

OmniTWAG 無線LAN接続

Omnitouch 無線LAN接続

無線LAN接続は、OmniTWAG 無線LAN接続

無線LAN接続

- 無線LAN接続
- 無線LAN WiFi 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN 2.0 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続

無線LAN接続

OmniTWAG無線LAN WiFi 無線LAN接続は、3GPP TWAG 無線LAN接続は、無線LAN接続は、無線LAN WiFi 無線LAN接続は、SIM 無線LAN接続

TWAG 無線LAN EAP-AKA無線LAN接続 - 無線LAN接続は、SIM 無線LAN WiFi 無線LAN接続は、無線LAN WiFi 無線LAN接続は、WiFi 無線LAN接続は、WiFi 無線LAN接続

無線LAN接続

無線LAN接続

- 電話番号を SIM に登録
- 電話番号を電話番号帳に登録
- 電話番号帳を WiFi/WPA2
- 電話番号を SIM に登録

□□□□□□□□

- 00000000000000000000
- 00000000000000000000
- 00000000WiFi 0000000000
- 000000WiFi 000000000000
- 0000000WiFi 000000000 IMSI
- 0000000000000000 WiFi 000000

0000/000

- 00000000000000000000
- 00000000000000000000
- 0000000000 WiFi 00

WiFi

WiFi WiFi

TWAG ☐ ☐ ☐ ☐ ☐ ☐

TWAG □□□□□□□□

- **WiFi** 無線 LAN RADIUS 接続
- 無線 LAN HSS/HLR 接続 Diameter SWx 接続

