The 5G Titanic

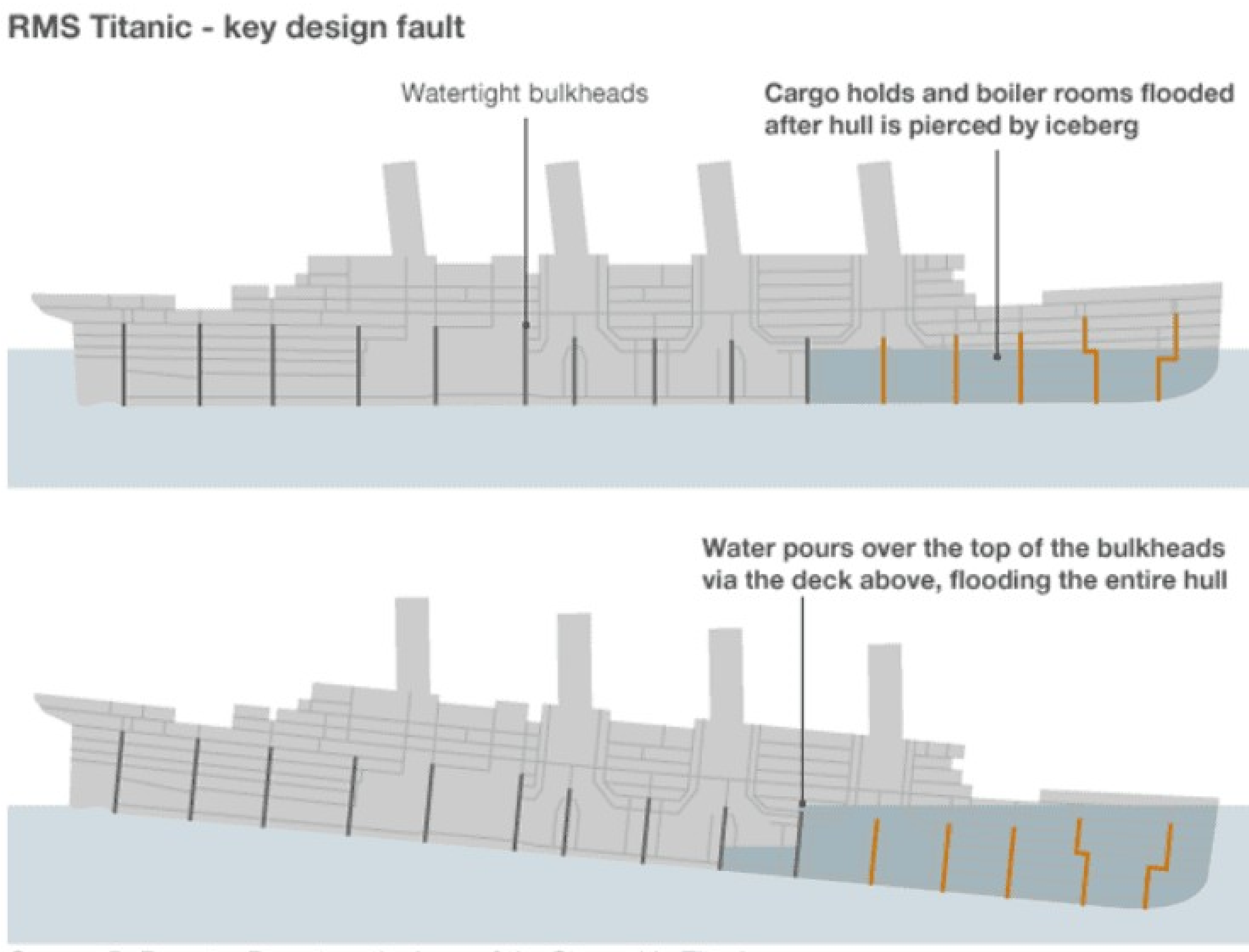
Dr. Altaf Shaik,
Robert Jaschek

Fast IOT &
Technische Universität Berlin

Reference: https://education.nationalgeographic.org/resource/titanic-sinks/

