Hacking
Android network

singi
sjh21a@gmail.com
FB : @sjh21a

오늘의 목차는?

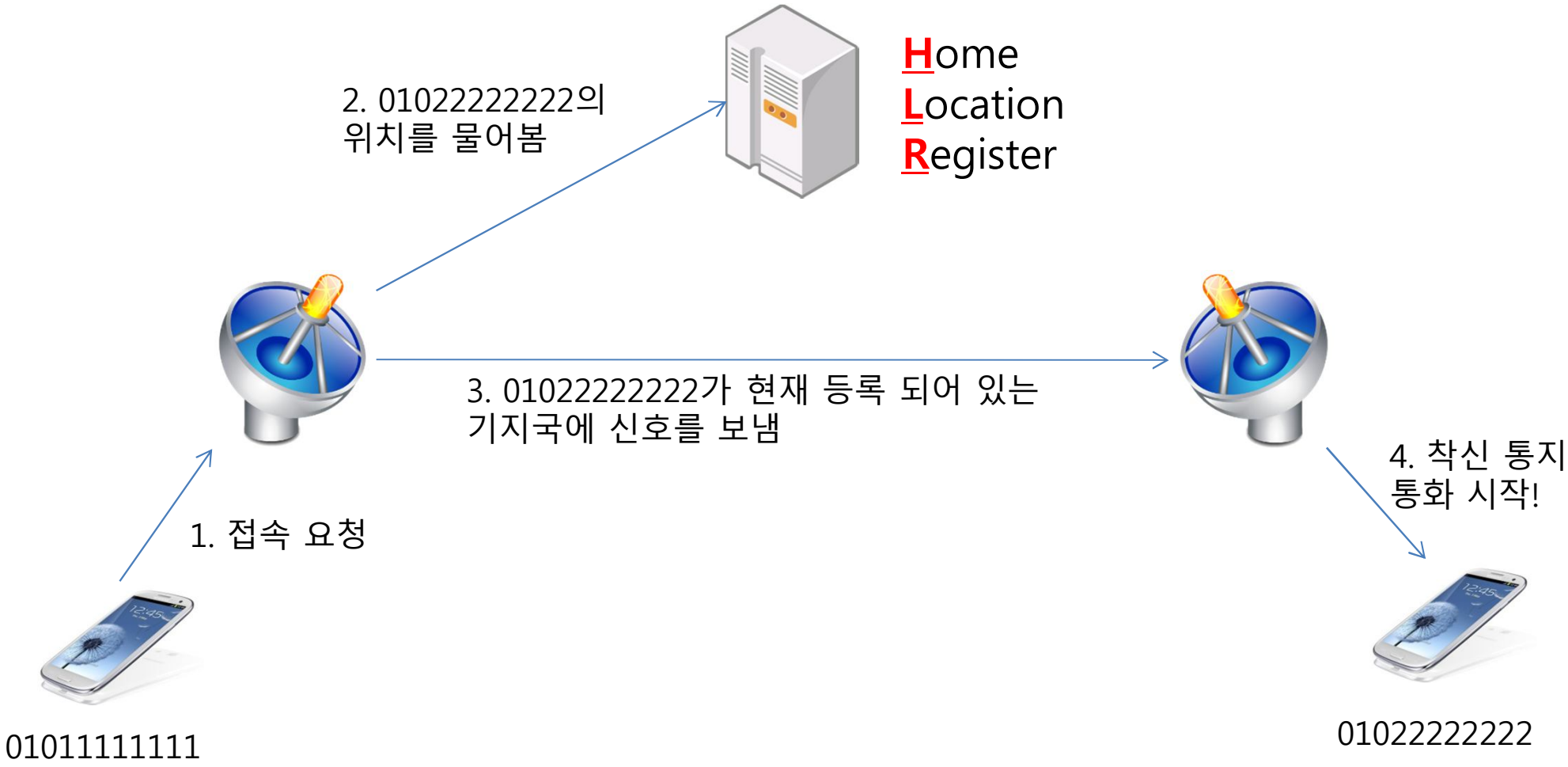
- 기본 3G/4G 통신망 구조 파악 (**쉬움!**)
- Android 통신구조 (**쉬움!**)
  - Android RIL
  - Vendor-RIL firmware 분석 해 보기
- AT command 종류 및 사용법 (**쉬움!**)
  - AT command로 SMS / 전화 해 보기
- RILD 후킹 및 제어 (**ㅅㅂ**)
  - 전화기에 fuzzing을 해보자!!

3G? 4G?

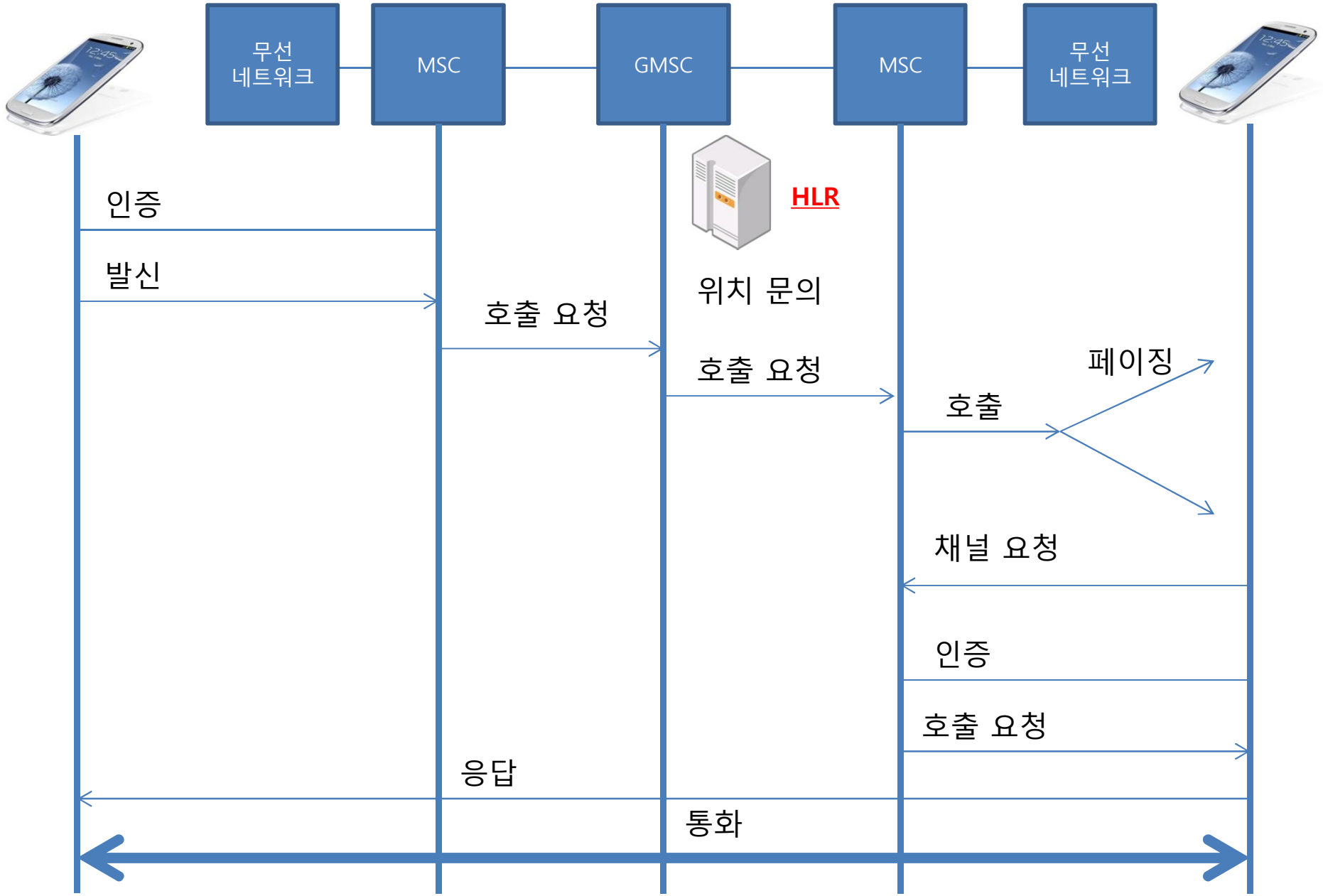
- 3G?
  - IMT-2000
  - 주파수 대역 : 2GHz (2000kHz)
  - WCDMA(2~2.4Mbps) -> HSPDA(14.4 Mbps)

