESLOTS, כלומר בזמן נתון המכשיר יקבל שידור על גבי **X** TIMESLOT (כמובן X מודולו 8).

ניתן למצוא את ה-TIMESLOT בחבילת Immediate Assignment אם נריץ:

```
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m BCCH
```

עם **FILTER** ב-Wireshark נוכל לראות חבילות רלוונטיות:

gsm_a.rr.dedicated_mode_or_tbf == 0

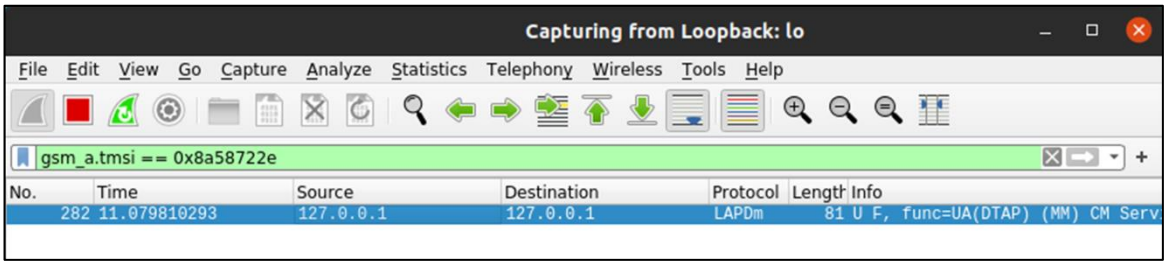
No.	Time	Source	Destination	Protocol	Length	Info
246	0.756884531	127.0.0.1	127.0.0.1	GSMTAP	81	(CCCH) (RR) Immediate Assignment
380	1.516159243	127.0.0.1	127.0.0.1	GSMTAP	81	(CCCH) (RR) Immediate Assignment

```
▾ Channel Description
    0110 0... = SDCCH/8 + SACCH/C8 or CBCH (SDCCH/8): 12
    Subchannel: 4
    .... .010 = Timeslot: 2
    100. .... = Training Sequence: 4
    ...0 .... = Hopping Channel: No
    ..00 .... = Spare: 0x00
    Single channel ARFCN: 115
```

הבעיה היא שכאן לא מצאתי אפשרות לבצע Filter לפי TMSI, במקרה הזה בשתי החבילות ה-TIMESLOT הינו 2, אבל כשלוכדים הרבה מידע יתכן שנקבל מספר TIMESLOTS, מאחר ואנחנו לא יודעים מראש את ה-TIMESLOT, נוכל לנסות את כולם וכאשר נקבל את ההודעה הרצויה באמצעות ה-FILTER שנשים על ה-TMSI הרצוי - נדע שזהו ה-TIMESLOT הנכון:

```
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 0
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 1
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 2
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 3
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 4
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 5
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 6
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 7
```

וכעת נפעיל את ה-FILTER על ה-TMSI שלנו:



אצלי ה-FRAME הוא 2, לכן כשביצעתי את הפקודה עם:

```
-t 2
```

קיבלתי את החבילה המעניינת, עכשיו נותר להבין מהו אלגוריתם ההצפנה ולפענח באמצעות ה-KC.

גילוי אלגוריתם ההצפנה

ישנם מספר אלגוריתמים (או אלגוריתם אחד עם בוריאציות שונות) המשומשים ברשת GSM, כל מכשיר משתמש באחד מהם (אני משתמש ב-iPhone SE במהלך ההדגמה), העיקריים הם A5/1 ו-A5/3 (כאשר הראשון ככל הנראה ניתן לפריצה באמצעות RAINBOW TABLES), אנחנו לא נצטרך לפרוץ אותם מאחר ויש לנו את המפתח KC.