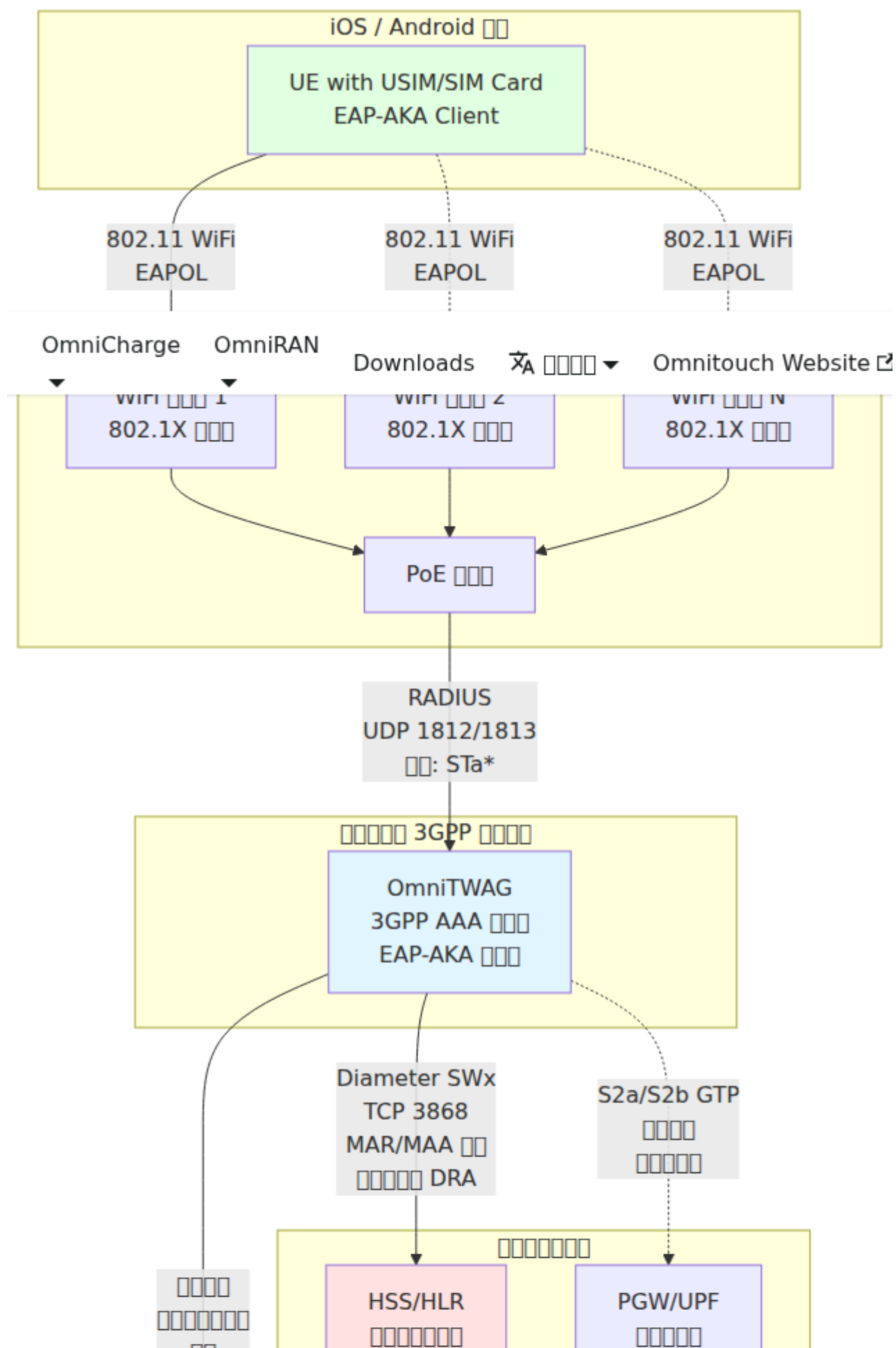
WiFi AP

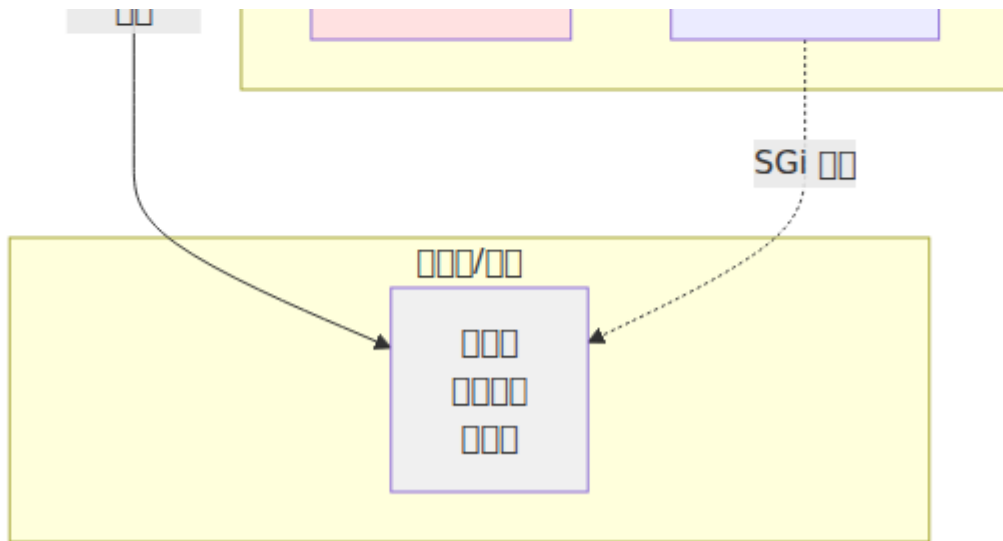
1. IMSI SIM
2. WiFi AP RADIUS TWAG

3. TWAG 00000 HSS 0000000000
 4. 000 TWAG 0000 EAP-AKA 00-0000
 5. 00000000000 WiFi 0000
 6. 00000000000000000000000000
-

□□□□

□□□□





네트워크

- **STa***: WiFi AP ↔ TWAG ↔ RADIUS/Diameter ↔ 3GPP ↔ AAA
- **SWx**: TWAG ↔ 3GPP AAA ↔ HSS ↔ Diameter
- **S2a/S2b**: ↔ GTP ↔
- **SGi**: ↔
- **802.11**: WiFi ↔
- **EAPOL**: EAP over LAN ↔ 802.1X ↔

네트워크

1

- 國際標準化組織 DRA
- 國際標準
- 國際標準化組織

```

graph LR
    OTWAG[OmniTWAG  
3GPP AAA] --- SWx[SWx  
TCP 3868] --- ODRA[OmniDRA  
Diameter]
    ODRA -- "5054" --- OHSS1[OmniHSS #1  
A]
    ODRA -- "5054" --- OHSS2[OmniHSS #2  
B]
    ODRA -- "5054" --- OHSS3[OmniHSS #3  
]
  
```

103

- 〇〇〇〇〇〇
- 〇〇〇 HSS 〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇 HSS〇
- 〇〇〇〇〇〇〇
- 〇〇〇〇