- 4G?
  - IMT Beyond 또는 IMT-Advanced
  - All IP Based
  - LTE (눝 아님)

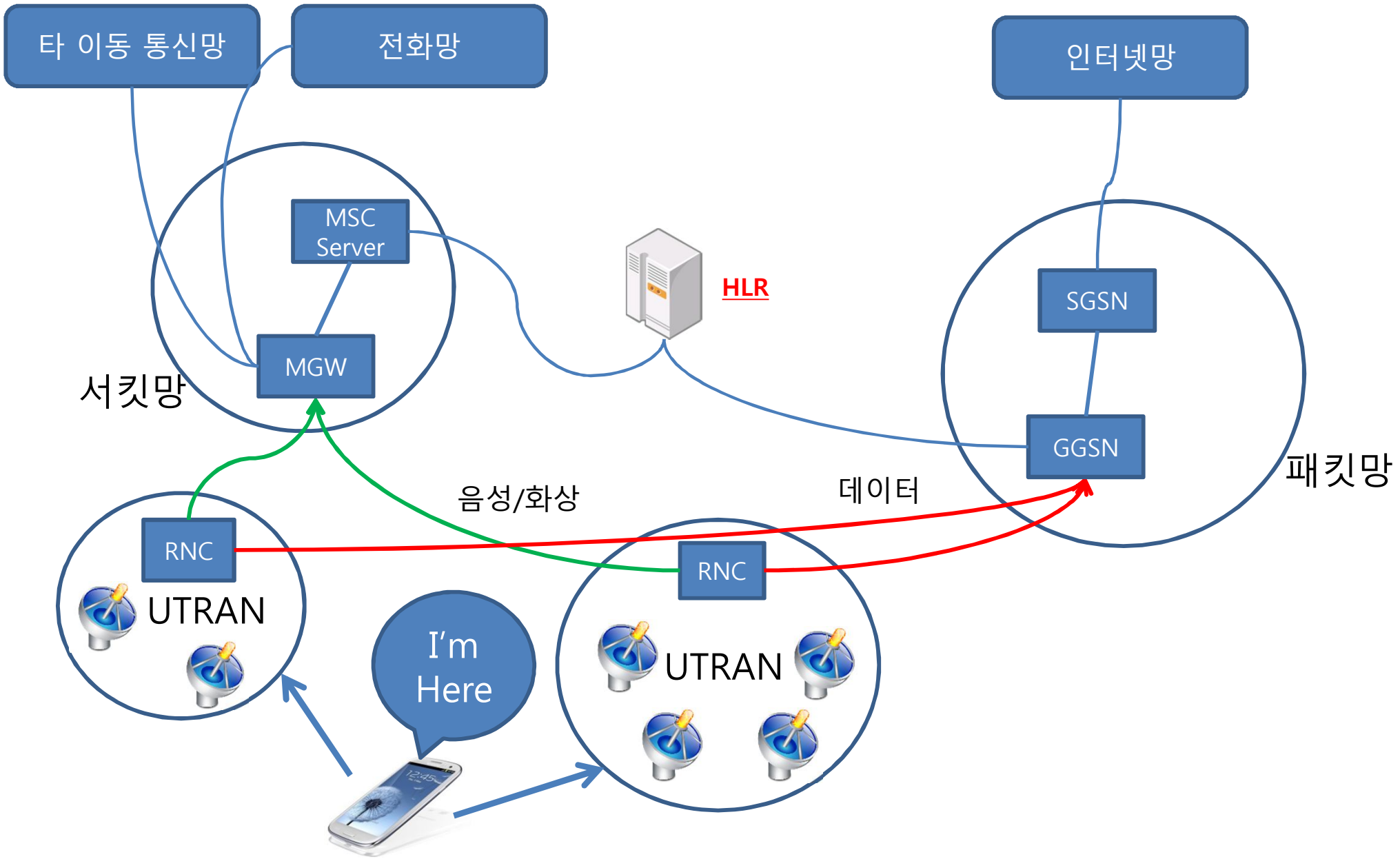
무선 전화는 어떻게 걸리고 받을까?



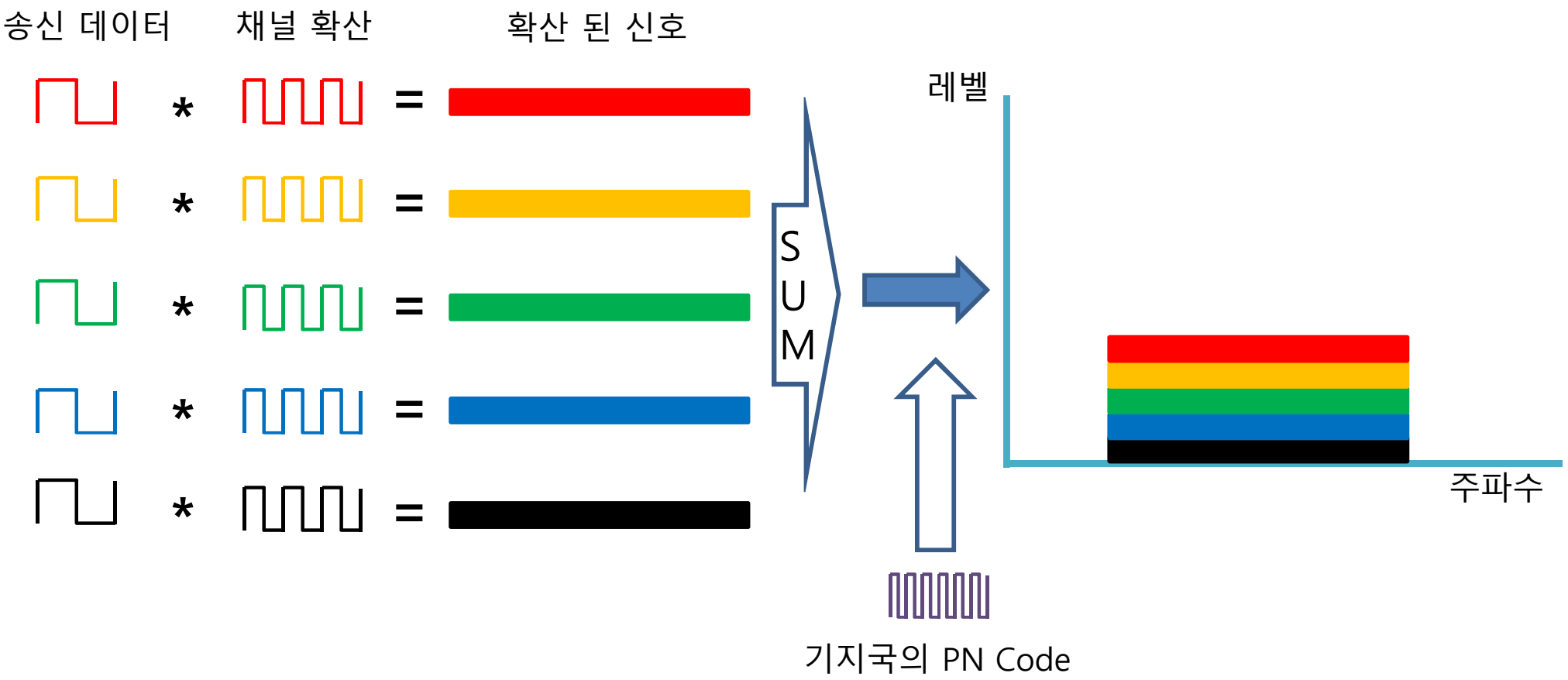
더 자세하게!


