How secure are your VoLTE and VoWiFi calls?

Priya Chalakkal

ERNW
providing security.

About me : Priya Chalakkal

- ERNW GmbH, Heidelberg
- Loves telco, pcaps, binaries, logs, protocols and all security stuff in general.
- Completed Masters in Security and Privacy from TU, Berlin and UNITN, Trento.
- https://priyachalakkal.wordpress.com/
- https://insinuator.net/

ERNW
providing security.

Agenda

- Introduction
- Fundamentals
- PART1: Attacks on OpenIMS (without IPSec)
- PART2: Attacks on real telecom providers (with IPSec)
- Demo
- Mitigation

Introduction - Telephony

Circuit Switched

- PSTN : *Public Switched Telephone Networks*
- Dedicated circuit – "Channel"
- Roots tracked back to 1876
  - Graham Bell got the first patent

Packet Switched

- Data sent as Packets
- Protocol stack: TCP/IP
- Eg:- Internet
- For voice - VoIP

Introduction - VoIP

ERNW
providing security.

Introduction – VoLTE/VoWiFi

VoLTE

- SK Telecom and LG U+Objective South Korea – 2012
- Vodafone Germany – VoLTE – March 2015

VoWiFi:

- Telekom Germany – VoWiFi – May 2016
- WiFi Calling

FUNDAMENTALS

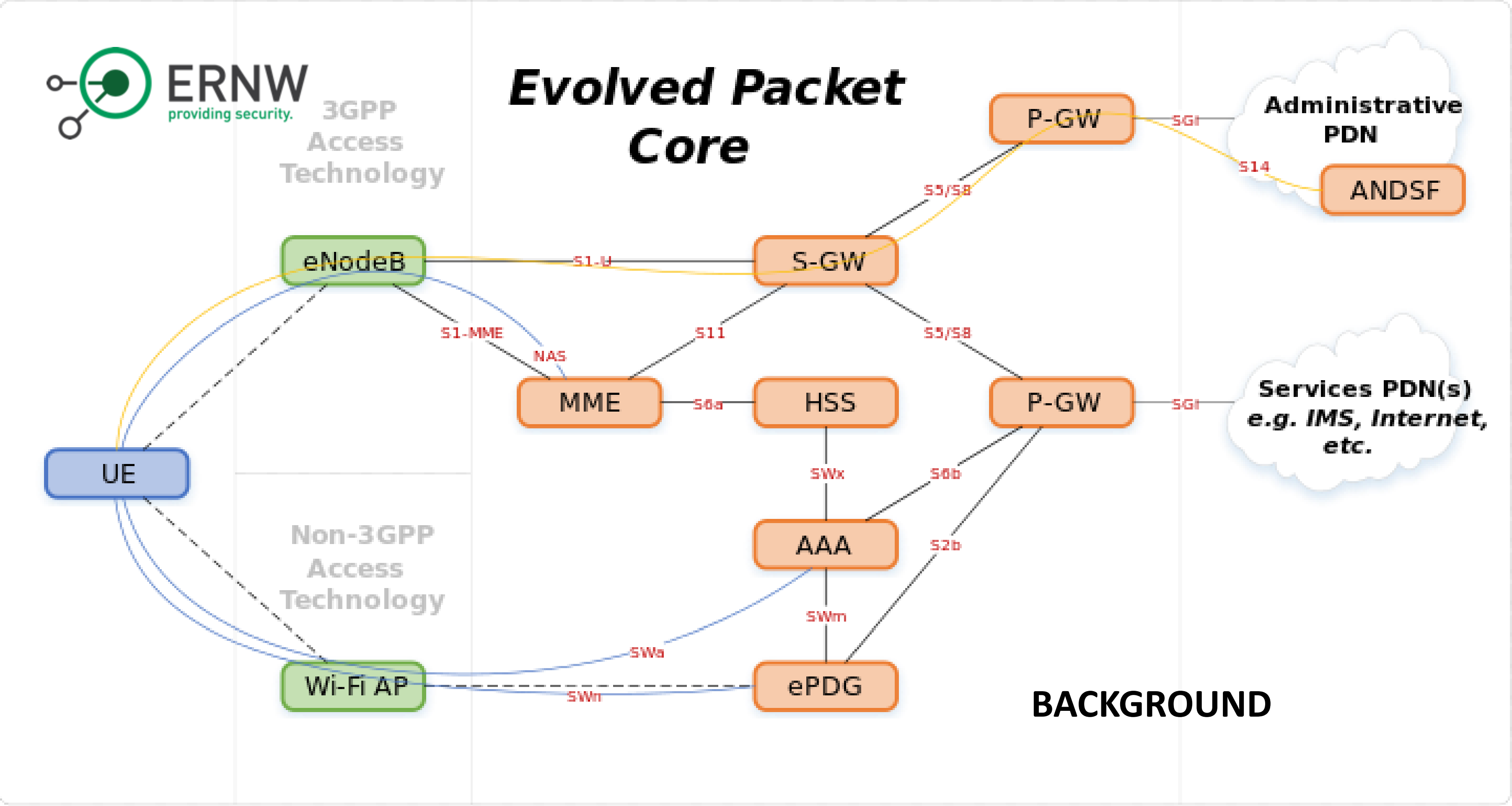
History of Mobile Communication

- GSM (2G)
  - Relies on Circuit Switching
  - Supports only Voice and SMS
- GPRS
  - Circuit – voice and SMS
  - Packet – Data
- UMTS (3G)
  - Similar to GPRS
  - Other network elements evolved

Voice and 4G

- LTE (4G): Supports only packet switching
- **Voice - VoLTE**
- **Circuit Switched Fall Back (CSFB)**
  - For voice, fall back to circuit switched networks.
- Other approaches
  - Simultaneous voice and LTE

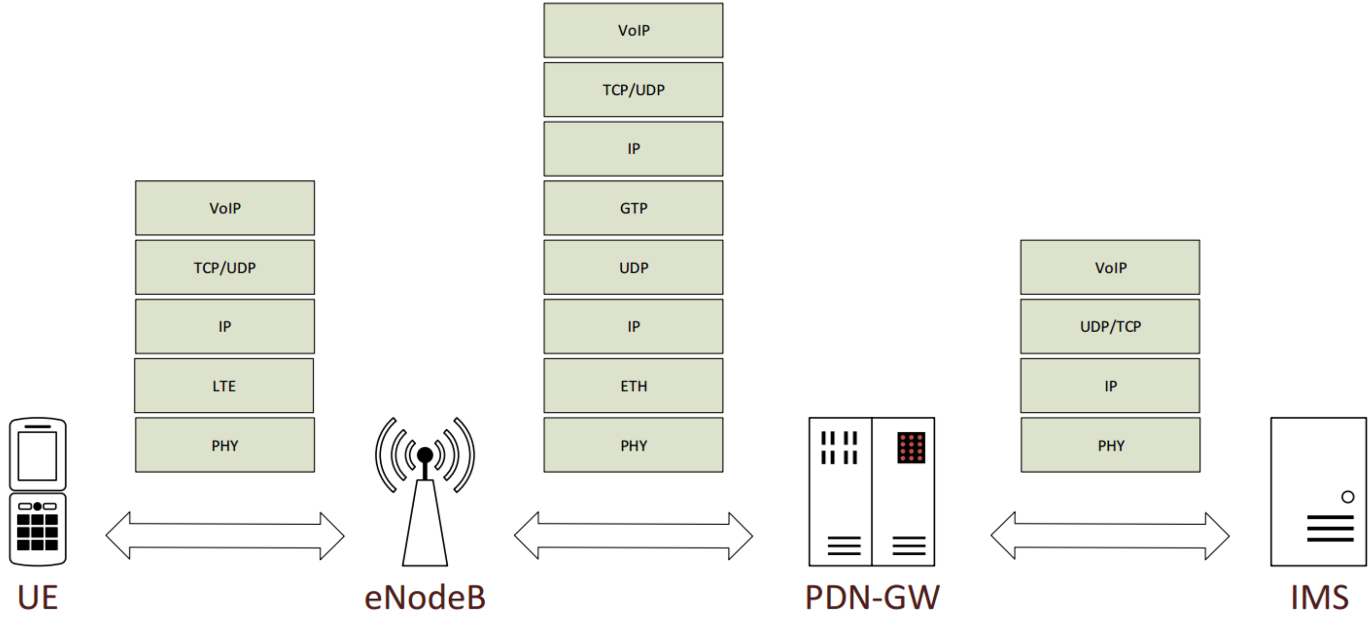
  etc..