TWAG OCS Diameter Gy

WiFi AP RADIUS TWAG Gy
DRA/OCS

OmniTWAG 3GPP AAA 模块

模块

OmniTWAG 模块

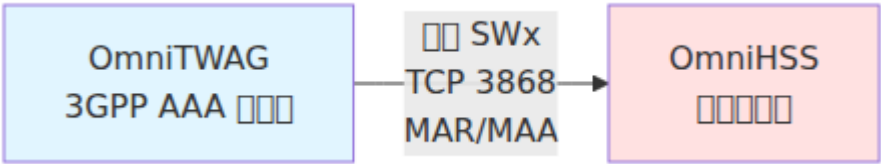
1. 模块

模块

模块

- 模块/模块 WiFi 模块
- 模块/模块
- 模块RADIUS 模块

模块



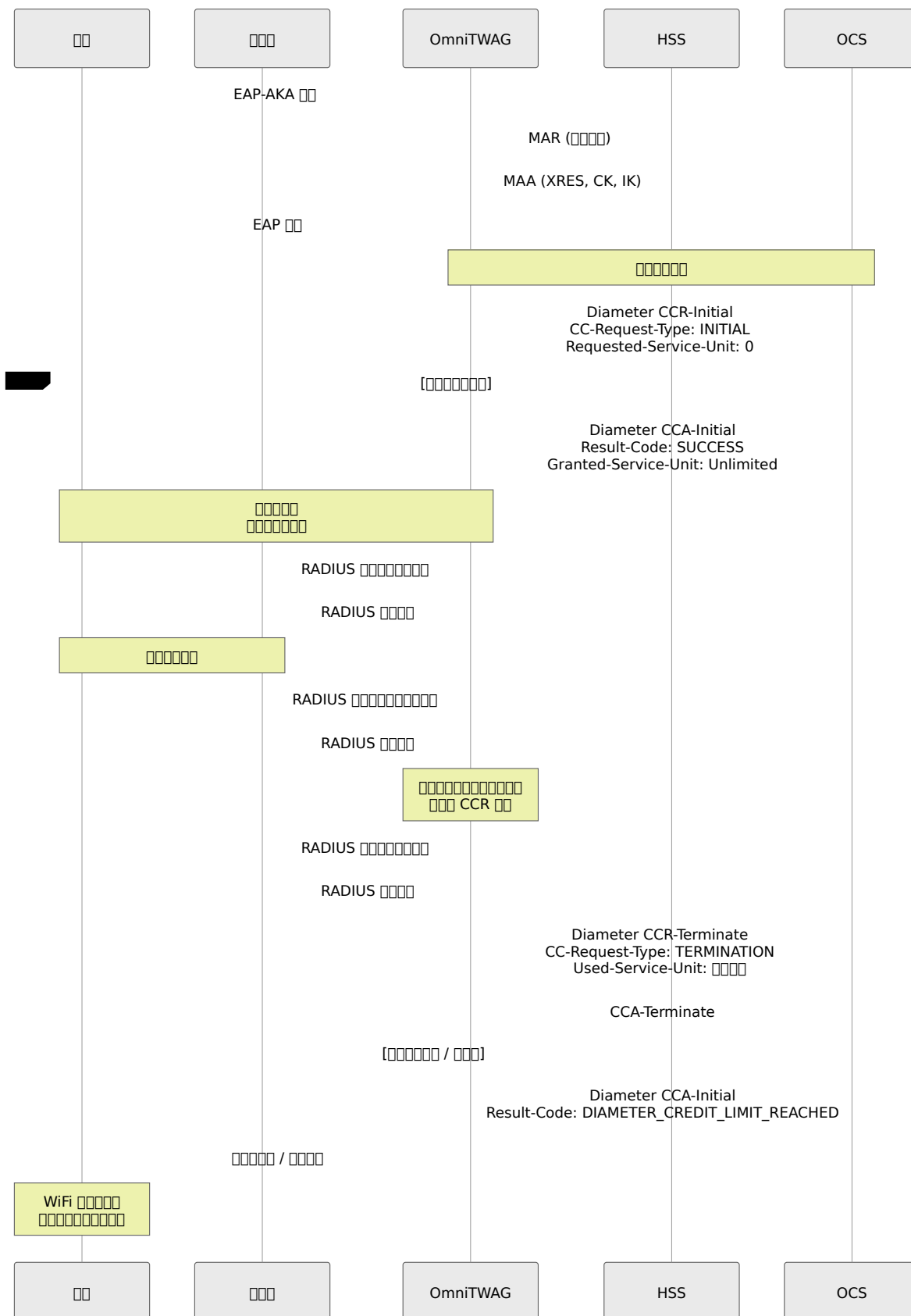
2. 模块

模块 WiFi 模块 OCS 模块 CCR-Initial 模块

模块

- 000000000000/00
- 00000000 WiFi 00
- 00000000000000
- 00 WiFi 0000000000/000000