무선 네트워크
MSC
GMSC
MSC
무선 네트워크
HLR
인증
발신
위치 문의
호출 요청
호출 요청
페이징
호출
채널 요청
인증
호출 요청
응답
통화

3G Network

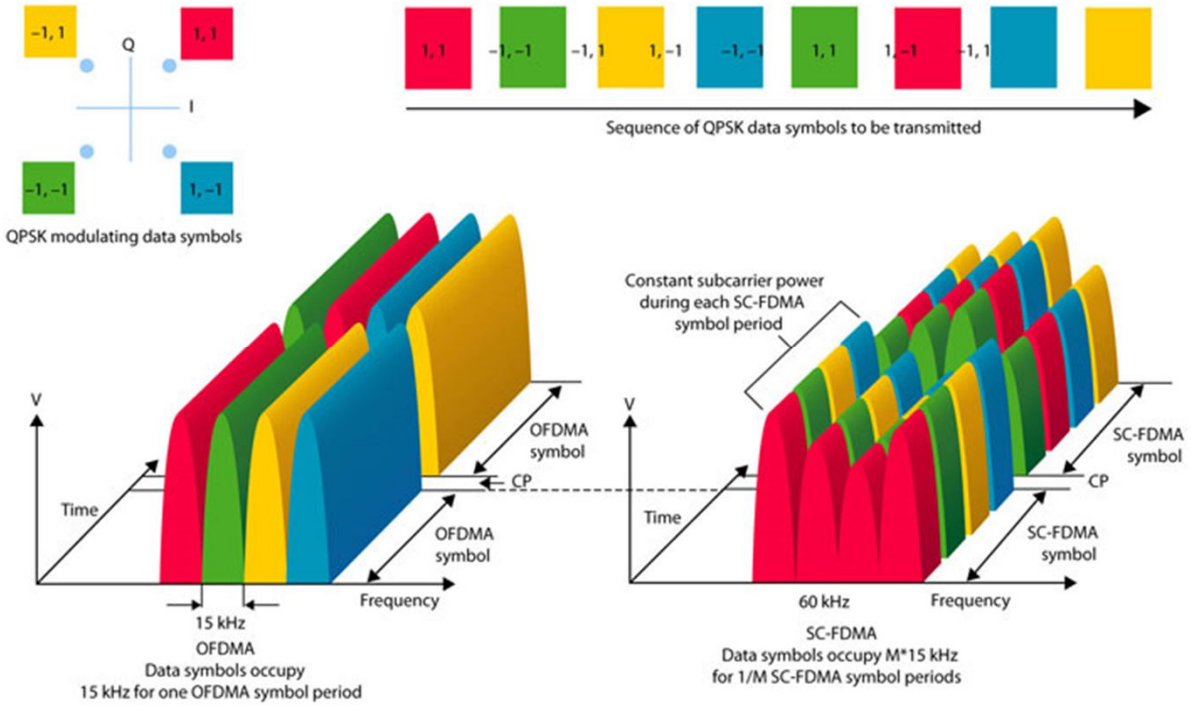

타 이동 통신망
전화망
인터넷망
MSC Server
MGW
서킷망
HLR
SGSN
GGSN
패킷망
음성/화상
데이터
RNC
UTRAN
RNC
UTRAN
I'm Here

3G 통신 구조(WCDMA)



4G 통신 구조(LTE)

- OFDMA(Orthogonal Frequency Division Multiple Access)



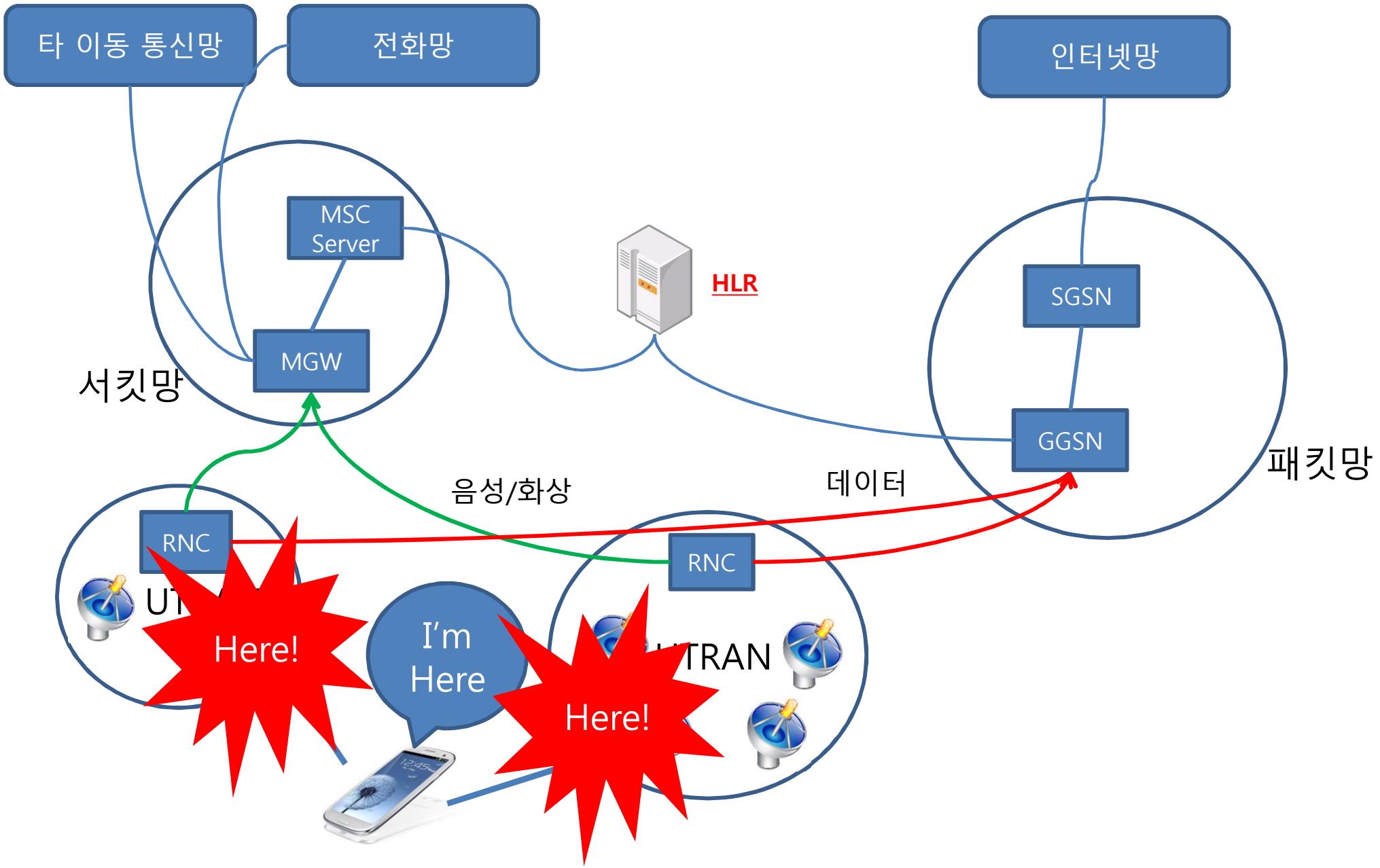
아...이제 그만...
정확히 뭘 해보려고???