BACKGROUND

Source: https://en.wikipedia.org/wiki/System_Architecture_Evolution

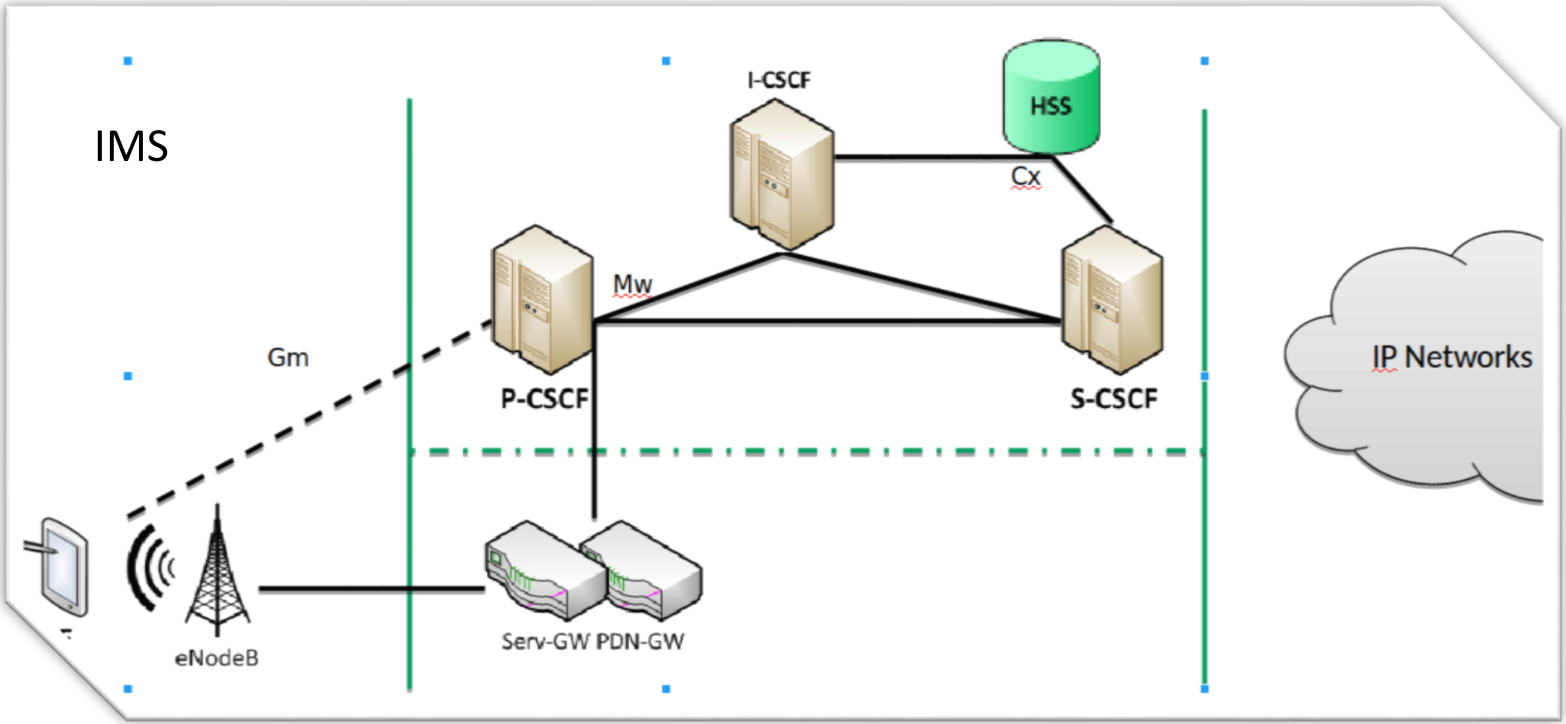
VoLTE Stack


VoIP
TCP/UDP
IP
LTE
PHY
VoIP
TCP/UDP
IP
GTP
UDP
IP
ETH
PHY
VoIP
UDP/TCP
IP
PHY
UE
eNodeB
PDN-GW
IMS

IMS – IP Multimedia Subsystem

- Backend: IMS Core
  - *IP Multimedia Subsystem*
  - Call session control functions (CSCF)
    - P-CSCF
    - S-CSCF
    - I-CSCF

ERNW
providing security.



IMS Signaling

SIP - Session Initiation Protocol

- Similar to HTTP (text based)
- TCP or UDP
- Contains SDP
    - Session Description Protocol
    - Describing multimedia session
    - Eg:- audio/video type

SIP call session

```
                     atlanta.com  . . . biloxi.com
                 .      proxy              proxy     .
               .                                       .
       Alice's  . . . . . . . . . . . . . . . . . . . .  Bob's
      softphone                                        SIP Phone
         |                |                |                |
         |    INVITE F1   |                |                |
         |--------------->|    INVITE F2   |                |
         |  100 Trying F3 |--------------->|    INVITE F4   |
         |<---------------|  100 Trying F5 |--------------->|
         |                |<-------------- | 180 Ringing F6 |
         |                | 180 Ringing F7 |<---------------|
         | 180 Ringing F8 |<---------------|     200 OK F9  |
         |<---------------|    200 OK F10  |<---------------|
         |    200 OK F11  |<---------------|                |
         |<---------------|                |                |
         |                       ACK F12                    |
         |------------------------------------------------->|
         |                   Media Session                  |
         |<================================================>|
         |                       BYE F13                    |
         |<-------------------------------------------------|
         |                     200 OK F14                   |
         |------------------------------------------------->|
         |                                                  |
```

```
INVITE sip: jennifer@csp.com SIP/2.0
Via: SIP/2.0/UDP [5555::a:b:c:d]:1400; branch=abc123
Max-Forwards:70
Route: <sip:[5555::55:66:77:88]:7531;lr>,< sip:orig@scscf1.home.fi;lr>
P-Access-Network-Info:3GPP-E-UTRAN-TDD;utran-cell-id-3gpp=244005F3F5F7
P-Preferred-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel
Privacy: none
From: <sip:kristiina@example.com>;tag=171828
To: <sip:jennifer@csp.com>
Call-ID: cb03a0s09a2sdfglkj490333
Cseq: 127 INVITE
Require: sec-agree
Proxy-Require: sec-agree
Supported: precondition, 100rel, 199
Security-Verify: ipsec-3gpp; alg=hmac-sha-1-96; spi-c=98765432;
spi-s=87654321; port-c=8642; port-s=7531
Contact: <sip:[5555::a:b:c:d]:1400;+g.3gpp.icsi-ref="urn%3Aurn-7%
3gpp-service.ims.icsi.mmtel"
Accept-Contact: *;+g.3gpp.icsi-ref="urn%3Aurn-7%
3gpp-service.ims.icsi.mmtel"
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER, MESSAGE, OPTIONS
Accept:application/sdp, application/3gpp-ims+xml
Content-Type: application/sdp
Content-Length: (...)

v=0
o=- 2890844526 2890842807 IN IP6 5555::a:b:c:d
s=-
c=IN IP6 5555::a:b:c:d
t=0 0
m=audio 49152 RTP/AVP 97 98
a=rtpmap:97 AMR/8000/1
a=fmtp:97 mode-change-capability=2; max-red=220
b=AS:30
b=RS:0
b=RR:0
a=rtpmap:98 telephone-event/8000/1
a=fmtp:98 0-15
a=ptime:20
a=maxptime:240
a=inactive
a=curr:qos local none
```