Titanic

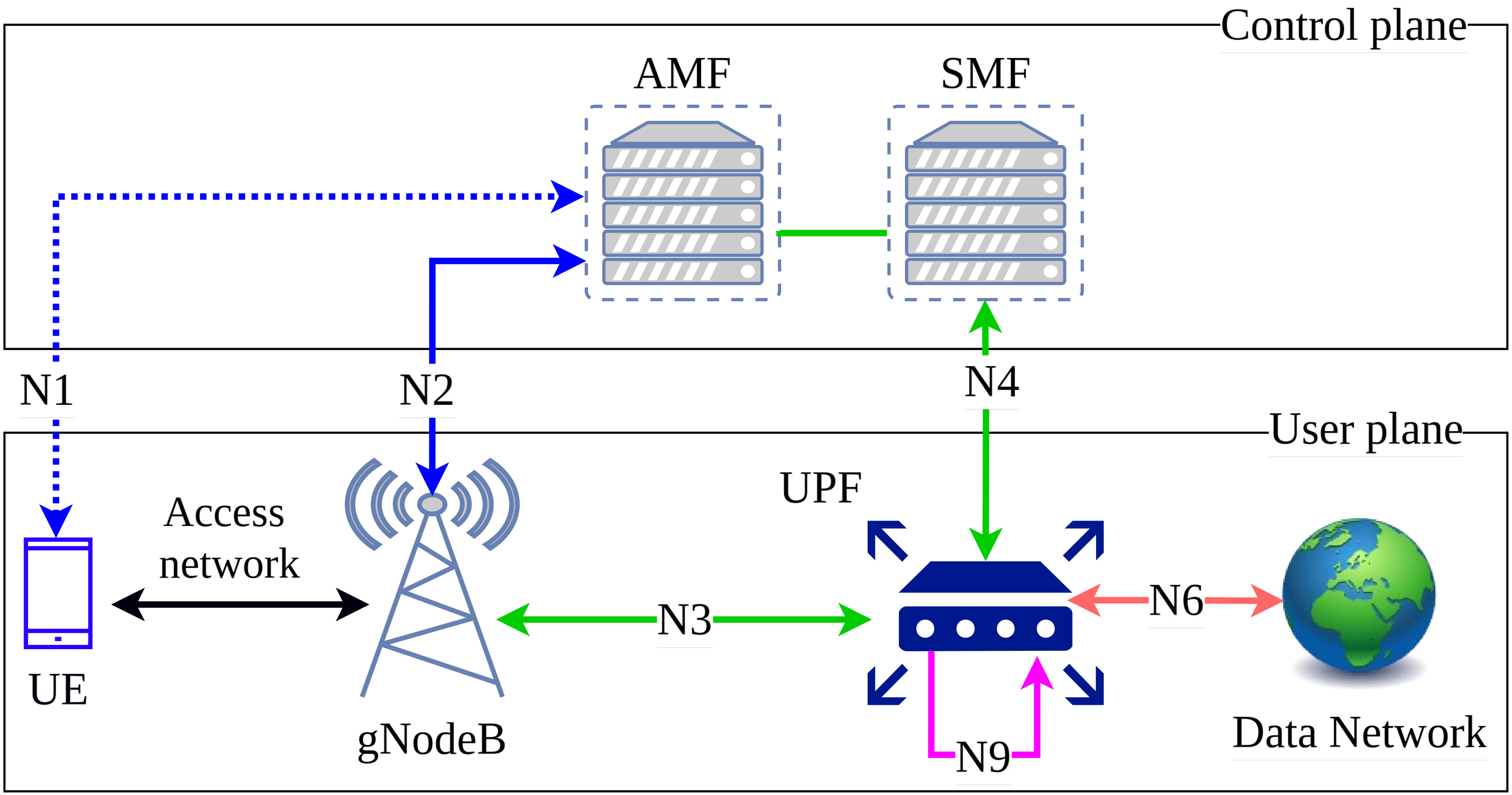
On April 15, 1912, the RMS Titanic sunk in the North Atlantic Ocean



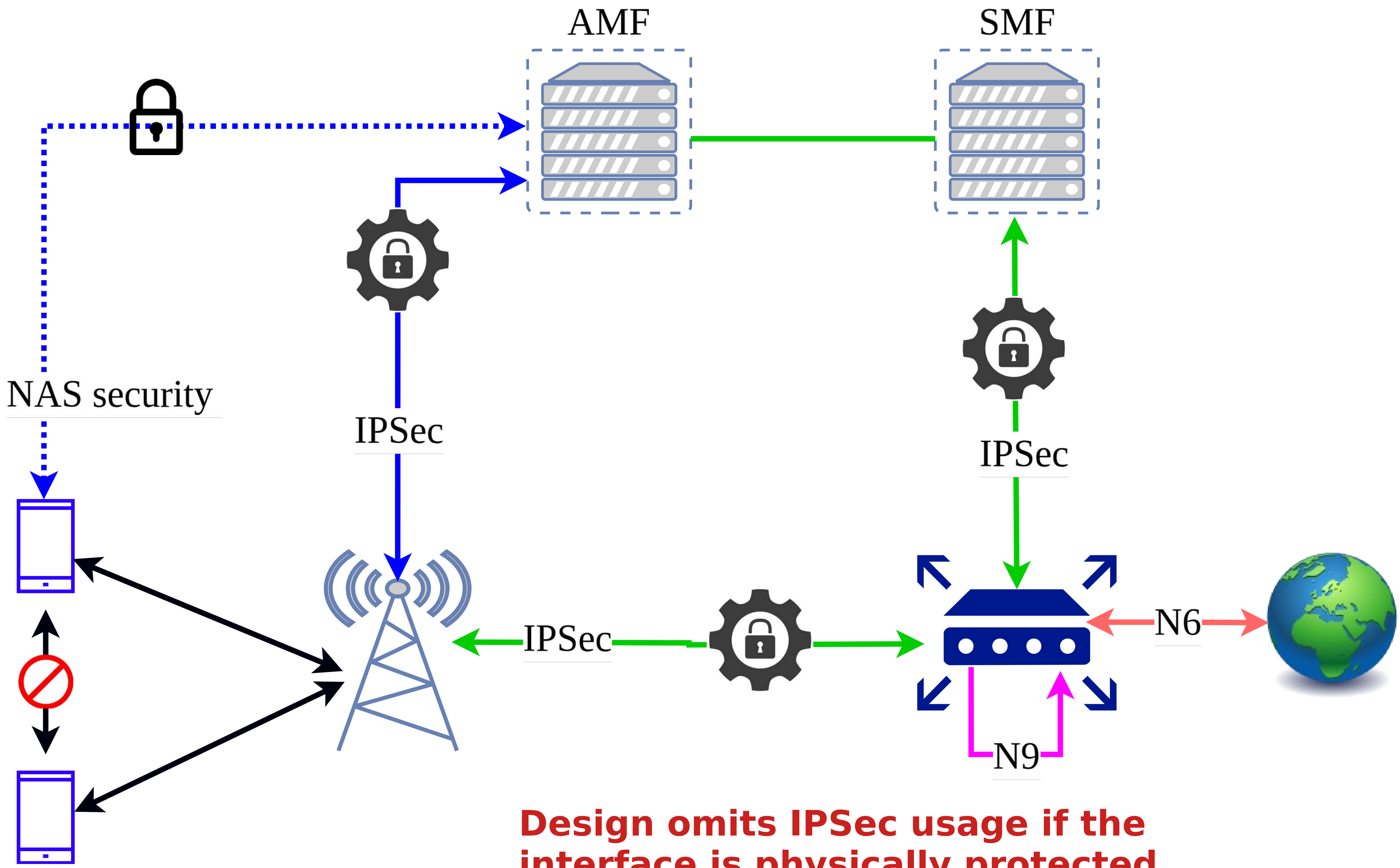
What 5G assumes?