- 대상은 Galaxy S3 / Android
  - Android의 통신 구조
  - RIL, Vendor-RIL
  - Vendor-RIL 추출 / 리버싱

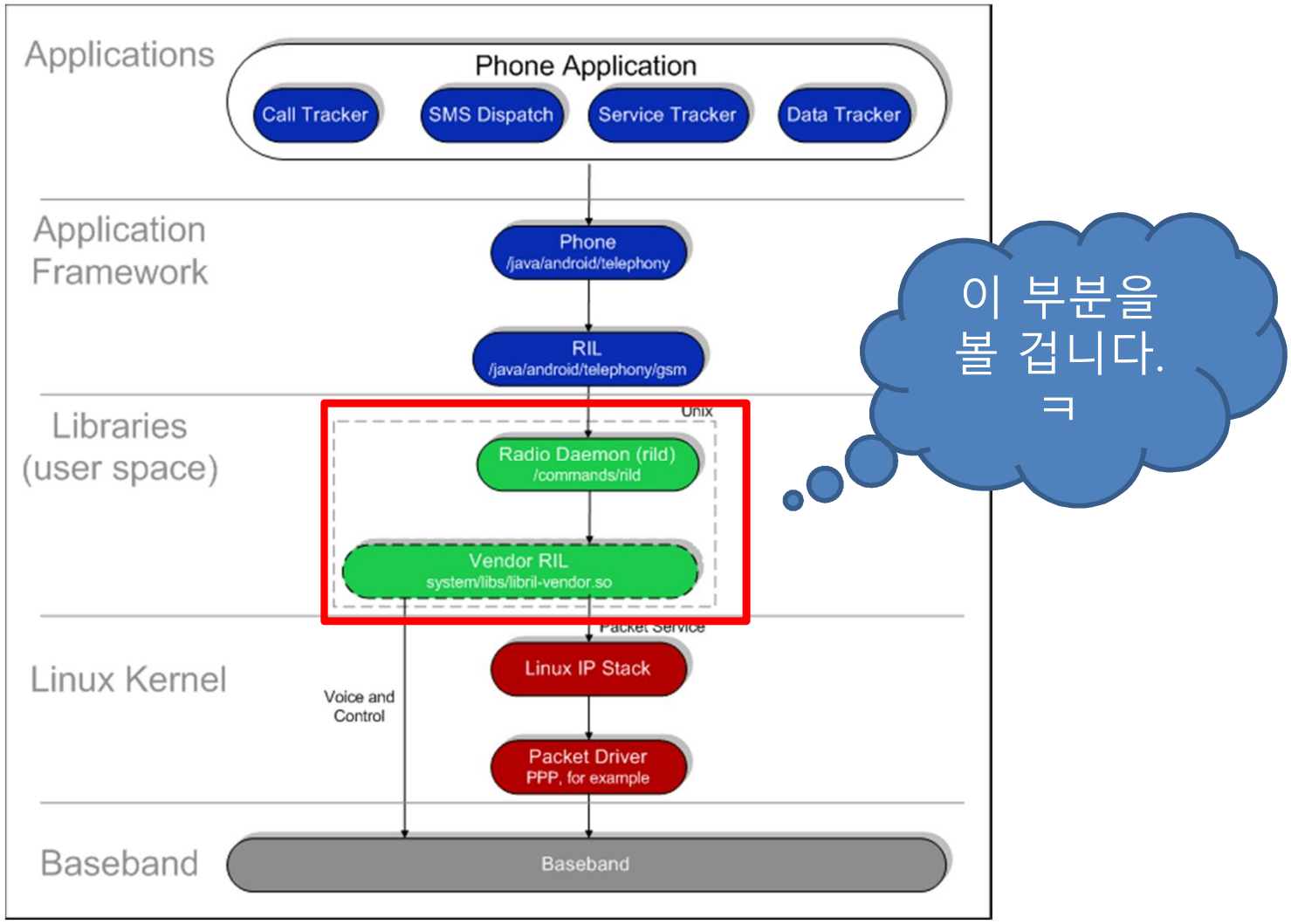
- AT command
  - 단말기에 AT command를 보내기

- RIL 후킹
  - 문자/전화/AT command를 사용 할 때 RIL 구조 파악!

이 부분을 자세히 볼 거에요!



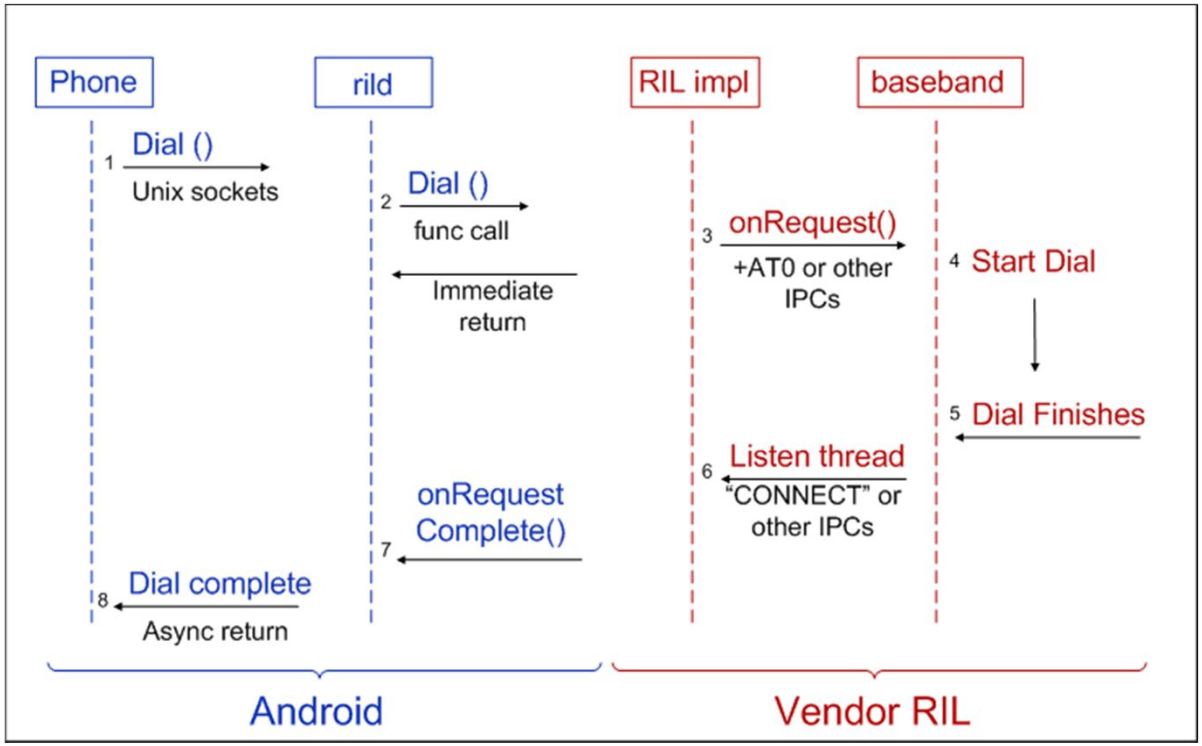
Android 통신 구조를 봅시다.