לכן מה שנעשה הוא לנסות להריץ את grgsm_decode עם כל אחד מהאלגוריתמים הללו ונראה האם המידע פוענח:

```
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 2 -e 3 -k 6F8BC4067E18D73A
```

נביט כעת ב-Wireshark:

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
2	0.121888603	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
3	0.147292496	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
4	0.172707542	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
5	0.197594815	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
6	0.323744673	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
7	0.348091586	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
8	0.374311839	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
9	0.400383957	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
10	0.531253613	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
11	0.557593989	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
12	0.583283922	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
13	0.607818243	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
14	0.728924379	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
15	0.779466391	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
16	0.783769099	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=1, N(S)=0(DTAP) (RR) Ciphering Mode Command
17	0.805241929	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
18	0.934797613	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
19	0.985199943	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
20	1.010108083	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
21	1.141534381	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
22	1.166862104	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
23	1.194215261	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
24	1.221011845	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
25	1.348838019	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
26	1.373492346	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
27	1.398536351	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
28	1.537843281	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 6
29	1.549139384	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
30	1.555037435	127.0.0.1	127.0.0.1	LAPDm	81	U F, func=UA(DTAP) (MM) CM Service Request
31	1.575642023	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
32	1.582723067	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=1, N(S)=0 (Fragment)
33	1.591108101	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 5
34	1.602634213	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
35	1.608665631	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=2, N(S)=1(DTAP) (MM) Authentication Request
36	1.627841066	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
37	1.633801722	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI
38	1.641787982	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 5
39	1.660047238	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=3, N(S)=2(DTAP) (RR) Ciphering Mode Command
40	1.780946590	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
41	1.806480249	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
42	1.832087388	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
43	1.957440979	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
44	2.008881660	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
45	2.035588631	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
46	2.160963397	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
47	2.185857742	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
48	2.211920600	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
49	2.238007785	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
50	2.365272826	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
51	2.392046945	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
52	2.444748456	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)

הניסיון הראשון לא צלח, תוכלו לראות שלאחר Ciphering mode command אין לנו מידע על החבילה כלומר unknown func. ניסיון שני (ואחרון):

```
michael@Matrix:~/rf$ grgsm_decode -a 115 -c call.cfile -m SDCCH8 -t 2 -e 1 -k 6F8BC4067E18D73A
```

41	1.674969311	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=3, N(S)=2(DTAP) (RR) Ciphering Mode Command
42	1.700798706	127.0.0.1	127.0.0.1	LAPDm	81	S, func=RR, N(R)=4
43	1.725819744	127.0.0.1	127.0.0.1	LAPDm	81	U F, func=UA
44	1.749926397	127.0.0.1	127.0.0.1	LAPDm	81	S, func=RR, N(R)=1
45	1.757508394	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 5
46	1.792995786	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
47	1.806681651	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 5
48	1.818674799	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
49	1.825093506	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=3, N(S)=1(DTAP) (SMS) CP-DATA (RP) RP-ACK (Network to
50	1.845801199	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown(DTAP) (SS)
51	1.851960733	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=4, N(S)=2 (Fragment)
52	1.859407281	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 6
53	1.876076448	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=4, N(S)=3 (Fragment)
54	1.900190140	127.0.0.1	127.0.0.1	GSM SMS	81	I, N(R)=4, N(S)=4(DTAP) (SMS) CP-DATA (RP) RP-DATA (Network to
55	1.907728680	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(CCCH) (RR) System Information Type 5
56	1.924225077	127.0.0.1	127.0.0.1	LAPDm	81	S, func=RR, N(R)=5
57	1.948830494	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=4, N(S)=3(DTAP) (RR) Channel Release