000



- CCR-I CCR-T OCS
- CCR

- 3GPP 3GPP
- 3GPP 3GPP

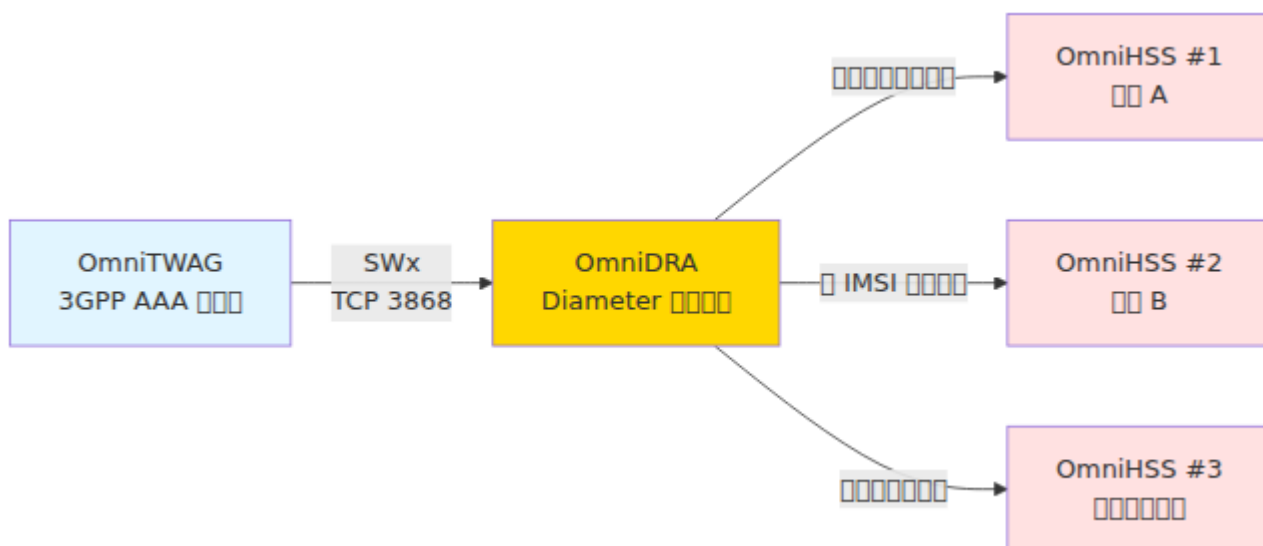
3. 3GPP Gy 3GPP

3GPP 3GPP WiFi 3GPP 3GPP 3GPP OCS 3GPP 3GPP

3GPP

- 3GPP
- 3GPP WiFi
- 3GPP 10GB 3GPP
- 3GPP

3GPP

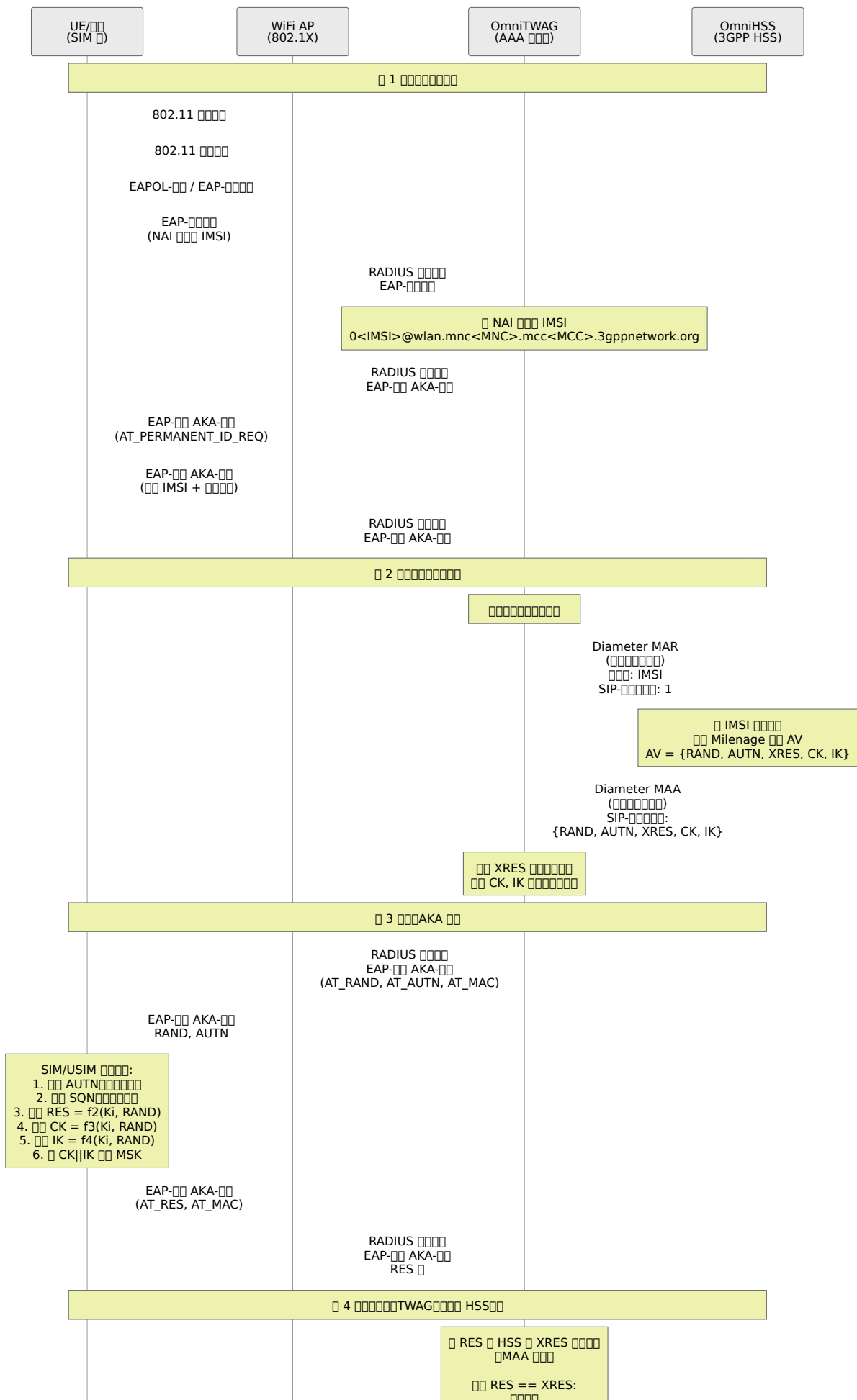


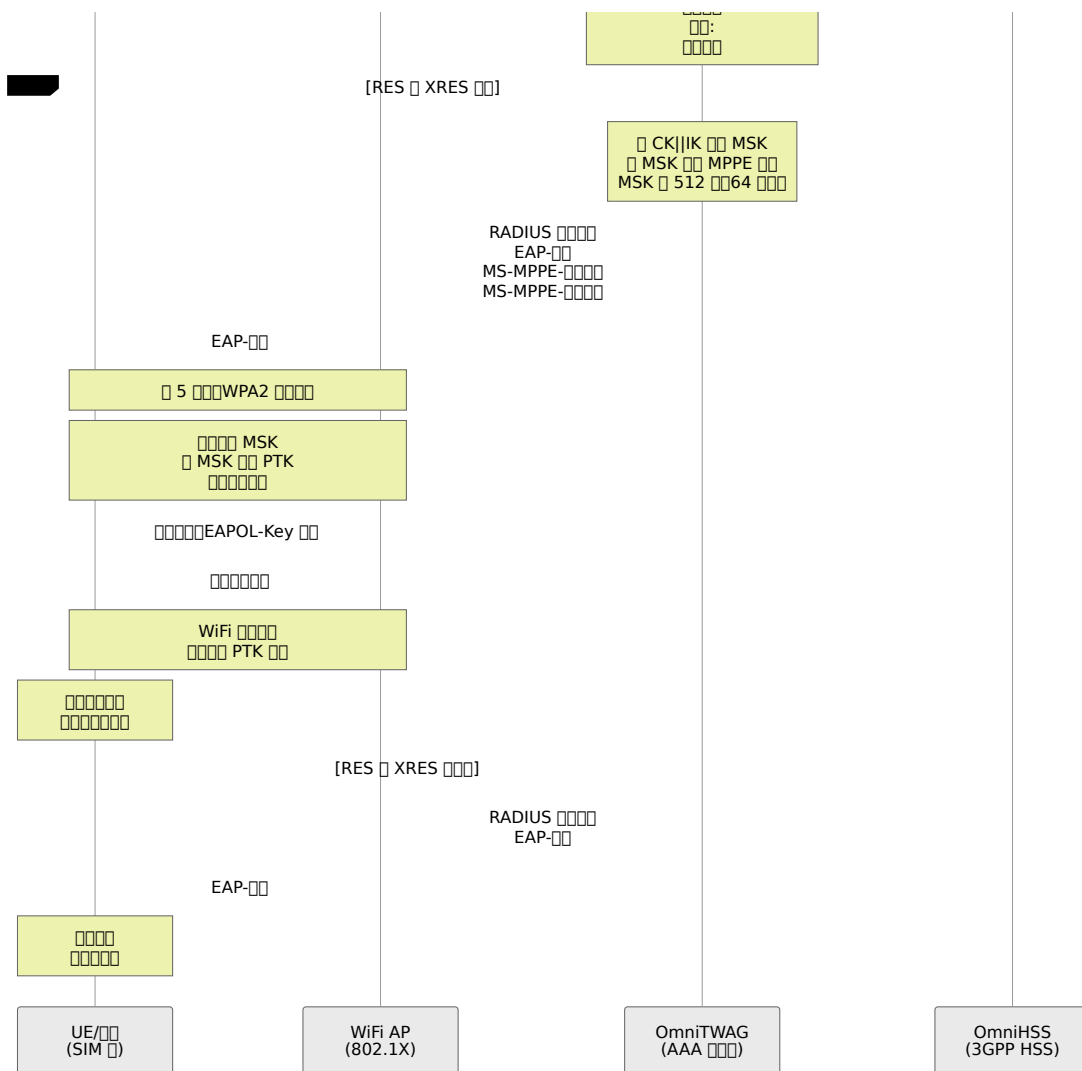
3GPP

- 3GPP CCR-I 3GPP CCR-U 3GPP CCR-T 3GPP OCS
- 3GPP 10MB 50MB 100MB
- 3GPP 80% 3GPP CCR
- 3GPP
- 3GPP
- 3GPP

□□□□

□□□ **EAP-AKA** □□□□





□□□□□□□□□□□□□□□□

1. **MAR/MAA** □ **HSS** □□□□□□□□□□□□ XRES □ MAA□□□□□□□□□□TWAG □□□□□□□□□□
2. **TWAG** □□ **RES** □□□HSS □□□□□□□□□□XRES□□□ TWAG □□□ UE □□□ RES □□□□□HSS □□□□□□□□□□
3. □□□□□□ **TWAG**□□□□□□□□□□□□ HSS □□□□□□□□□□□□——□□□□□ 3GPP □□□□□AAA □□□□TWAG□□□□□□□□□□