RIL? Vendor RIL?

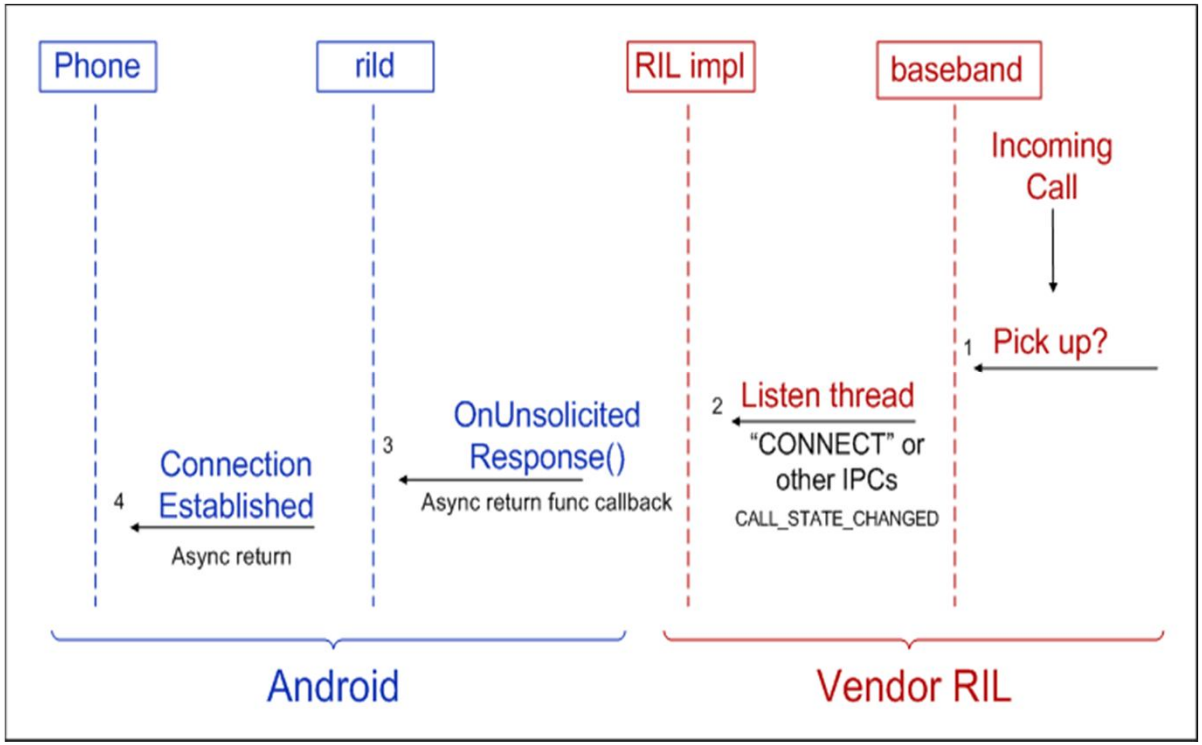
- RIL
  - Android.telephony 와 radio h/w 간의 layer
- RILD(daemon)
  - Android.telephony의 요청을 처리함.
  - Solicited 명령을 통해 Vendor RIL로 넘김.
  - Solicited?
- Vendor RIL
  - Radio h/w의 모든 통신 담당.
  - Unsolicited 명령을 통해 RILD로 넘김.
  - Unsolicited??

Solicited 명령?



- Solicited
  - Android 에서 modem쪽으로 통신 할 때 사용됨.
    - SMS 송신, Network, etc…

Unsolicited 명령?



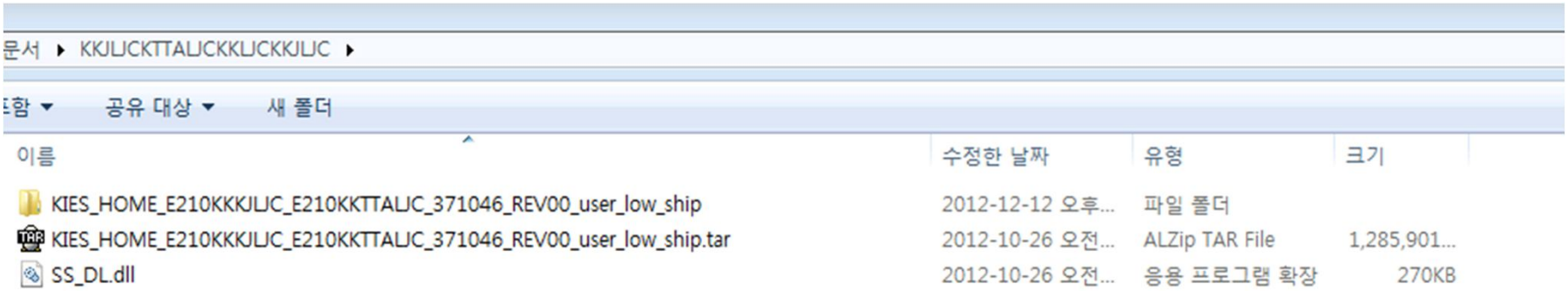
- Unsolicited?
  - Modem쪽에서 Android로 통신 할 때 사용됨.
    - 전화 받기, SMS 수신, etc…

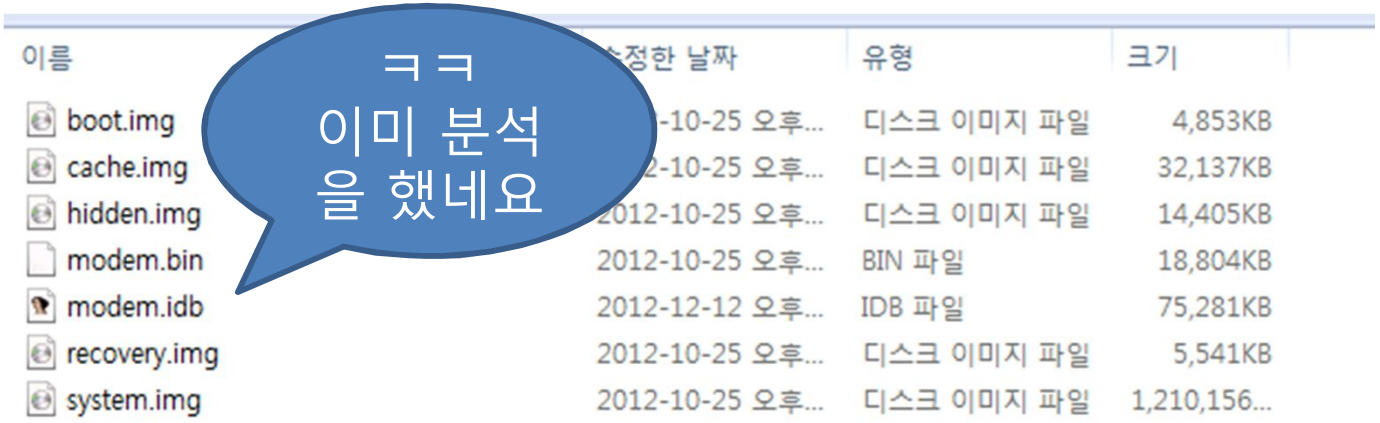
RIL 분석은 어떻게 할까??

- RIL는 OpenSource 다!
  - 그냥 볼 수 있음.
  - Java와 C로 이루어져 있음.