כעת הדברים נראים יותר מעניינים, נוכל לראות חבילה שהפרוטוקול שלה הינו **GSM_SMS**, נפתח אותה ונקבל:

```
Wireshark · Packet 52 · Loopback: lo
  ▾ Protocol Discriminator: SMS messages (9)
      .... 1001 = Protocol discriminator: SMS messages (0x9)
      0... .... = TI flag: allocated by sender
      .000 .... = TIO: 0
    DTAP Short Message Service Message Type: CP-DATA (0x01)
  ▾ CP-User Data
      Length: 45
      RPDU (not displayed)
▾ GSM A-I/F RP - RP-DATA (Network to MS)
    Message Type RP-DATA (Network to MS)
  ▾ RP-Message Reference
      RP-Message Reference: 0x01 (1)
  ▸ RP-Originator Address - (            )
  ▸ RP-Destination Address
  ▾ RP-User Data
      Length: 33
      TPDU (not displayed)
▾ GSM SMS TPDU (GSM 03.40) SMS-DELIVER
    0... .... = TP-RP: TP Reply Path parameter is not set in this SMS SUBMIT/DELIVER
    .0.. .... = TP-UDHI: The TP UD field contains only the short message
    ..0. .... = TP-SRI: A status report shall not be returned to the SME
    .... 0... = TP-LP: The message has not been forwarded and is not a spawned message
    .... .1.. = TP-MMS: No more messages are waiting for the MS in this SC
    .... ..00 = TP-MTI: SMS-DELIVER (0)
  ▸ TP-Originating-Address - (            )
  ▸ TP-PID: 0
  ▸ TP-DCS: 0
  ▸ TP-Service-Centre-Time-Stamp
    TP-User-Data-Length: (16) depends on Data-Coding-Scheme
  ▾ TP-User-Data
      SMS text: Test no hopping

0000  09 01 2d 01 01 07 91 79  52 55 00 10 05 00 21 04   ··-····y RU····!·
0010  0c 91 79 52 90 22 32 24  00 00 12 21 80 91 91 65   ··yR·"2$ ···!···e
0020  80 10 d4 f2 9c 0e 72 bf  41 e8 37 1c 9e 76 9f 41   ····· r· A·7··v·A
```

נקודות למחקר מתקדם

אתייחס כאן לנקודות ההנחה והקשיים הגלומים בהם:

1. **הודעת ה-SMS מתקבלת כאשר המכשיר המקבל מחובר לרשת GSM (דור שני):**
   א. נדרש ציוד מתקדם כדי להאזין לתדרים בדור גבוה יותר.
   ב. אין DECODERS מוכרים עבור החבילות הללו ב-grgsm, נדרש לכתוב DECODERS בהתבסס על הפרוטוקולים הנמצאים בשימוש בדורות אלו, ולהבין מהם האתגרים החדשים ואיך להתגבר.

2. **התוקף מאזין לאנטנה בעת קבלת ה-SMS:**
   בלתי נמנע

3. **האנטנה אינה מבצעת CHANNEL HOPPING:**
   ישנן מספר אפשרויות להתגבר על כך חלקן באמצעות SDR והאחרות באמצעות GNU_RADIO, ישנם מספר מאמרים ברשת על כך.

4. **לתוקף יש גישה לכרטיס ה-SIM ממש לאחר קבלת ה-SMS:**
   כפי שציינתי ישנה אפשרות 'לשבור' חלק מהאלגוריתמים ללא המפתח.

לסיכום

עולם הטלקום בהחלט מסתורי וסובב אותנו ביום יום, מאמר זה היה נגיעה קטנה במורכבות של העולם הזה והבעיות הטמונות בו, בנוסף למורכבות הזו עולם הכרטיסים החכמים בכלל והסימים בפרט גם מאד מיושן וסגור, המאמר אכן מתייחס לרשת 2G אבל עם ציוד לא כל כך יקר ניתן לחסום תדרים של דורות גבוהים יותר ולבצע Downgrade Attack.

דרכי התגוננות:

ראשית, כיום ניתן לעבור לכרטיסי **eSIM** צ'יפ שמוטמע על לוח המכשיר ואין צורך בכרטיס SIM פלסטיק אלא רק לסרוק QR CODE - כך ניתן להתגבר על כל בעיות הנובעות מחילוץ מידע בצורה פיזית מה-SIM, שנית, ודאו שאתם על רשת מדור גבוה בעת קבלת מידע רגיש, והכי חשוב היזהרו מ-SMS ותמיד העדיפו להשתמש באפליקציות שמשתמשות בהצפנות קצה לקצה.

קצת על עצמי

מיכאל טויטו בן 29, מפתח בחברת **Annatel**, בוגר מדמ"ח ומתמטיקה באוניברסיטת אריאל, Tech Geek, מומחה לכרטיסים חכמים, RPI, SDR, ומנגן בגיטרה ופסנתר.

תודות

תודה ללירון שמעוני על ההזדמנות להיכנס לעולם הטלקום בכלל והסימים בפרט, לד"ר עמית דביר וד"ר אייל ברלינר מאוניברסיטת אריאל על הצבת האתגר והגהות על המאמר.

לקריאה נוספת

- מאמר על פיצוח 5/1A
- שני מאמרים על פענוח SMS
- סדרת סרטונים על Sniffing GSM