SIP

SDP

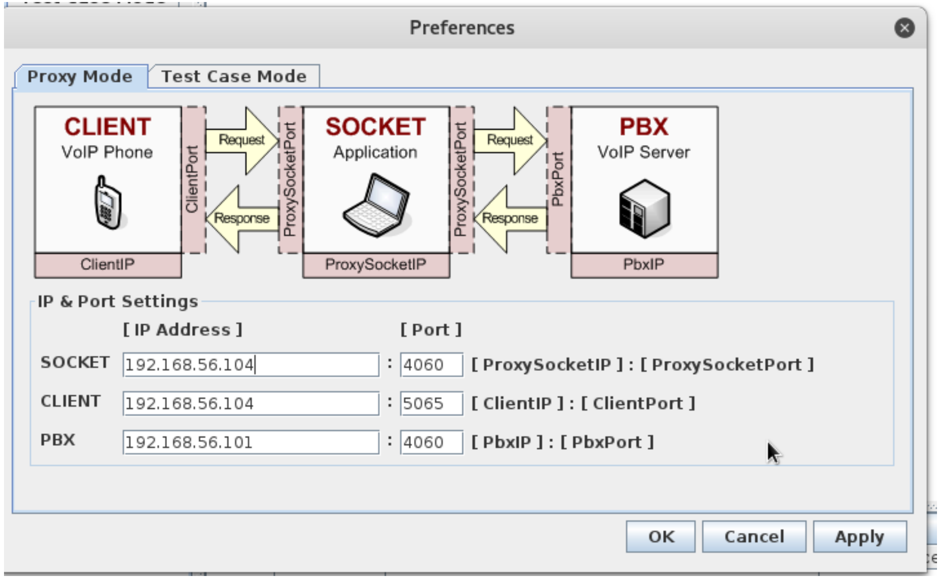
10

PART1: Attacking OpenIMS

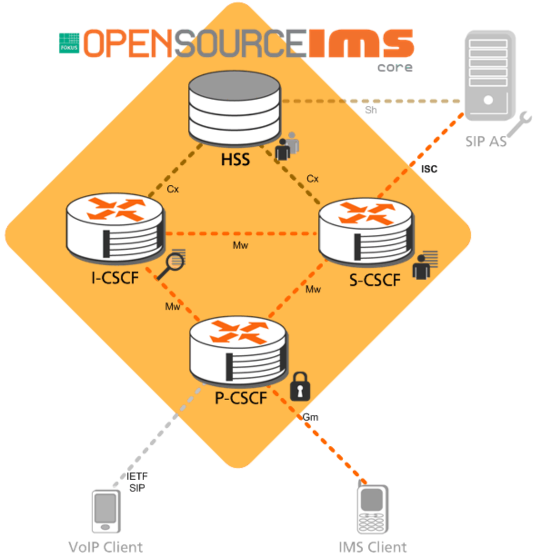
ERNW
providing security.

Requirements

- OpenIMS
- SIP Proxy
- Viproy toolkit for Attack1
- IMS clients – twinkle (in ubuntu), boghe (in windows)



Preferences
Proxy Mode
Test Case Mode
CLIENT
VoIP Phone
ClientIP
ClientPort
Request
Response
ProxySocketPort
SOCKET
Application
ProxySocketIP
ProxySocketPort
Request
Response
PbxPort
PBX
VoIP Server
PbxIP
IP & Port Settings
[IP Address]
[Port]
SOCKET 192.168.56.104 : 4060 [ProxySocketIP] : [ProxySocketPort]
CLIENT 192.168.56.104 : 5065 [ClientIP] : [ClientPort]
PBX 192.168.56.101 : 4060 [PbxIP] : [PbxPort]
OK
Cancel
Apply



OPENSOURCEIMS
core
HSS
Sh
SIP AS
ISC
Cx
Cx
I-CSCF
Mw
S-CSCF
Mw
Mw
P-CSCF
Gm
IETF
SIP
VoIP Client
IMS Client

ERNW
providing security.

Attack modeling

- VoLTE and VoWiFi makes use of SIP
- This is experimental tests on OpenIMS with desktop clients
- Mainly SIP header injection
- Without IPSec in any communication
- Both attacker and victim is a registered user.

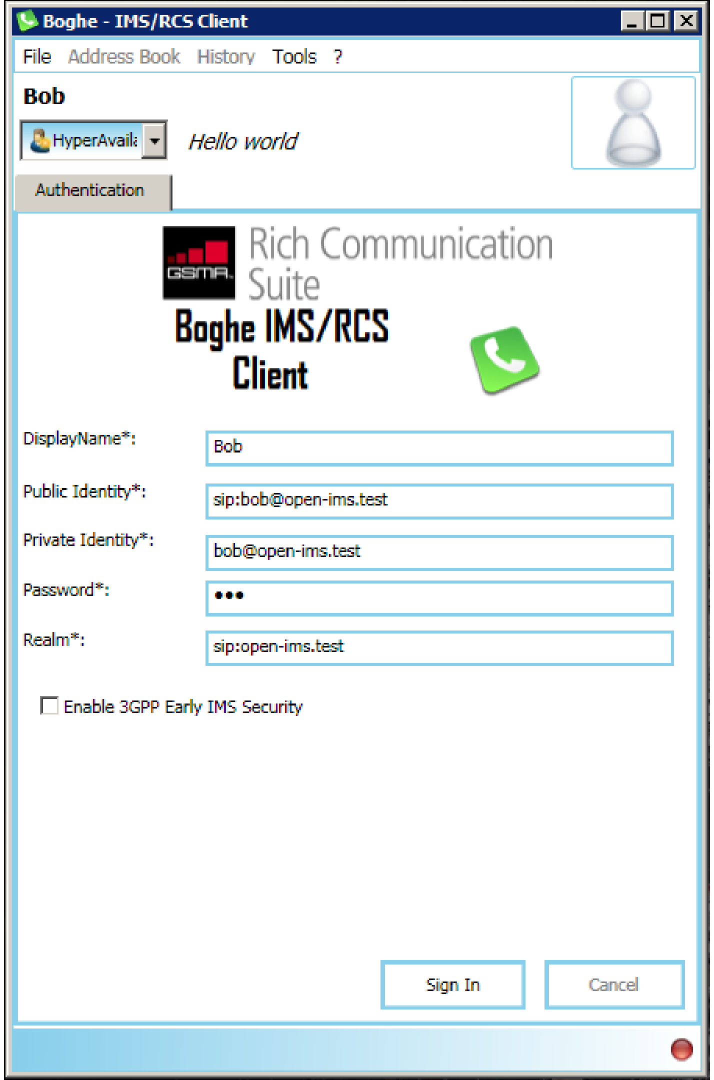
```
       =[ metasploit v4.13.5-dev                          ]
+ -- --=[ 1607 exploits - 943 auxiliary - 276 post        ]
+ -- --=[ 458 payloads - 39 encoders - 9 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf auxiliary(viproy_msrp_header_fuzzer_with_invite) >
```