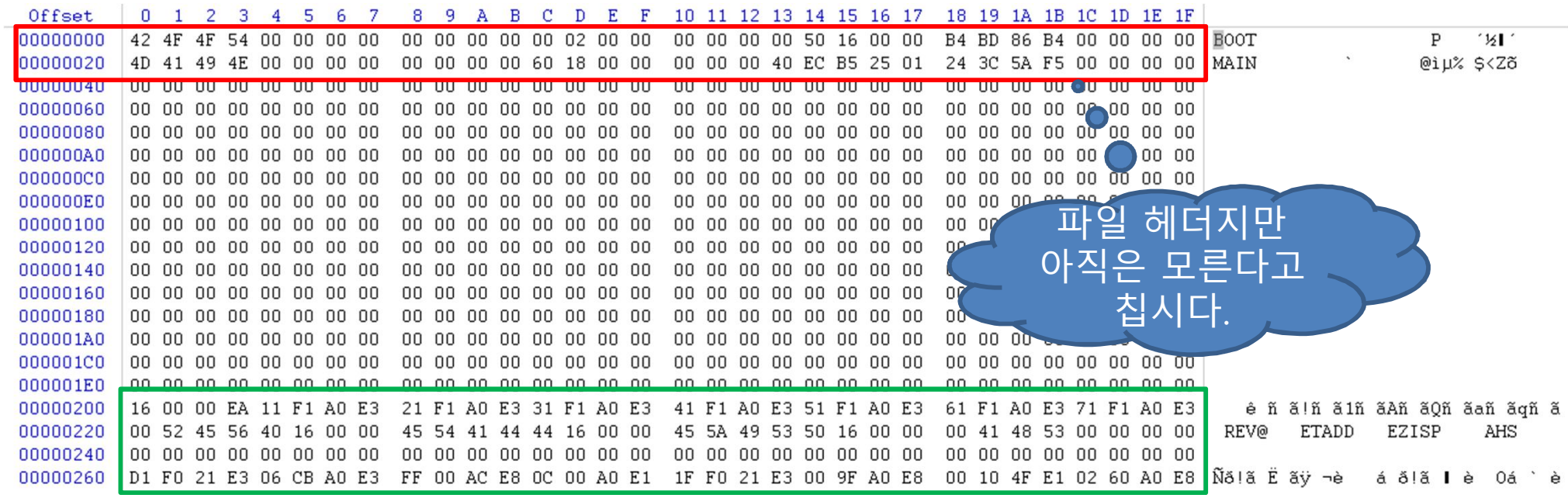
- 하지만, Vendor-RIL은 얘기가 다르다.
  - 각 제조사 마다 자신들만의 modem 코드들을 가지고 있음.
  - 분석 해 볼 것은 Samsung Galaxy S3의 Modem 코드.
  - 어떻게 분석을 시작해 볼까?

업데이트 시, firmware 추출!





Modem.bin 분석!


Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
00000000 42 4F 4F 54 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 50 16 00 00 B4 BD 86 B4 00 00 00 00 BOOT P ´½†´
00000020 4D 41 49 4E 00 00 00 00 00 00 00 00 60 18 00 00 00 00 00 40 EC B5 25 01 24 3C 5A F5 00 00 00 00 MAIN ` @ìµ% $<Zõ
00000060
00000080
000000A0
000000C0
000000E0
00000100
00000120
00000140
00000160
00000180
000001A0
000001C0
000001E0
00000200 16 00 00 EA 11 F1 A0 E3 21 F1 A0 E3 31 F1 A0 E3 41 F1 A0 E3 51 F1 A0 E3 61 F1 A0 E3 71 F1 A0 E3 ê ñ ã!ñ ã1ñ ãAñ ãQñ ãañ ãqñ ã
00000220 00 52 45 56 40 16 00 00 45 54 41 44 44 16 00 00 45 5A 49 53 50 16 00 00 00 41 48 53 00 00 00 00 REV@ ETADD EZISP AHS
00000240
00000260 D1 F0 21 E3 06 CB A0 E3 FF 00 AC E8 0C 00 A0 E1 1F F0 21 E3 00 9F A0 E8 00 10 4F E1 02 60 A0 E8 Ñð!ã Ë ãÿ ¬è á ð!ã ▌ è Oá ` è
파일 헤더지만
아직은 모른다고
칩시다.

IDA로 열어봅시다♥

```
ROM:0005AF6C
ROM:0005AF6C ; =============== S U B R O U T I N E =======================================
ROM:0005AF6C
ROM:0005AF6C
ROM:0005AF6C sub_5AF6C                                 ; CODE XREF: sub_942920:loc_9429A4↓p
ROM:0005AF6C                 LDR     R0, =0x4118CA90
ROM:0005AF6E                 LDR     R0, [R0,#4]
ROM:0005AF70                 BX      LR
ROM:0005AF70 ; End of function sub_5AF6C
ROM:0005AF70
ROM:0005AF72
ROM:0005AF72 ; =============== S U B R O U T I N E =======================================
ROM:0005AF72
ROM:0005AF72
ROM:0005AF72 sub_5AF72                                 ; CODE XREF: sub_942920+14↓p
ROM:0005AF72                 LDR     R0, =0x4118CA90
ROM:0005AF74                 LDRB    R0, [R0]
ROM:0005AF76                 BX      LR
ROM:0005AF76 ; End of function sub_5AF72
ROM:0005AF76
ROM:0005AF78
ROM:0005AF78 ; =============== S U B R O U T I N E =======================================
ROM:0005AF78
ROM:0005AF78
ROM:0005AF78 sub_5AF78                                 ; CODE XREF: sub_942920+30↓p
ROM:0005AF78                 LDR     R1, =0x42EC84E4
ROM:0005AF7A                 MOVS    R0, #0
ROM:0005AF7C                 LDR     R2, =0x4118CA90
ROM:0005AF7E                 MOVS    R3, #0x12
ROM:0005AF80                 STR     R0, [R1,#0xC]
ROM:0005AF82                 STRB    R0, [R2]
ROM:0005AF84                 STR     R3, [R2,#4]
ROM:0005AF86                 ADR     R3, unk_5B038
ROM:0005AF88                 STR     R3, [R1,#4]
ROM:0005AF8A                 ADR     R3, unk_5B0A8
```

이게 뭐지!?ㅋㅋ

```
ROM:0005BDF7  0000001E  C  A<<< LTE DM Command List >>>₩n
ROM:0005BE18  00000011  C  [01] ltedm help₩n
ROM:0005BE2C  0000000F  C  [02] ltedm on₩n
ROM:0005BE3C  00000010  C  [03] ltedm off₩n
ROM:0005BE4C  00000014  C  [04] ltedm default₩n
ROM:0005BE60  0000001A  C  [05] ltedm vu     [RATE]₩n
ROM:0005BE7C  0000001B  C        ex) ltedm vu 1000 ₩n
ROM:0005BE98  0000002F  C             -> LTEDM_VU update rate is 1000ms₩n
ROM:0005BEC8  0000001A  C  [06] ltedm phy    [RATE]₩n
ROM:0005BEE4  0000001A  C  [07] ltedm l1     [RATE]₩n
ROM:0005BF00  0000001A  C  [08] ltedm mac    [RATE]₩n
ROM:0005BF1C  0000001A  C  [09] ltedm rlc    [RATE]₩n
ROM:0005BF38  0000001A  C  [10] ltedm pdcp   [RATE]₩n
ROM:0005BF54  0000001A  C  [11] ltedm rrc    [RATE]₩n
ROM:0005BF70  0000001A  C  [12] ltedm nas    [RATE]₩n
ROM:0005BF8C  0000001A  C  [15] ltedm pal    [RATE]₩n
ROM:0005BFA8  00000013  C  [16] ltedm status₩n
ROM:0005BFBC  00000024  C  ₩₩CALPSS₩₩LteCommon₩₩Code₩₩Src/lte_dm.c
ROM:0005BFE0  0000000D  C  LTE DM : ON₩n
ROM:0005BFF0  0000000E  C  LTE DM : OFF₩n
ROM:0005C003  00000018  C  @<<< LTE DM Status >>>₩n
ROM:0005C0...  00000010  C  %s Rate : %dms₩n
ROM:0005C0...  0000001C  C  LTE DM set a default value₩n
ROM:0005C048  00000017  C  LTE DM VU Rate : %dms₩n
```

LTE **D**evice **M**anager Command List

통신사 정보 알아내기!