CUPS

Control user plane separation



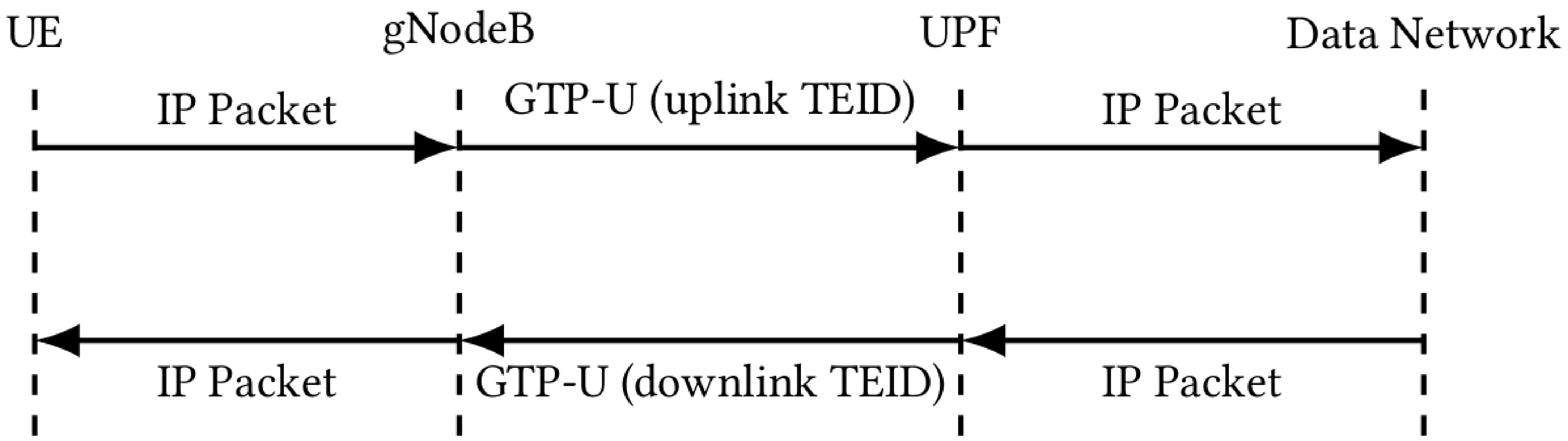
Security features



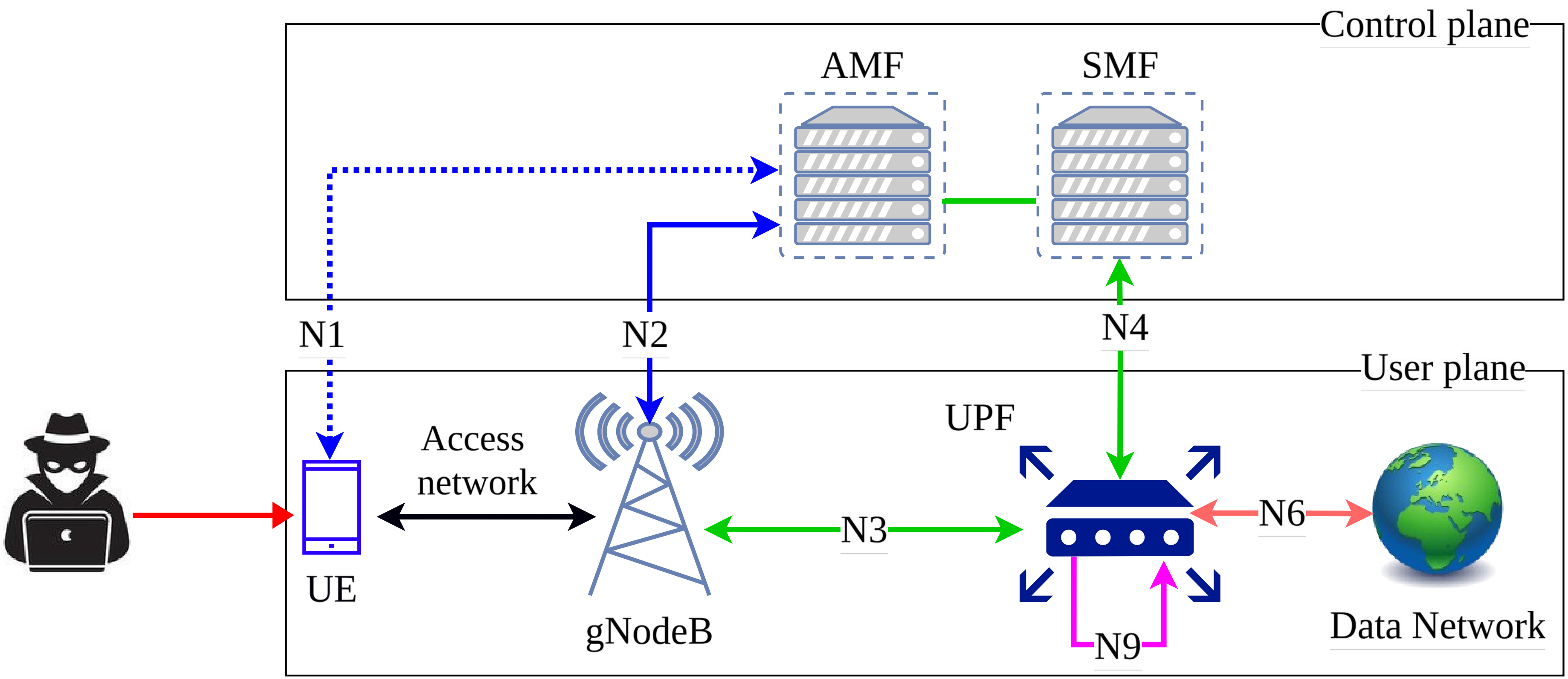
Design omits IPSec usage if the interface is physically protected.

5G data flow

GTP: GPRS tunneling protocol (Age: 26)



Positioning the 5G attacker



But what if that separation fails?

Protocol tunneling via GTP-U

- Encapsulating one protocol inside user-plane traffic to reach a specific node
- Why GTP-U: A protocol that lacks built-in integrity checks or source authentication.
- Simple forwarding logic based solely on IP address and identifiers
  - No inspection of payload contents
- Delivers encapsulated inner payloads to internal GTP-U-capable nodes (e.g., UPF, gNodeB)
- **Sending GTP-U encapsulated packets to networks is considered fraud**

Protocol tunneling - packet

- **GTP-U-in-GTP-U** encapsulated packet
  - Standard protocol compliant

General **GTP-U-in-GTP-U** encapculated packet structure

src	dst	src	dst	TEID	src	dst	src	dst	TEID	src	dst	
IP		UDP		GTP	IP		UDP		GTP	IP		
Outer GTPH					Inner GTPH					Payload		