Attack1: MSRP fuzzing

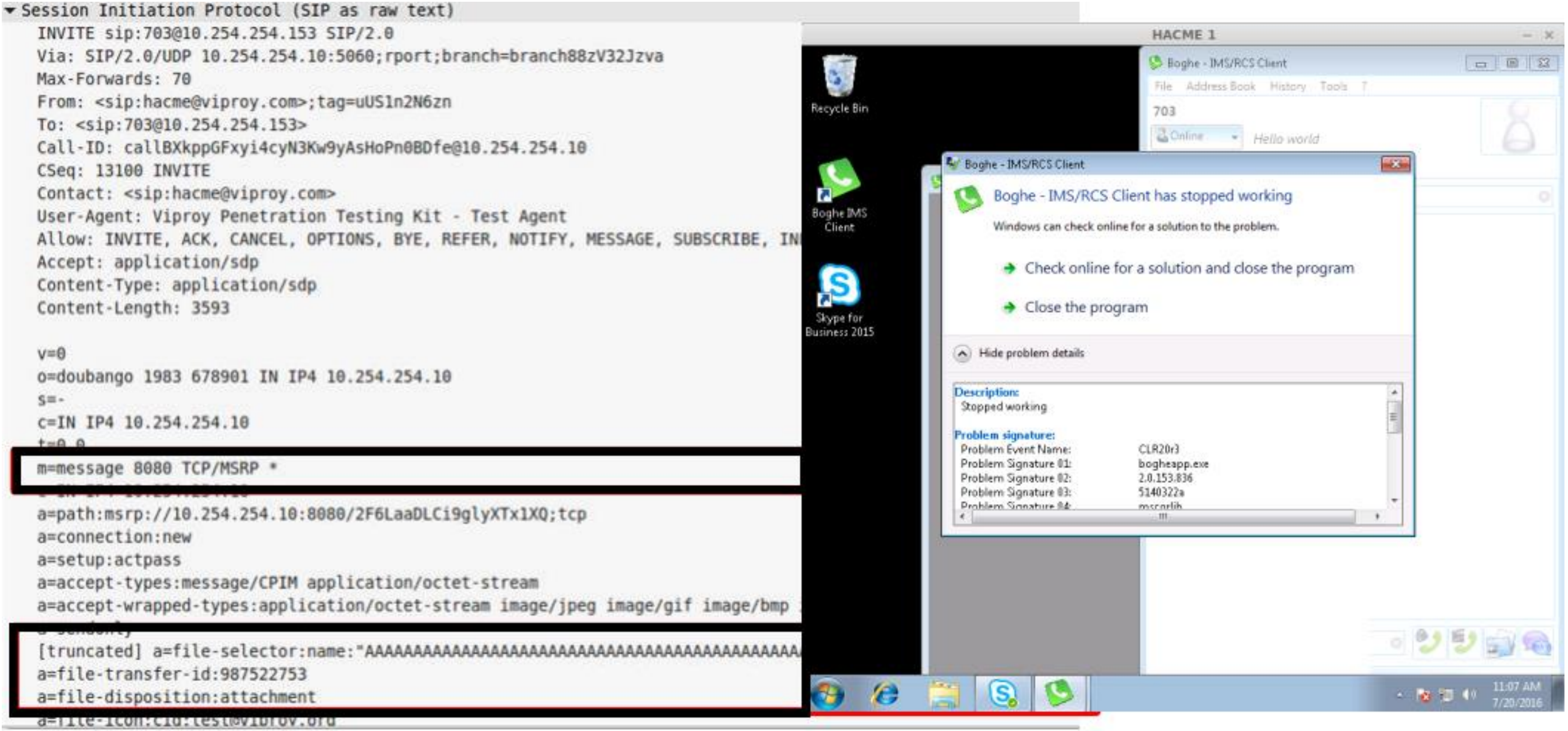
- MSRP – protocol for transmission of series of related instant messages in context of communication session
- Evil sends fuzzed input in one of the MSRP header field to Alice
  - a=file-selector:name:"AAAAAAAAAAA..."
- This is an automated test vector in Viproy toolkit.

Result 1

- Crashes the IMS client of Receiver (Boghe IMS client is used in this case)
- Neither IMS nor client performed input validation.



Result1: MSRP fuzzing



Source: Fatih Ozvaci- Voip wars: The phreakers awaken

Attack2: Location manipulation

- P-Access-Network-Info - defines the user location in the access network
- Contains information such as:
  - Mobile Network Code (MNC)
  - Mobile Country Code (MCC)
  - Local Area Code (LAC)
  - Cell Identifier
- The attacker sends an INVITE request to Alice with a crafted location.

Result2

- Modified P-Access-Network-Info is accepted by IMS and sent to Alice
- No cross validation with HSS for user location.
- Can evade **lawful interception techniques.**
- NOT about privacy

Attack3: Roaming Information

- P-Visited-Network-ID header field that decides the access network that serves the user.
- Attacker sends a REGISTER request to IMS with an pre-added P-Visited-Network-ID header.

Result3

- P-CSCF just appends the network identity to the existing header field
- Attacker can use this to make his roaming calls as local calls

Output from S-CSCF packet dump:

P-Visited-Network-ID: open-ims_fake.test, open-ims.test

Attack4: Extra header field

- SIP protocol is an extensible protocol
  - Allows to add customized header fields
- Evil sends an INVITE request to Alice containing a custom header field X-Header

ERNW
providing security.

Result4

```
▷ Via: SIP/2.0/UDP 127.0.0.1:6060;received=127.0.0.1;rport=6060;branch=z9hG4bK3fc4
▷ Via: SIP/2.0/UDP 127.0.0.1:6060;branch=z9hG4bK3fc4.07ebc004.0
▷ Via: SIP/2.0/UDP 0.0.0.0:4060;received=127.0.0.1;branch=z9hG4bK3fc4.d87f5ce1.0
▷ Via: SIP/2.0/UDP 192.168.56.103:5060;rport=40303;branch=z9hG4bK79178419f7f6d3d08
  Max-Forwards: 13
  X-Header: "This is an extra header, I will send it to you for free"
  Content-Type: application/sdp
```


Source https://insinuator.net/2017/02/exploitation-of-ims-in-absence-of-confidentiality-and-integrity-protection/

More attack possibilities

- Spoofing
- Injection – XML, SQL,
- Denial of Service
- Fuzzing
- …
- …

Attacking OpenIMS summary

- 4 attacks on OpenIMS
  - MSRP fuzzing
  - User location manipulation
  - Roaming information manipulation
  - Extra header field injection
- These are **Man in the End** attacks
- **Without** IPSec

ERNW
providing security.