- /data/data/com.android.providers.telephony/databases/telephony.db
  - 각 국 3g/4g 이동통신망 사업자의 접속 정보

21	21	Mobistar	20610	206	10	mworld.be	mobistar		mobistar	212.65.63.143	8080			
22	22	Base	20620	206	20	gprs.base.be	base		base	172.31.198.37	5080			
23	23	BASE MMS	20620	206	20	mms.base.be	base		base			217.72.235.1	8080	http://mmsc.bas
24	24	Orange World	20801	208	01	orange	orange		orange					
25	25	Orange MMS	20801	208	01	orange.acte	orange		orange			192.168.10.200	8080	http://mms.oran
26	26	Orange Entrepris	20801	208	01	orange-mib	orange		orange	172.16.2.8	8000			
27	27	Orange Internet	20801	208	01	orange.fr	orange		orange					
28	28	Orange Internet E	20801	208	01	internet-entrepris	orange		orange					

- /data/data/com.android.providers.telephony/databases/nwk_info.db
  - 현재 기기에 설정 되어 있는 이동통신망 접속 정보

_id	name	numeric	mcc	mnc	apn	user	server	password	proxy	port
1	KT IMS	45008	450	08	ims.ktfwing.com					
2	KT	45008	450	08	lte.ktfwing.com					

AT Command?

- 팩스 모뎀용으로 개발 됨.
- 상업적으로 표준화가 되어, 전 세계에서 사용 중.
- 일반적으로, Terminal을 통해 사용자와 모뎀이 명령어를 주고 받음.
- Android 에서 사용되는 3G Modem의 AT command?

AT Command LIST

- 3GPP의 범용 AT Command

명령어	기능
AT+CAAP	Automatic answer for eMLPP Service
AT+CACM	Accumulated call meter
AT+CAHLD	Leave an ongoing Voice Group or Voice Broadcast Call
AT+CFUN	Set phone functionality
AT+CREG	Network registration
AT+CMOD	Call mode
...이 아래 빙산이 있습니다.	

- 참고 자료 [5], [6]번을 참고 하시면 됨.

AT command의 사용 방법은??

- RIL Daemon 에서 사용 하는 디바이스 드라이버를 찾아야 함.
  - /system/build.prop
  - /dev/smd0 or /dev/ttyGS0
  - /dev/ttyS0

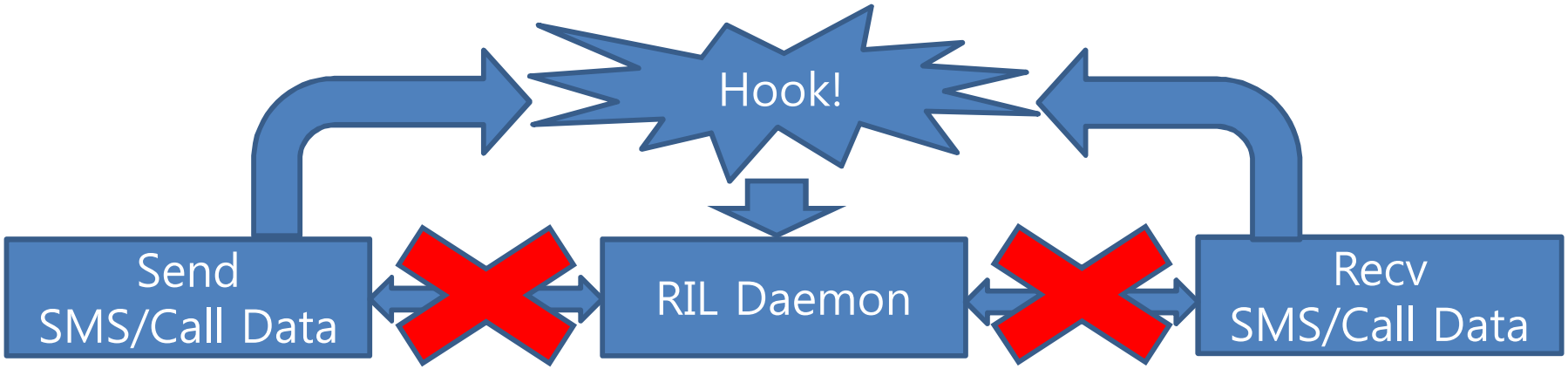
```
#
# system.prop for smdk4x12
#

rild.libpath=/system/lib/libsec-ril.so
rild.libargs=-d /dev/ttyS0
ro.sf.lcd_density=320
ro.lcd_min_brightness=20
```

- 명령어 전송은 어떻게 할까?
  - Exam> echo –ne "AT₩r" > [Device]
  - Galaxy S 시리즈에서도 가능.
  - 단, 해외 버전만...가능 ㅠ_ㅠ

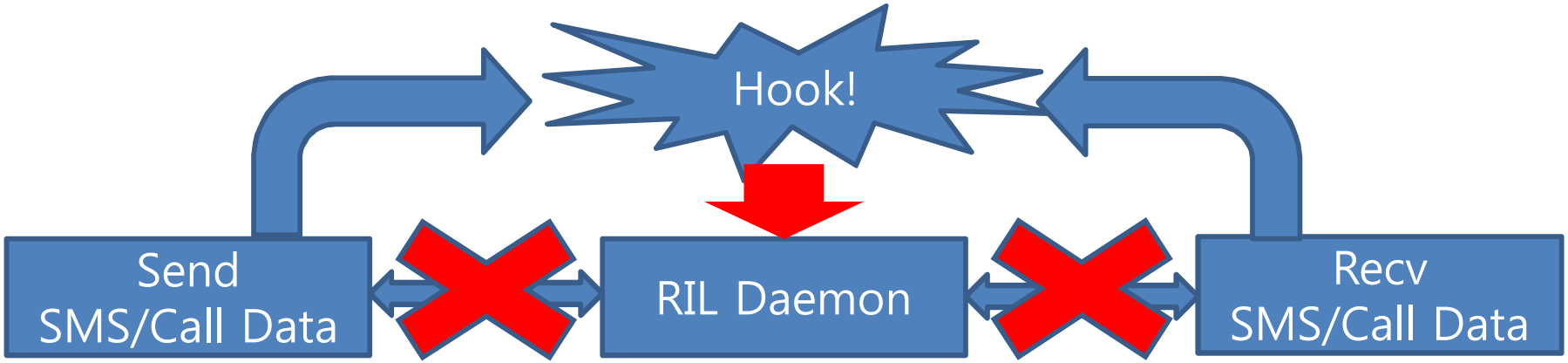
그럼 제 발표는 여기서...?