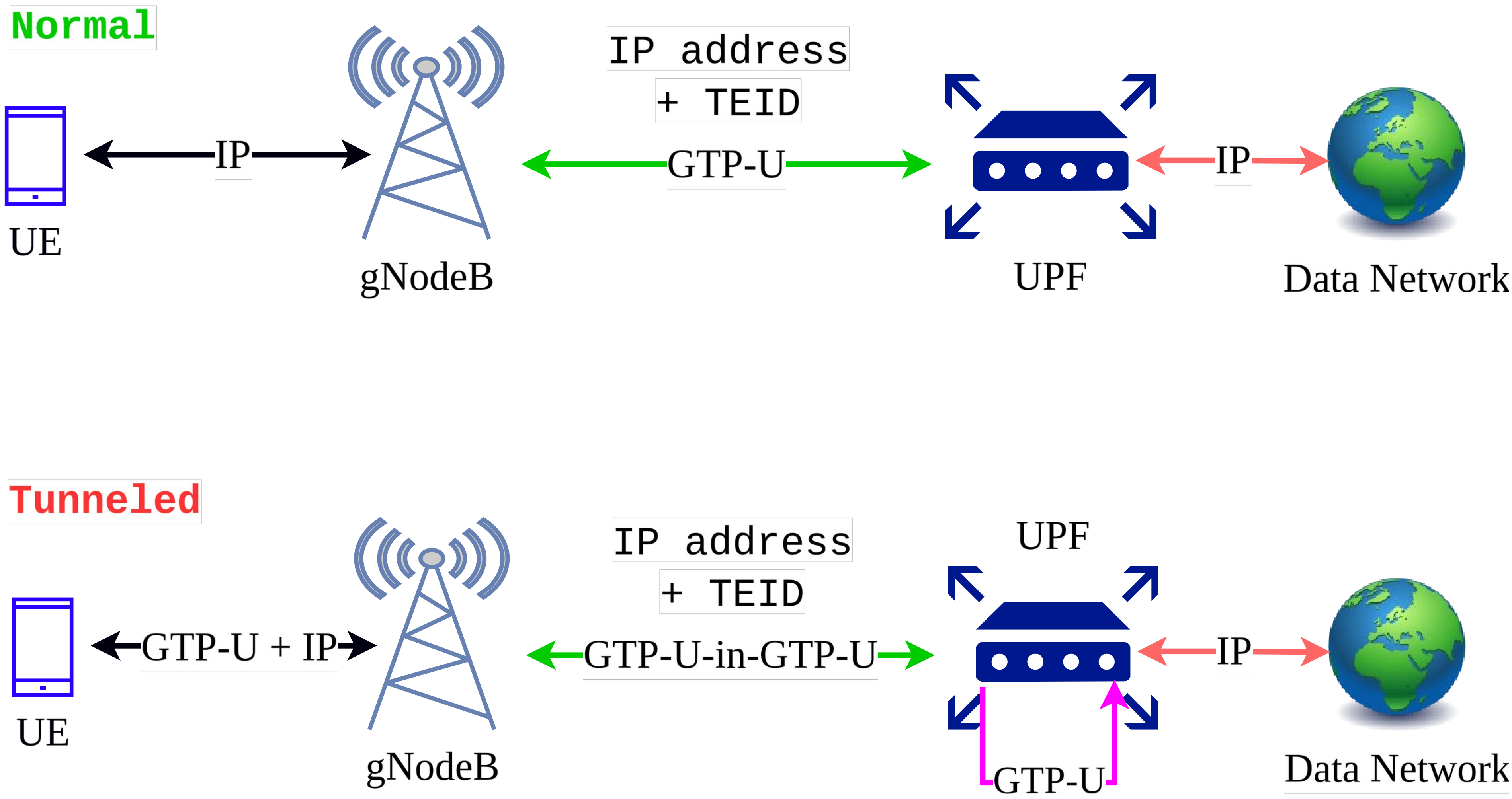
How to craft

- Discover and craft packet with internal IP addresses and ports
  - from search engines, recon, insiders, intermediaries
- Enumerate and forge target users tunnel identifier, and IP address

General **GTP-U-in-GTP-U** encapculated packet structure

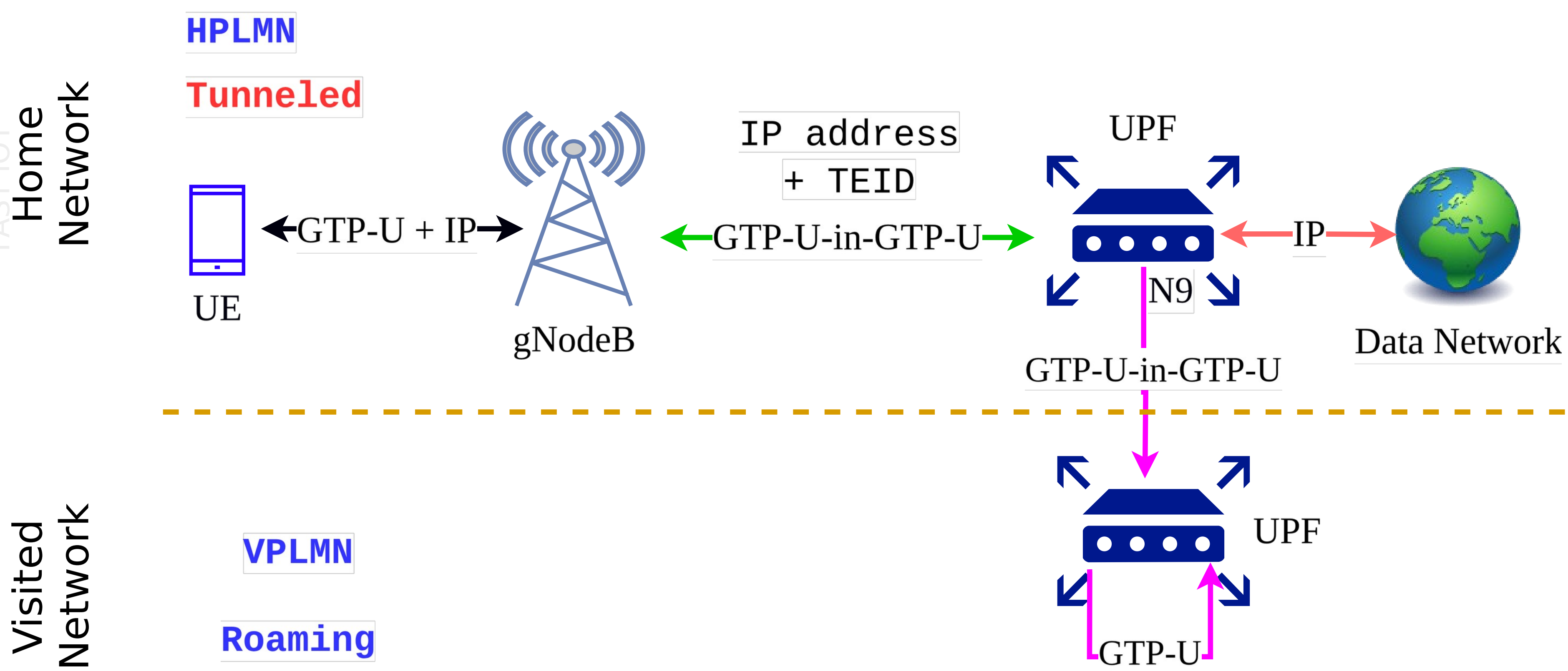
<table>
<tr><td>src</td><td>dst</td><td>src</td><td>dst</td><td>TEID</td><td>src</td><td>dst</td><td>src</td><td>dst</td><td>TEID</td><td>src</td><td>dst</td><td></td></tr>
<tr><td colspan="2">IP</td><td colspan="2">UDP</td><td>GTP</td><td colspan="2">IP</td><td colspan="2">UDP</td><td>GTP</td><td colspan="2">IP</td><td></td></tr>
<tr><td colspan="5">Outer GTPH</td><td colspan="5">Inner GTPH</td><td colspan="3">Payload</td></tr>
</table>