How to prevent tampering SIP Attacks?

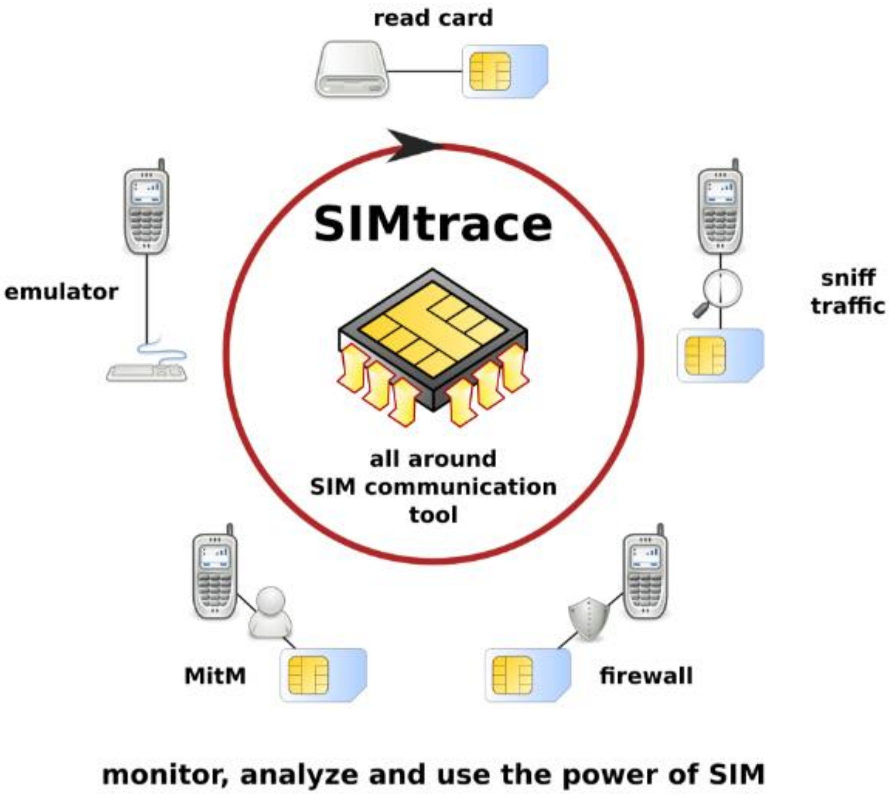
- Bring integrity protection?
- Can IPSec solve this?
- Many real telecom provides actually have IPSec in place.
- Can we still mess with SIP headers in real providers?

PART2: ATTACKING TELECOM PROVIDERS

ERNW
providing security.

Requirements

- VoLTE/VoWiFi enabled SIM cards
- SIMTrace hardware
- VoLTE/VoWiFi enabled phones
- Wireshark - Gcrypt



http://shop.sysmocom.de/products/simtrace

ERNW
providing security.

Attack modeling

- Sniffing VoLTE – rmnet0, rmnet1
- Sniffing VoWiFi – epdg1, wlan0
- Sniffing ISIM interface using SIMTrace
- IPSec
  - ESP encapsulation for both VoLTE and VoWiFi
  - Integrity protection enabled for VoLTE/VoWiFi
  - Encryption for VoWiFi (only in wlan0)

ESP Packets

Apply a display filter … <Ctrl-/>

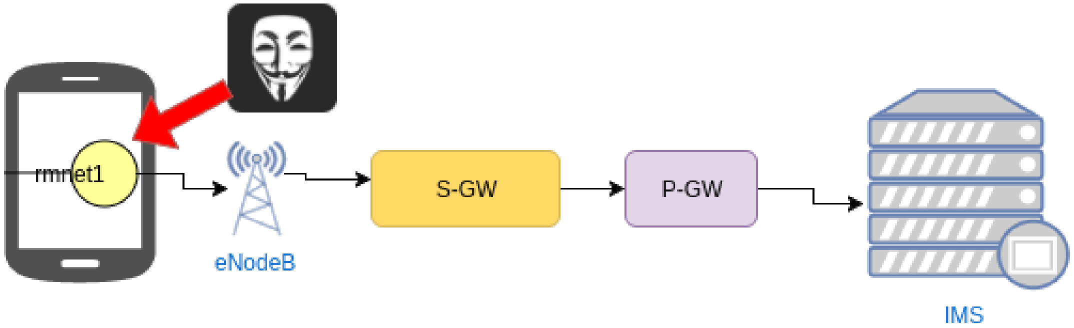
No.	Time	Source	Destination	Protocol	Length	Info
1	2016-10-12 04:51:54.040307	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	1256	ESP (SPI=0x8115e84f)
2	2016-10-12 04:51:54.129889	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	1204	ESP (SPI=0x00001534)
3	2016-10-12 04:51:54.155814	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	1364	ESP (SPI=0x00001533)
4	2016-10-12 04:51:54.156085	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	1364	ESP (SPI=0x00001533)
5	2016-10-12 04:51:54.156311	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	1364	ESP (SPI=0x00001533)
6	2016-10-12 04:51:54.156688	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	84	ESP (SPI=0x8115e84f)
7	2016-10-12 04:51:54.157246	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	84	ESP (SPI=0x8115e84f)
8	2016-10-12 04:51:54.157701	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	84	ESP (SPI=0x8115e84f)
9	2016-10-12 04:51:54.161144	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	1364	ESP (SPI=0x00001533)
10	2016-10-12 04:51:54.161794	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	300	ESP (SPI=0x00001533)
11	2016-10-12 04:51:54.161938	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	84	ESP (SPI=0x8115e84f)
12	2016-10-12 04:51:54.162481	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	84	ESP (SPI=0x8115e84f)
13	2016-10-12 04:51:54.219780	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	744	ESP (SPI=0x8115e84f)
14	2016-10-12 04:51:54.261618	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	84	ESP (SPI=0x00001533)
15	2016-10-12 04:51:58.534180	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	1340	ESP (SPI=0x8115e84f)
16	2016-10-12 04:51:58.534246	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	1112	ESP (SPI=0x8115e84f)
17	2016-10-12 04:51:58.582614	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	84	ESP (SPI=0x00001533)
18	2016-10-12 04:51:58.582923	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	84	ESP (SPI=0x00001533)
19	2016-10-12 04:51:58.788646	2a01:598:400:3002::5	2a01:59f:a021:caf7:2:2:d483:4be0	ESP	456	ESP (SPI=0x00001533)
20	2016-10-12 04:51:58.789033	2a01:59f:a021:caf7:2:2:d…	2a01:598:400:3002::5	ESP	84	ESP (SPI=0x8115e84f)