- AT Command는 전송은 디바이스 문제로 포기!
- 그럼 이제 무엇을? 이대로 진짜 끝?
- rild을 Hooking 해 보기!
- Rild을 통해 오고 가는 SMS/Call 데이터를 제어 하기!
- 잘 되려나? ㅋㅋㅋ

다시 한번 설명 해보면,



그냥 엿보기만!



데이터 변조!
퍼징!

무엇을 어떻게 할까?

- Android용으로 포팅 된 Hijack 툴을 사용.
  - http://www.mulliner.org/android/

- 후킹할 대상은?

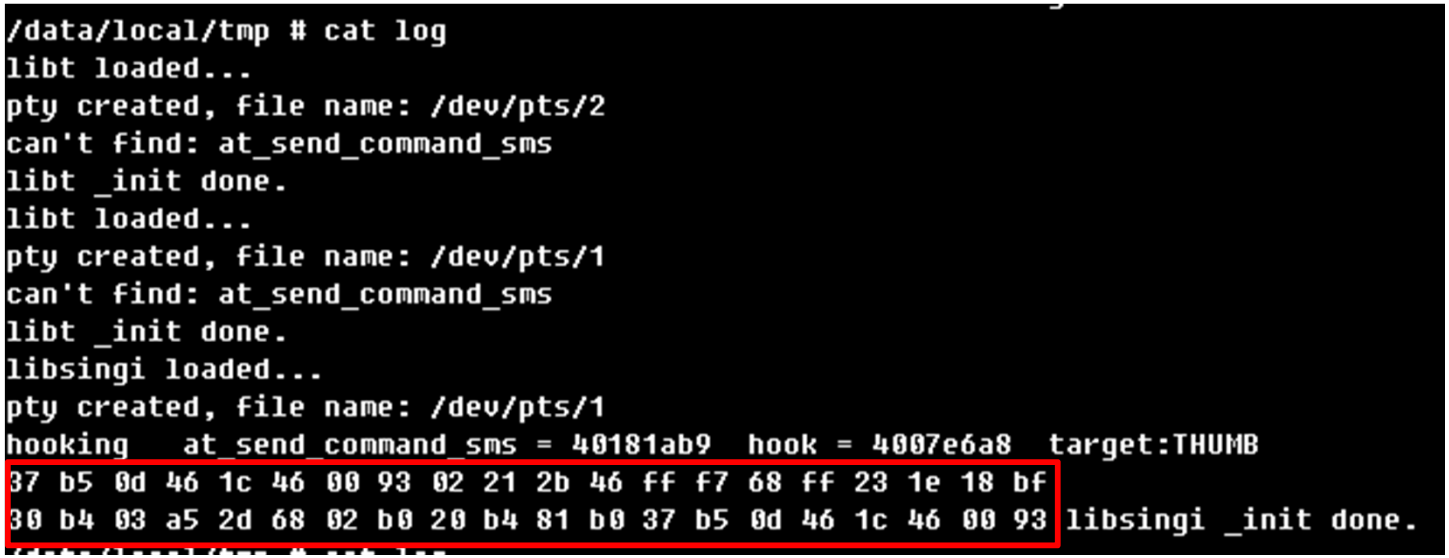
```
u0_a180@android:/data/data/berserker.android.apps.sshdroid/home # ps | grep rild
28684 radio      0:00 /system/bin/rild
28787 root       0:00 grep rild
u0_a180@android:/data/data/berserker.android.apps.sshdroid/home # cat /proc/28684/maps
```

```
401a4000-4025a000 r-xp 00000000 b3:09 1366     /system/lib/libsec-ril.so
4025a000-4025b000 r--p 000b5000 b3:09 1366     /system/lib/libsec-ril.so
4025b000-40260000 rw-p 000b6000 b3:09 1366     /system/lib/libsec-ril.so
```

잘 들어갔는지 확인!ㅋㅋ

```
/data/local/tmp # ./hijack -p 1669 -l libsingi.so -d
mprotect: 0x400e85d8
dlopen: 0xb000585d
pc=400e8c74 lr=400f52fd sp=beed3b68 fp=beed3d38
r0=fffffffc r1=beed3b70
r2=0 r3=0
stack: 0xbeeb3000-0xbeed4000 leng = 135168
executing injection code at 0xbeed3b18
library injection completed!
/data/local/tmp # cat /proc/1669/maps
00008000-00009000 r-xp 00000000 5d:18 183        /system/bin/rild
00009000-0000a000 rw-p 00001000 5d:18 183        /system/bin/rild
00832000-00837000 rw-p 00000000 00:00 0          [heap]
10000000-10001000 ---p 00000000 00:00 0
10001000-10100000 rw-p 00000000 00:00 0
40016000-40017000 r--p 00000000 00:00 0
40020000-40035000 r-xp 00000000 5d:18 818        /system/lib/libm.so
40035000-40036000 rw-p 00015000 5d:18 818        /system/lib/libm.so
40059000-4005c000 r-xp 00000000 5d:18 817        /system/lib/liblog.so
4005c000-4005d000 rw-p 00003000 5d:18 817        /system/lib/liblog.so
4005d000-4006c000 r-xp 00000000 5d:18 778        /system/lib/libcutils.so
4006c000-4006d000 rw-p 0000f000 5d:18 778        /system/lib/libcutils.so
4006d000-4007c000 rw-p 00000000 00:00 0
4007c000-4007f000 r-xp 00000000 5d:18 1248       /system/lib/libsingi.so
4007f000-40080000 rw-p 00002000 5d:18 1248       /system/lib/libsingi.so
40091000-40099000 r--s 00000000 00:0c 462        /dev/__properties__ (deleted)
40099000-400b0000 r-xp 00000000 5d:18 883        /system/lib/libz.so
400b0000-400b1000 rw-p 00017000 5d:18 883        /system/lib/libz.so
400b1000-400d4000 r-xp 00000000 5d:18 765        /system/lib/libbinder.so
400d4000-400da000 rw-p 00023000 5d:18 765        /system/lib/libbinder.so
400dc000-4011e000 r-xp 00000000 5d:18 766        /system/lib/libc.so
4011e000-40121000 rw-p 00042000 5d:18 766        /system/lib/libc.so
```

Opcode 패치 결과


/data/local/tmp # cat log
libt loaded...
pty created, file name: /dev/pts/2
can't find: at_send_command_sms
libt _init done.
libt loaded...
pty created, file name: /dev/pts/1
can't find: at_send_command_sms
libt _init done.
libsingi loaded...
pty created, file name: /dev/pts/1
hooking at_send_command_sms = 40181ab9 hook = 4007e6a8 target:THUMB
37 b5 0d 46 1c 46 00 93 02 21 2b 46 ff f7 68 ff 23 1e 18 bf
30 b4 03 a5 2d 68 02 b0 20 b4 81 b0 37 b5 0d 46 1c 46 00 93 libsingi _init done.

It’s fuzzing time!

[Demo]

(뻥인지 아닌지 시험!)

감사합니다!

(서로 질문은 없는걸로ㅋㅋㅋ)

- 참고 자료

[1] 쉽게 설명한 3G/4G 이동 통신 시스템/이상근 외 공저/홍릉 과학 출판사

[2] WiBro/WiMAX LTE 모바일 브로드밴드/강철희 외 편저/광문각

[3] http://www.kandroid.org/online-pdk/guide/telephony.html

[4] http://mobiledevdesign.com/tutorials/WIreless_Everywhere_Not_Quite_Yet/index2.html

[5] http://mod-book.ru/files/Gobi2k/Documents/AT_Command_Set_Gobi.pdf

[6] http://forum.xda-developers.com/showthread.php?t=1471241

[7] http://www.mulliner.org/android/