Protocol tunneling - flow



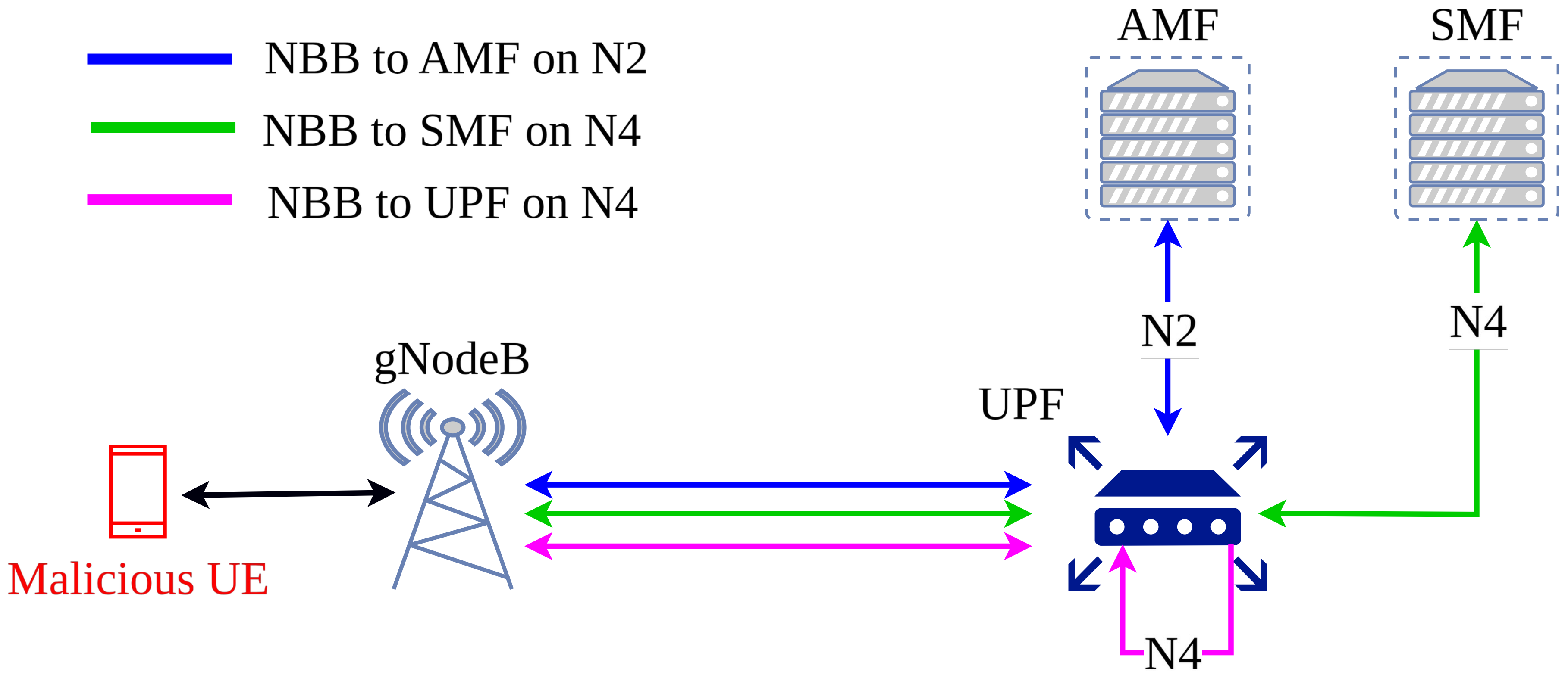
Protocol tunneling - roaming

- 5G has N9 interface – connect roaming interfaces
- Packet could be tunneled internationally – a vulnerable UPF will execute it