□□□□□

□□□ NAI □□□□□□□□□□□□□□□□IMSI□□□

50557000000000000001@wlan.mnc057.mcc505.3gppnetwork.org

0<IMSI>@wlan.mnc<MNC>.mcc<MCC>.3gppnetwork.org

IMSI 0 IMSI SIM

MSK

MSK EAP-AKA 512 64 WiFi

MSK

1. UE TWAG MSK
2. UE SIM CK/IK
3. TWAG HSS CK/IK
4. $MSK = PRF'(CK || IK, \text{"Full Authentication", IMSI, ...})$

MSK

1. **PMK** $PMK = MSK$ 256 32
2. **WPA2** UE AP PMK PTK
3. WiFi PTK TK

MSK

- MSK WiFi
- WiFi
- EAP WiFi
- MSK
- MSK MSK

SQL

1. AUTS -
2. EAP-AKA AT-AUTS
3. TWAG AUTS HSS
4. HSS
- 5.

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

TWAG `config/` Elixir `config/runtime.exs`

[illegible]

Diameter

```
config :diameter_ex
```

```

config :diameter_ex,
  diameter: %{
    # Diameter 配置
    service_name: :omnitouch_twag,

    # 监听 Diameter 请求 IP 地址
    listen_ip: "10.5.198.200",

    # Diameter 监听端口 3868
    listen_port: 3868,

    # Diameter Origin-Host
    host: "omnitwag",

    # Diameter Origin-Realm
    realm: "epc.mnc057.mcc505.3gppnetwork.org",

    # Diameter HSS/DRA AAA 配置
    peers: [
      %{
        # Diameter Origin-Host
        host: "omni-hss01.epc.mnc057.mcc505.3gppnetwork.org",

        # Diameter Origin-Realm
        realm: "epc.mnc057.mcc505.3gppnetwork.org",

        # IP 地址 HSS/DRA
        ip: "10.179.2.140",

        # 监听端口 3868
        port: 3868,

        # TLS 配置
        tls: false,

        # 传输方式: diameter_tcp 或 :diameter_sctp
        transport: :diameter_tcp,

        # 是否主动建立连接 true/false
        initiate_connection: true
      }
    ]
  }

```

```
]
}
```

3GPP TS 23.003

```
epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

- MNC = 057
- MCC = 505

DRA OmniDRA IP DRA HSS DRA
IMSI HSS

RADIUS

config :omnitwag

```

config :omnitwag,
  radius_config: %{
    # RADIUS server IP address
    # port = radius server port
    allowed_source_subnets: ["10.7.15.0/24", "192.168.1.0/24"],

    # RADIUS secret
    # AP secret
    secret: "YOUR_STRONG_SECRET_HERE"
  }

```

Configuration

- RADIUS server IP address
- `allowed_source_subnets` AP IP
- RADIUS secret 1812/1813

Configuration

```
allowed_source_subnets: ["10.7.15.0/24", "192.168.1.0/24"]
```

Configuration/Secret

Prometheus

config :omnitwag

```
config :omnitwag,
  prometheus: %{
    # Prometheus
    port: 9568
  }
```

http://<twag-ip>:9568/metrics

1812	UDP	RADIUS
1813	UDP	RADIUS
3868	TCP	Diameter (SWx HSS/DRA)
443	TCP	HTTPS Web
8444	TCP	HTTPS REST API
9568	TCP	Prometheus

OmniTWAG WiFi AP

- **WPA2-802.1X** 配置
- **RADIUS** 配置
- **EAP-AKA** 配置

支持的Cisco Aironet、Aruba、Ubiquiti UniFi、Ruckus、hostapd 的 AP

配置 AP 配置

1. **WPA2-802.1X** 配置
2. **RADIUS** 配置 TWAG IP 地址
3. **RADIUS** 配置 1812
4. **RADIUS** 配置 1813 配置
5. **RADIUS** 配置 TWAG 配置
6. **EAP** 配置 EAP-AKA 配置

Cisco AP 配置

CLI 配置

```
! 配置 RADIUS 配置
radius-server host 10.5.198.200 auth-port 1812 acct-port 1813 key
YOUR_SHARED_SECRET

! 配置 SSID 配置 802.1X
dot11 ssid OPERATOR-WIFI
    vlan 10
    authentication open eap eap_methods
    authentication network-eap eap_methods
    authentication key-management wpa version 2

! 配置 SSID 配置
interface Dot11Radio0
    encryption mode ciphers aes-ccm
    ssid OPERATOR-WIFI
```

Web 配置

1. 配置 配置 → **AAA** → **RADIUS** 配置

2. 配置 RADIUS 服务器 `10.5.198.200:1812` 连接
3. 配置 **WLAN** 服务
4. 配置安全协议 **WPA2-PSK**
5. 配置 EAP 认证协议 **EAP-AKA** 认证
6. 配置 RADIUS 认证

hostapd 配置

配置 Linux 上的 AP 服务 OpenWrt 配置

```
# /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=OPERATOR-WIFI

# WPA2-
wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=CCMP
ieee8021x=1

# RADIUS 
auth_server_addr=10.5.198.200
auth_server_port=1812
auth_server_shared_secret=YOUR_SHARED_SECRET

acct_server_addr=10.5.198.200
acct_server_port=1813
acct_server_shared_secret=YOUR_SHARED_SECRET