Test 1: Sniffing VoLTE/VoWiFi Interfaces

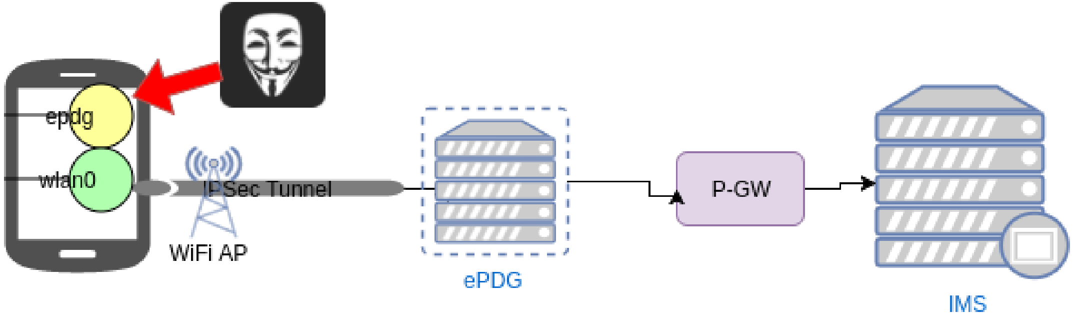
- VoLTE – rmnet1/rmnet0
- VoWiFi –
  - Epdg1 – hidden virtual interface with **non-encrypted** traffic
  - Wlan0 – encrypted traffic

```
Sniffing VoLTE interface :
$ adb shell
$ tcpdump -i rmnet1 -n -s 0 -w - | nc -l 127.0.0.1 -p 11233
$ adb forward tcp:11233 tcp:11233 && nc 127.0.0.1 11233 | wireshark -k -S -i -
```

VoLTE sniffing



VoWiFi sniffing



Observations

- No encryption in VoLTE
  - Only integrity with ESP
- Encryption in VoWiFi
- Hidden interface with **non-encrypted** traffic detected in VoWiFi

Results1: Information disclosures

```
▾ Session Initiation Protocol (INVITE)
  ▾ Request-Line: INVITE sip:+[redacted]@ims.telekom.de;user=phone SIP/2.0
      Method: INVITE
    ▸ Request-URI: sip:[redacted]@ims.telekom.de;user=phone
      [Resent Packet: False]
  ▾ Message Header
      Content-Length: 828
    ▸ Route: <sip:[2a01:598:400:3002::5]:5063;lr>,<sip:[2A01:598:400:3002::5]:5063;transport=TCP;lr>
      Allow: INVITE,ACK,OPTIONS,CANCEL,BYE,UPDATE,INFO,REFER,NOTIFY,MESSAGE,PRACK
    ▸ Via: SIP/2.0/TCP [2a01:59f:a021:caf7:2:2:d483:4be0]:6000;branch=z9hG4bK1465682047smg;transport=TCP
      User-Agent: SM-G920F-XXU4DPGU Samsung IMS/5.0
      P-Access-Network-Info: IEEE-802.11;i-wlan-node-id=[redacted]
      Supported: 100rel,timer,precondition,histinfo,sec-agree,gruu
    ▸ Security-Verify: ipsec-3gpp;q=0.5;alg=hmac-sha-1-96;prot=esp;mod=trans;ealg=null;spi-c=3132874533;
      Proxy-Require: sec-agree
      Require: sec-agree
    ▾ Contact: <sip:+[redacted]@[2a01:59f:a021:caf7:2:2:d483:4be0]:6000>;+g.3gpp.icsi-ref="urn%3Aurn-
      ▸ Contact URI: sip:+[redacted]@[2a01:59f:a021:caf7:2:2:d483:4be0]:6000
        Contact parameter: +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel"
        Contact parameter: +sip.instance="<urn:gsma:imei:[redacted]>"\r\n
      Max-Forwards: 70
    ▸ CSeq: 1 INVITE
      Call-ID: 3771911545@2a01:59f:a021:caf7:2:2:d483:4be0
    ▸ To: <sip:+[redacted]@ims.telekom.de;user=phone>
    ▸ From: <sip:+[redacted]@ims.telekom.de>;tag=3835380880
      Content-Type: application/sdp
      Accept-Contact: *;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel"
      Accept: application/sdp,application/3gpp-ims+xml
      Session-Expires: 1800;refresher=uac
```

- IMEI in SIP REGISTER (before authentication)

```
Contact:
<sip:262011202xxxxxx@[x.x.x.x]:6000>;q=0.50;+g.3gpp.icsi-ref=
"urn%3Aurn-7%3A3gpp-service.ims.xxx";
+g.3gpp.smsip;+sip.instance="<urn:gsma:imei:35490xxx-xxxxxx-0>"
```

- UTRAN Cell ID
  - outgoing packets like SIP REGISTER, outgoing SIP INVITE, SIP SUBSCRIBE messages contains the location information

```
##FOR VOLTE
INVITE sip:alice@open-ims.test SIP/2.0
...
User-Agent: Samsung IMS/5
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=00000001
Content-Length: 117

##FOR VOWIFI
P-Access-Network-Info:IEEE-802.11;i-wlan-node-id=003a9axxxxxx
```

- IMEI of caller
    - SIP INVITE incoming request consists of a parameter that contains the IMEI number of the caller.

```
Accept-Contact:*;+sip.instance="<urn:gsma:imei:354xxxxx7-
xxxxxx-0>";+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-
service.ims.xxxx";explicit;require
```

- IMSI of caller leaked
  - In SIP INVITE incoming request

```
INVITE sip:262011202xxxx@[x.x.x.x]:6000 SIP/2.0
```

Private IP of IMS

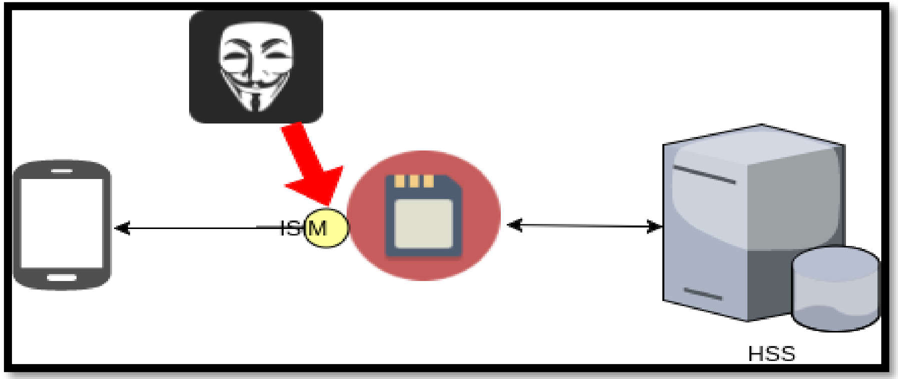
- Found within SIP INVITE in incoming calls

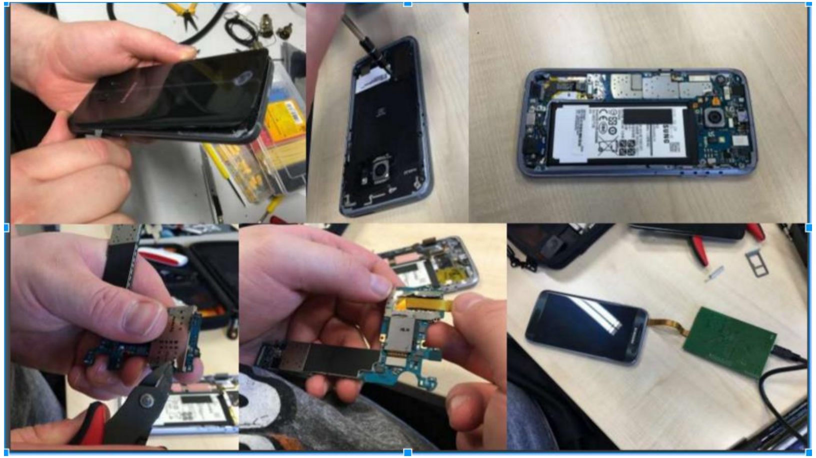
```
To: <sip:+49151xxxxxxxx@62.xxx.xxx.xxx>
From: <sip:+49176xxxxxxxx@10.xxx.xxx.xxx>;
tag=h7g4Esbg_mavodi-a-10b-3c-2-ffffffff-
_000050ED9CA4-1224-xxxx-xxxx
```