Network boundary bridging

- Routing user-plane traffic across architectural trust boundaries
  - Reach isolated control-plane NF like AMF, SMF
- Misconfigured routing and lack of egress filtering at UPF allow redirection to control-plane interfaces
- Target AMF (via NGAP) or, SMF & UPF (via PFCP)
  - Simple setup and association request messages to communicate



Trying it in the field

Setup

- **Six 5G Core networks**
  - 4 open source and 2 commercial (private)
  - isolated lab environments, containerized
  - Standard configurations, no custom firewalls
- One SDR based radio base station
  - From srsRAN project, connects to all cores
- Several 5G Smartphones and SIM cards
  - Sends encapsulated GTP-U packets to the UPF
  - protocol-compliant payloads such as ICMP, UDP, NGAP, PFCP
  - Fast automated enumeration of data plane identifiers IP, TEID, SEID
- Prior knowledge
  - Target UPF, AMF and SMF IP addresses

5G Core
Open5GS
free5GC
OAI-5G
SD-Core
PC1
PC2

What we found – vulnerabilities and vectors

Processing tunneled packets

- Outer GTP header gets correctly parsed
  - Sent under the attacker’s legit connection
- Inner GTP header is redirected to a target network element
  - **Tunnelled**: the malicious payload sent to UPF or gNodeB
  - **Bridged**: the malicious payload sent by AMF/SMF
- Payload can be processed or discarded – depends on guessed identifiers

Tunneled packet - target gNodeB

gNB	UPF	gNB	UPF	Attacker	Attacker	**gNB**	**UPF**	**gNB**	**Victim**	src	**Victim**
IP		UDP		GTP	IP		UDP		GTP	IP	
Outer GTPH					Inner GTPH					Payload	

Tunneled packet - target UPF

gNB	UPF	gNB	UPF	Attacker	Attacker	**UPF**	src	**UPF**	Victim	**Victim**	dst
IP		UDP		GTP	IP		UDP		GTP	IP	
Outer GTPH					Inner GTPH					Payload	

Tunneled packet sample

```
> Internet Protocol Version 4, Src: 22.10.0.2, Dst: 22.10.0.1
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
v GPRS Tunneling Protocol
    > Flags: 0x34
      Message Type: T-PDU (0xff)
      Length: 74
      TEID: 0x0000bc42 (48194)
      Next extension header type: PDU Session container (0x85)
    > Extension header (PDU Session container)
> Internet Protocol Version 4, Src: 10.45.0.9, Dst: 22.10.0.1
> User Datagram Protocol, Src Port: 9090, Dst Port: 2152
v GPRS Tunneling Protocol
    > Flags: 0x30
      Message Type: T-PDU (0xff)
      Length: 30
      TEID: 0x0000000e (14)
> Internet Protocol Version 4, Src: 10.45.0.9, Dst: 22.10.0.3
> User Datagram Protocol, Src Port: 9090, Dst Port: 9090
> Data (2 bytes)
```

Boundary traversal

- Lack interface isolation and packet path validation
  - Perimissive routing opens internal paths even with physical or logical separation
  - e.g., Opens a non-existent path from UPF to AMF via *SCTP/NGAP setup*
- UPF to SMF
  - Existent and accessible with simple *PFCP association*
- Source-NAT can distort traffic origin visibility
  - UPF applies source NAT to packets from UE
  - AMF or SMF trust attacker-generated SCTP or PFCP packets as they appear to originate from the UPF itself

Boundary traversal

Tunneled packet - target AMF

gNB	UPF	gNB	UPF	Attacker	Attacker	**UPF**	src	**UPF**	**victim**	**victim**	**AMF**	src	**AMF**
IP		UDP		GTP	IP		UDP		GTP	IP		SCTP	
Outer GTPH					Inner GTPH					NGAP			

Tunneled packet - target SMF

gNB	UPF	gNB	UPF	Attacker	Attacker	**UPF**	src	**UPF**	**victim**	**victim**	**SMF**	src	**SMF**
IP		UDP		GTP	IP		UDP		GTP	IP		UDP	
Outer GTPH					Inner GTPH					PFCP			

TEID Enumeration - how

Exploiting standard comliant error responses in tunnel management messages

3GPP TS 29.281 (Sec 7.3)

#	IP address	TEID	Action taken by UPF
1	Unassigned	Existent	IP spoofing detected (packet drop)
2	Assigned	Existent not matching	IP spoofing detected (packet drop)
3	Assigned	Matching	Process packet
4	Both	Non-existent	GTP error indication