# EAP 
eap_server=0

# 2.0 - 
interworking=1
internet=1
anqp_3gpp_cell_net=505,057
domain_name=wlan.mnc057.mcc505.3gppnetwork.org
nai_realm=0,wlan.mnc057.mcc505.3gppnetwork.org,0,21[2:1][5:7]
roaming_consortium=505057
hs20=1
```


- 802.11n 802.11ac 802.11ad
- 802.11n 802.11ac 802.11ad
- 802.11n 802.11ac 802.11ad
- 802.11n 802.11ac 802.11ad
- 802.11n 802.11ac 802.11ad

2.0 AP

AP

1. **802.11u** ANQP
2. **WPA2**-802.1X
3. **EAP-AKA** EAP-AKA
4. **ANQP**

hostapd AP

```
# 2.0 / Passpoint
interworking=1
internet=1
asra=0
esr=0
uesa=0

# ANQP
anqp_3gpp_cell_net=505,057
domain_name=omnitouchns.com,wlan.mnc057.mcc505.3gppnetwork.org

# NAI
nai_realm=0,wlan.mnc057.mcc505.3gppnetwork.org,0,21[2:1][5:7]
# <>,<>,<eap->[auth-id:auth-val]
# 21 = EAP-AKA
# 2:1 = SIM
# 5:7 = EAP EAP-AKA

# 
roaming_consortium=505057
# MCC=505MNC=057

# 
venue_group=1
venue_type=8
venue_name=eng: WiFi

# WPA2-
wpa=2
wpa_key_mgmt=WPA-EAP
rsn_pairwise=CCMP
ieee8021x=1

# RADIUS OmniTWAG
auth_server_addr=10.5.198.200
auth_server_port=1812
auth_server_shared_secret=YOUR_SHARED_SECRET

acct_server_addr=10.5.198.200
acct_server_port=1813
acct_server_shared_secret=YOUR_SHARED_SECRET

# SSID
```

```
ssid=OperatorWiFi
utf8_ssid=1

# 2.0
hs20=1
hs20_oper_friendly_name=eng: WiFi
```

WiFi

WiFi

1. Passpoint WiFi
2. AP ANQP
3. ANQP MCC/MNC
 -
4. \geq $>$
 - EAP-AKA
5. $>$
 - WiFi
- 6.

WiFi

1. MCC/MNC
- 2.
3. WPA2-PSK
- 4.
5. WiFi

WiFi

Web

https://<twag-ip>/

📄

- **RADIUS** 📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡
- 📡📡📡📡📡📡 AP📡📡📡📡📡📡 SSID 📡📡
- 📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡
- **Diameter** 📡📡📡📡📡 HSS/DRA 📡📡📡

Prometheus 📄

📄 Prometheus 📄 TWAG 📄📄

```
# prometheus.yml
scrape_configs:
  - job_name: 'omnitwag'
    static_configs:
      - targets: ['10.5.198.200:9568']
    metrics_path: '/metrics'
    scrape_interval: 15s
```

📄📄📄📄

RADIUS 📄📄📄📄

- `radius_access_request_count` - 📡📡📡 RADIUS 📡📡📡📡📡
- `radius_access_accept_count` - 📡📡📡📡📡📡📡
- `radius_access_reject_count` - 📡📡📡📡📡📡📡
- `radius_access_challenge_count` - 📡📡📡📡📡📡📡
- `radius_accounting_request_count{status_type}` - 📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡
- `radius_active_clients_count` - 📡📡📡📡📡📡📡 5 📡📡📡📡📡
- `radius_access_points_count` - 📡📡📡📡📡 5 📡📡📡📡📡

EAP-AKA 📄📄📄📄

- `eap_aka_identity_count` - EAP-AKA 📡📡📡
- `eap_aka_challenge_count` - EAP-AKA 📡📡📡
- `eap_aka_sync_failure_count` - 📡📡📡📡📡 SQN 📡📡📡📡📡

- `eap_aka_auth_success_count` - 成功回数
- `eap_aka_auth_reject_count` - 失敗回数

Diameter 監視項目

- `diameter_message_count{application, command, direction}` - Diameter メッセージの送信/受信回数

Erlang VM 監視項目

- `vm_memory_total` - VM 総メモリ使用量
- `vm_memory_processes` - Erlang プロセスのメモリ使用量
- `vm_memory_processes_used` - Erlang プロセスが実際に使用しているメモリ量
- `vm_memory_system` - Erlang システムのメモリ使用量
- `vm_memory_atom` - アトムメモリ使用量
- `vm_memory_atom_used` - アトムメモリが実際に使用している量
- `vm_memory_binary` - バイナリメモリ使用量
- `vm_memory_code` - コードメモリ使用量
- `vm_memory_ets` - ETS テーブルのメモリ使用量

Erlang VM 監視項目

- `vm_system_info_process_count` - Erlang プロセス数
- `vm_system_info_port_count` - ポート数
- `vm_system_info_atom_count` - アトム数
- `vm_system_info_schedulers` - スcheduler数
- `vm_system_info_schedulers_online` - オンラインの scheduler 数

Erlang VM 監視項目

- `vm_statistics_run_queue` - 実行キューの長さ
- `vm_total_run_queue_lengths_total` - 全実行キューの長さの総和
- `vm_total_run_queue_lengths_cpu` - CPU 実行キューの長さの総和
- `vm_total_run_queue_lengths_io` - IO 実行キューの長さの総和

監視項目

- RADIUS 向 EAP-AKA 认证
- 认证失败后重试 5 次
- VM 每秒 5 个 Erlang 认证
- 部署 Prometheus 监控 `http://<twag-ip>:9568/metrics` 查看

部署

TWAG 使用 Elixir 和 Logger 记录日志

配置 systemd