Test 2: ISIM sniffing for extracting CK/IK

```
└[~/thesis/simtrace/host]> sudo ./simtrace
simtrace - GSM SIM and smartcard tracing
(C) 2010 by Harald Welte <laforge@gnumonks.org>

Entering main loop
ATR APDU: 3b 9f 96 80 1f c6 80 31 e0 73 fe 21 1b 66 d0 02 06 e2 0f 18 01 f0
PPS(Fi=9/Di=6) APDU: 00 a4 00 04 02 3f 00 61 2e
APDU: 00 c0 00 00 2e 62 2c 82 02 78 21 83 02 3f 00 a5 09 80 01 61 83 04 00 00 57 6a 8a 01 05 8b 03
APDU: 00 a4 00 0c 02 2f e2 90 00
APDU: 00 b0 00 00 0a 98 94 20 00 00 21 09 68 85 19 90 00
APDU: 00 a4 00 04 02 2f 05 61 1e
APDU: 00 c0 00 00 1e 62 1c 82 02 41 21 83 02 2f 05 a5 03 80 01 61 8a 01 05 8b 03 2f 04 04 02 00 08
APDU: a4 00 04 02 a4 2f 06
APDU: 61 21 00 c0 00 00 21
APDU: c0 62 1f 82 05 42 21
APDU: 00 38 08 83 02 2f 06
APDU: a5 03 80 01 61 8a 01
APDU: 05 8b 03 2f 06 01 80
APDU: 02 01 c0 88 01 30 90
APDU: 00 00 b2 04 04 38 b2
APDU: 80 01 18 a4 06 83 01
APDU: 0b 95 01 08 80 01 02
APDU: a0 18 a4 06 83 01 01
APDU: 95 01 08 a4 06 83 01
APDU: 0b 95 01 08 a4 06 83
APDU: 01 0c 95 01 08 80 01
APDU: 01 90 00 84 01 d4 a4
APDU: 06 83 01 0b 95 01 08
APDU: 90 00 00 a4 00 04 02
APDU: a4 2f 05 61 1e 00 c0
APDU: 00 00 1e c0 62 1c 82
APDU: 02 41 21 83 02 2f 05
APDU: a5 03 80 01 61 8a 01
APDU: 05 8b 03 2f 06 04 80
APDU: 02 00 08 88 01 28 90
APDU: 00 00 b0 00 00 08 b0
APDU: 64 65 65 6e ff ff ff
APDU: ff 90 00 80 10 00 00
APDU: 20 10 ff ff ff ff 7f
APDU: 9d 00 df ff 00 1f e2 00 00 00 c3 eb 00 00 00 01 48 00 50 00 00 00 00 08 00 00 60 91 0f 00 a4
83 02 2f 00 a5 03 80 01 61 8a 01 05 8b 03 2f 06 07 80 02 00 2c 88 01 f0 91 0f 00 a4 00 04 02 a4 2f
03 80 01 61 8a 01 05 8b 03 2f 06 01 80 02 01 c0 88 01 30 91 0f 00 b2 07 04 38 b2 80 01 1a a4 06 83
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 91 0f 00 a4 00 04 02 a4
```

ISIM sniffing with SIMTrace





Security protocol: EAP-AKA

```
Peer                                                 Authenticator
   |                     EAP-Request/Identity                  |
   |<----------------------------------------------------------|
   |                                                           |
   | EAP-Response/Identity                                     |
   | (Includes user's NAI)                                     |
   |---------------------------------------------------------->|
   |                             +-------------------------------+
   |                             | Server runs AKA algorithms,   |
   |                             | generates RAND and AUTN.      |
   |                             +-------------------------------+
   |                          EAP-Request/AKA-Challenge        |
   |                          (AT_RAND, AT_AUTN, AT_MAC)       |
   |<----------------------------------------------------------|
+---------------------------------------+                      |
| Peer runs AKA algorithms,             |                      |
| verifies AUTN and MAC, derives RES    |                      |
| and session key                       |                      |
+---------------------------------------+                      |
   | EAP-Response/AKA-Challenge                                |
   | (AT_RES, AT_MAC)                                          |
   |---------------------------------------------------------->|
   |                           +---------------------------------+
   |                           | Server checks the given RES,    |
   |                           | and MAC and finds them correct.|
   |                           +---------------------------------+
   |                                            EAP-Success    |
   |<----------------------------------------------------------|
```

GSM SIM traffic

Source	Destination	sport	dport	Protocol	Info
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=35072
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 SELECT File DF.GSM-ACCESS
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 SELECT File 4f52
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 GET RESPONSE
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=0
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 SELECT File ADF
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 SELECT File EF.PSLOCI
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 GET RESPONSE
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=0
127.0.0.1	127.0.1.1	49482	53	DNS	Standard query 0x5e58 A prx1.ernw.net
127.0.1.1	127.0.0.1	53	49482	DNS	Standard query response 0x5e58 A prx1.ernw.net A 62.159.96.83
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=36608
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=36608
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 SELECT /ADF
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 RUN GSM ALGORITHM / AUTHENTICATE
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 GET RESPONSE
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=40448
127.0.0.1	127.0.0.1	42129	4729	GSM SIM	ISO/IEC 7816-4 UPDATE BINARY Offset=35584

What can we find here?

- AKA parameters –
  - RAND - random challenge
  - AUTN – server authentication
- IPSec keys
  - IK – integrity key
  - CK – cyphering key

ERNW
providing security.

How to extract it?

- Wireshark dissector

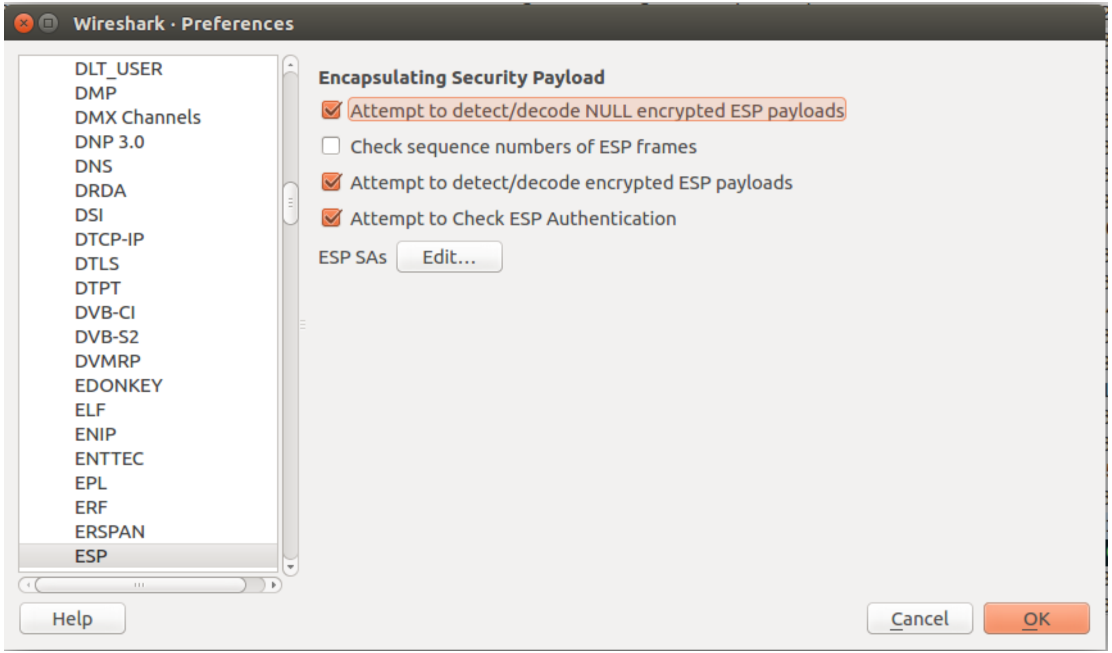
Byte(s)	Description	Length
1	'Successful 3G authentication' tag = 'DB'	1
2	Length of RES (L3)	1
3 to (L3+2)	RES	L3
(L3+3)	Length of CK (L4)	1
(L3+4) to (L3+L4+3)	CK	L4
(L3+L4+4)	Length of IK (L5)	1
(L3+L4+5) to (L3+L4+L5+4)	IK	L5