```
# 启动 twag
journalctl -u twag -f

# 查看 100 条日志
journalctl -u twag -n 100

# 后台运行 twag
journalctl -u twag -b

# 查看指定时间段的日志
journalctl -u twag --since "2025-10-12 10:00:00" --until "2025-10-12 11:00:00"
```

配置 twag

- `RADIUS 认证端口 1812` - 认证端口
 - `向 {IP}: 认证 - 向 AP 向 RADIUS 认证`
 - `向 1 认证 - 向 EAP 认证`
 - `向 2 认证AKA 认证 - 认证`
 - `认证 - 认证`
 - `认证 - 认证`
 - `认证 AP: {IP} - 认证 AP`
-

配置

配置

配置 WiFi

配置

1. 配置 TWAG 配置 `journalctl -u twag -f`
2. 配置 AP 配置 TWAG 配置 RADIUS 配置
3. 配置 RADIUS 配置 TWAG `tcpdump -i eth0 port 1812`
4. 配置 HSS/配置

配置

- RADIUS 配置
- 配置 UDP 1812/1813
- RES/XRES 配置 SIM Ki 配置 HSS 配置
- 配置 SQL 配置
- AP 配置 TWAG 配置

Diameter 配置

配置 Diameter 配置 HSS/DRA

配置

1. 配置 `telnet <hss-ip> 3868`
2. 配置 Diameter 配置 Origin-Host 配置 Origin-Realm 配置 IP
3. 配置 HSS/DRA 配置
4. 配置 TCP 3868

配置

- 配置 IP/配置
- 配置 TCP 3868
- Origin-Host/Realm 配置

- HSS/DRA 和 TWAG 的

和

和 >5 的

和

1. 和 HSS 的
2. 和 ping <hss-ip> mtr <hss-ip>
3. 和 TWAG 的 top htop
4. 和 Diameter 的

和

- HSS 的
- 和
- TWAG 的 CPU/和
- 和

和

和

```
# 和 RADIUS 的
tcpdump -i eth0 -n port 1812 or port 1813 -w radius.pcap

# 和 Diameter 的
tcpdump -i eth0 -n port 3868 -w diameter.pcap

# 和 AP 的
tcpdump -i eth0 -n host 10.7.15.72 and port 1812 -w radius-ap1.pcap
```

和 Wireshark 和 RADIUS 和 Diameter 的

和

TWAG

```
# shell TWAG
iex --sname debug --remsh twag@hostname --cookie <cookie>
```

IEx

```
# 
CryptoState.keys()

# 
CryptoState.get("0505338057900001867@wlan.mnc057.mcc505.3gppnetwork.c

# AP
APState.list()

# 
ClientUsage.list()
```

AAA

AAA	AAA	AAA
AAA	AAA	AAA AP TWAG AAA RADIUS AAA
RES AAA <XRES>AAA <RES>	AAA	AAA SIM Ki AAA HSS AAA
Diameter AAA	AAA HSS	AAA AAA HSS AAA
AAA EAP AAA	AAA EAP AAA AAA	AAA AP AAA AAA AP
AAA EAP-AKA AAA	AAA EAP-AKA AAA AAA	AAA EAP-AKA AAA
AAA AAA	SQL AAA	AAA AAA

AAA

OmniTWAG AAA 3GPP IETF AAA

- **3GPP TS 23.402** AAA 3GPP AAA
- **3GPP TS 24.302** AAA 3GPP AAA EPC
- **3GPP TS 29.273** AAA Diameter SWx/SWm AAA
- **3GPP TS 33.402** AAA 3GPP AAA
- **3GPP TS 35.206** AAA Milenage AAA
- **RFC 2865** AAA RADIUS AAA
- **RFC 2866** AAA RADIUS AAA
- **RFC 3579** AAA RADIUS AAA EAP AAA
- **RFC 4187** AAA EAP-AKA AAA
- **RFC 5448** AAA EAP-AKA' AAA



OmniTWAG 與 Omnitouch 整合 3GPP WiFi 認證

1. 認證流程
2. 3GPP SWx EAP-AKA RADIUS 認證
3. SIM 認證
4. MSK WPA2 認證
5. 2.0 認證
6. 認證
7. HSS OmniDRA 認證/認證

2.0 2025 OmniTWAG - WiFi 認證 © 2025 Omnitouch
???