Table 4.4: Parsing the ISIM Authenticate response to get IK and CK

Result2: Extracting IK/CK

```
▸ User Datagram Protocol, Src Port: 52725 (52725), Dst Port: 4729 (4729)
▾ GSM SIM 11.11
    0000 .... = Class Coding: ISO/IEC 7816-4 (0x00)
    .... 00.. = Secure Messaging Indication: No SM used between terminal and card (0x00)
    .... ..00 = Logical Channel number: 0
    Instruction: GET RESPONSE (0xc0)
    Length of Expected Response Data: 53
    RES Length: 08
    RES Value: f74105e9ac41cc7a
    CK Length: 10
    CK: 3ee2824f414d4be3ddea7807a68632fa
    IK Length: 10
    IK : 347c59d30bba9f1968285908f89f996c
    Status Word: 9000 Normal ending of the command
```

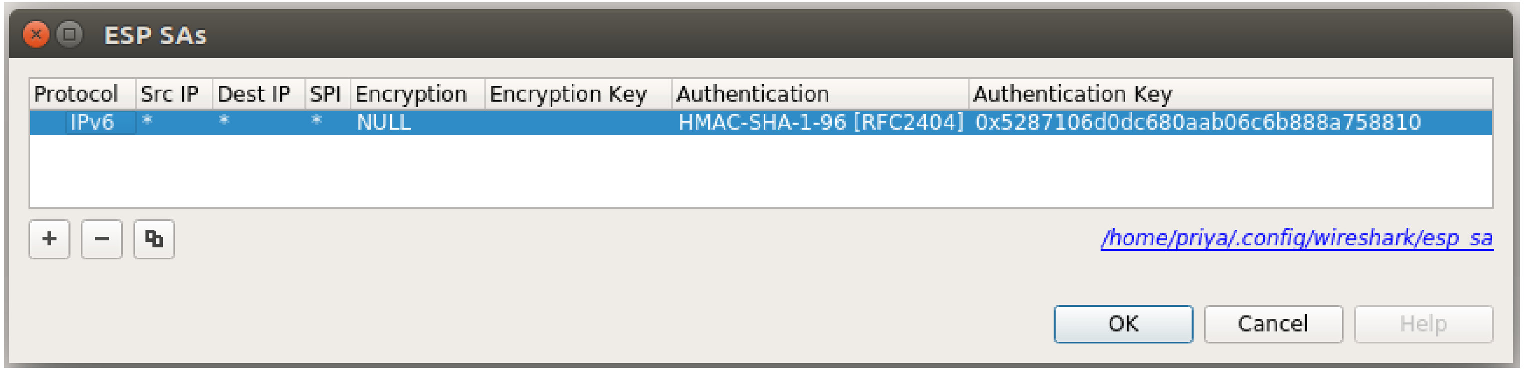
Are the keys used in ESP?



Failed authentication

```
▸ Frame 11: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
▸ Linux cooked capture
▸ Internet Protocol Version 6, Src: 2a01:59f:89a1:af67:2:3:f992:90bf, Dst: 2a01:598:401:3002::4
▾ Encapsulating Security Payload
    ESP SPI: 0xf5f9672e (4126762798)
    ESP Sequence: 1
  ▸ Data (44 bytes)
  ▾ Authentication Data [incorrect, should be 0x102DC16067AB36900D86827A]
      [Good: False]
      [Bad: True]
```

Set up SA with obtained IK



Success: Key validation

```
▸ Frame 12: 108 bytes on wire (864 bits), 108 bytes captured (864 bits)
▸ Linux cooked capture
▸ Internet Protocol Version 6, Src: 2a01:598:401:3002::4, Dst: 2a01:59f:89a1:af67:2:3:f992:90bf
▾ Encapsulating Security Payload
    ESP SPI: 0x00001c17 (7191)
    ESP Sequence: 1
  ▸ Data (32 bytes)
  ▾ Authentication Data [correct]
      [Good: True]
      [Bad: False]
```

Summary: Testing UE

- Test1: Sniffing VoLTE/VoWiFi interfaces
  - Use case identification
  - Results: Information disclosures like **IMEI, IMSI, private IPs**.
- Test2: ISIM sniffing with SIMTrace
  - Result: IK/CK
  - Wireshark dissector for extraction
  - Validation using Wireshark Gcrypt with authentication check in ESP

ERNW
providing security.

Simple demo of replay attack of SIP INVITE in a hidden non-IPSec channel

Final Summary

- Current implementations of VoLTE/VoWiFi make use of IPSec
- 4 experimental attacks on OpenIMS without ipsec
- Sniffing on VoLTE/VoWiFi interfaces with ipsec
  - Information disclosures identified
- ISIM Sniffing with SIMTrace
- Wireshark dissector
  - Extracted CK/IK
  - Verified obtained IK with wireshark Gcrypt

Mitigation

- **Never rely on user end security**
- Traffic monitoring
  - In PDN gateways that performs deep packet inspection
  - Whitelist rules in place that determines the expected value in each SIP header field.
- Encryption
  - To protect against info disclosures

ERNW providing security.

```
##IPTABLES ON ANDROID TO ROUTE TRAFFIC TO LAPTOP AND BACK

iptables -F
iptables -t nat -F
echo 1 > /proc/sys/net/ipv4/ip_forward
RMNET=`ip addr show dev rmnet1 |grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}"`
WLAN=`ip addr show dev wlan0 | grep inet | grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" | grep -v 255`
IMS="10.0.0.1"
MITM="192.168.0.2"
iptables -t nat -A OUTPUT -d $IMS -j DNAT  --to-destination $MITM
iptables -t nat -A POSTROUTING -o wlan0 -d $MITM   -j SNAT --to-source $WLAN
iptables -t nat -A POSTROUTING -o rmnet1 -s $MITM  -d $IMS -j SNAT --to-source $RMNET
iptables -t nat -L -vn
```

ERNW
providing security.

Questions?

White paper:
https://www.ernw.de/download/newsletter/ERNW_Whitepaper_60_Practical_Attacks_On_VoLTE_And_VoWiFi_v1.0.pdf

Thanks to Hendrik, my mentor.

schalakkal@ernw.de

@priyachalakkal

www.ernw.de

www.insinuator.net