Exploitable for Enumeration

TEID Enumeration - how

As seen from the attacker mobile

o.	Time	Source	Destination	Protocol	Length	Info
7	0.00…	10.45.0.3	10.33.33.13	GTP <ICMP>	128	Echo (ping) request id=0x0005, seq=1/256, ttl=64
8	0.00…	10.45.0.3	10.33.33.13	GTP <ICMP>	128	Echo (ping) request id=0x0005, seq=1/256, ttl=64
9	0.00…	10.45.0.3	10.33.33.13	GTP <ICMP>	128	Echo (ping) request id=0x0005, seq=1/256, ttl=64
10	0.00…	10.45.0.3	10.33.33.13	GTP <ICMP>	128	Echo (ping) request id=0x0005, seq=1/256, ttl=64
11	0.00…	10.33.33.13	10.45.0.3	GTP	60	Error indication
12	0.00…	10.33.33.13	10.45.0.3	GTP	60	Error indication
13	0.00…	10.33.33.13	10.45.0.3	GTP	60	Error indication
14	0.00…	10.33.33.13	10.45.0.3	GTP	60	Error indication
15	0.00…	10.33.33.13	10.45.0.3	GTP	60	Error indication
16	0.00…	10.33.33.13	10.45.0.3	ICMP	84	Echo (ping) reply id=0x0005, seq=1/256, ttl=64
17	0.00…	10.33.33.13	10.45.0.3	GTP	60	Error indication

```
        0000 .... = PDU Type: DL PDU SESSION INFORMATION (0)
        .... 0000 = Spare: 0x0
        0... .... = Paging Policy Presence (PPP): Not Present
        .0.. .... = Reflective QoS Indicator (RQI): Not Present
        ..00 0001 = QoS Flow Identifier (QFI): 1
      Next extension header type: UDP Port number (0x40)
  ▾ Extension header
      Extension Header Length: 1
      UDP Port: 0
      Next extension header type: No more extension headers (0x00)
    TEID Data I: 0x000087b4 (34740)
  ▾ GSN address : 10.33.33.13
      GSN address length: 4
      GSN address IPv4: 10.33.33.13
```

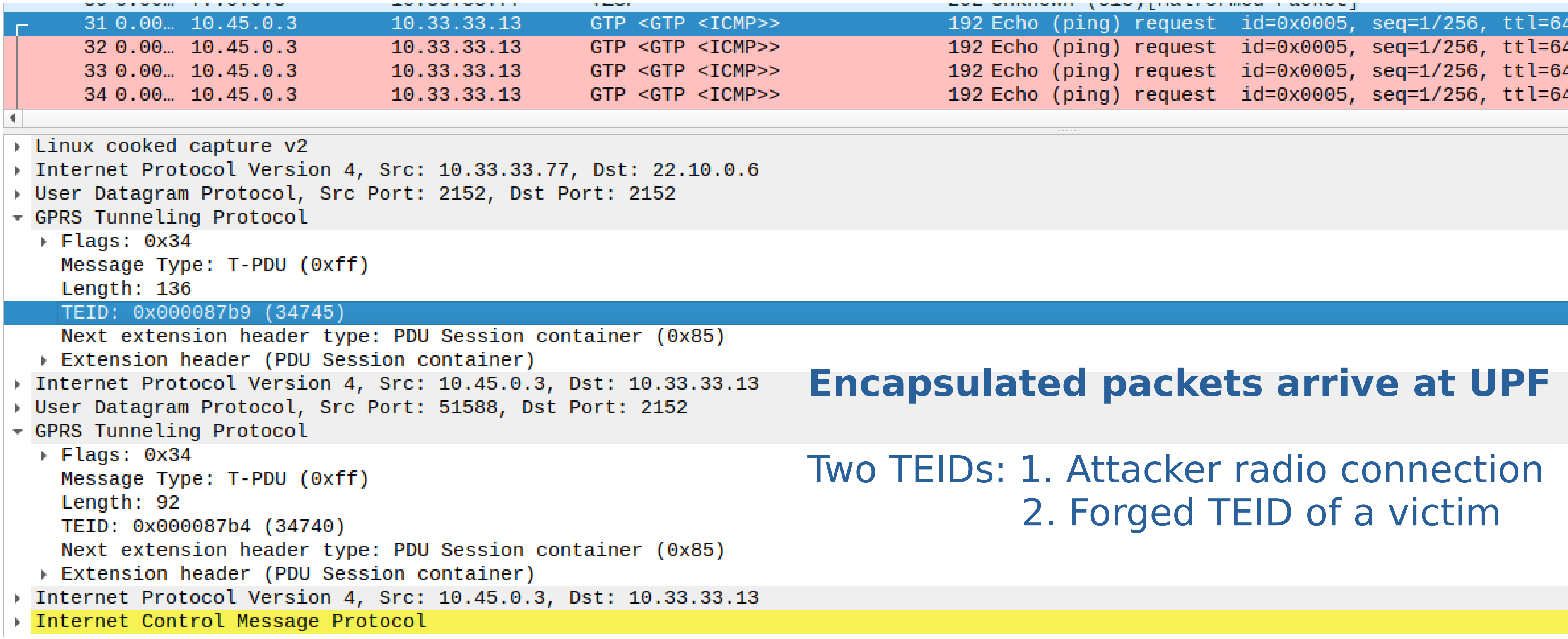
Error indications for all invalid TEIDs

No error indications for all valid TEIDs

If TEID-IP matches **ping reply**

TEID Enumeration - how

As seen from the UPF



Encapsulated packets arrive at UPF

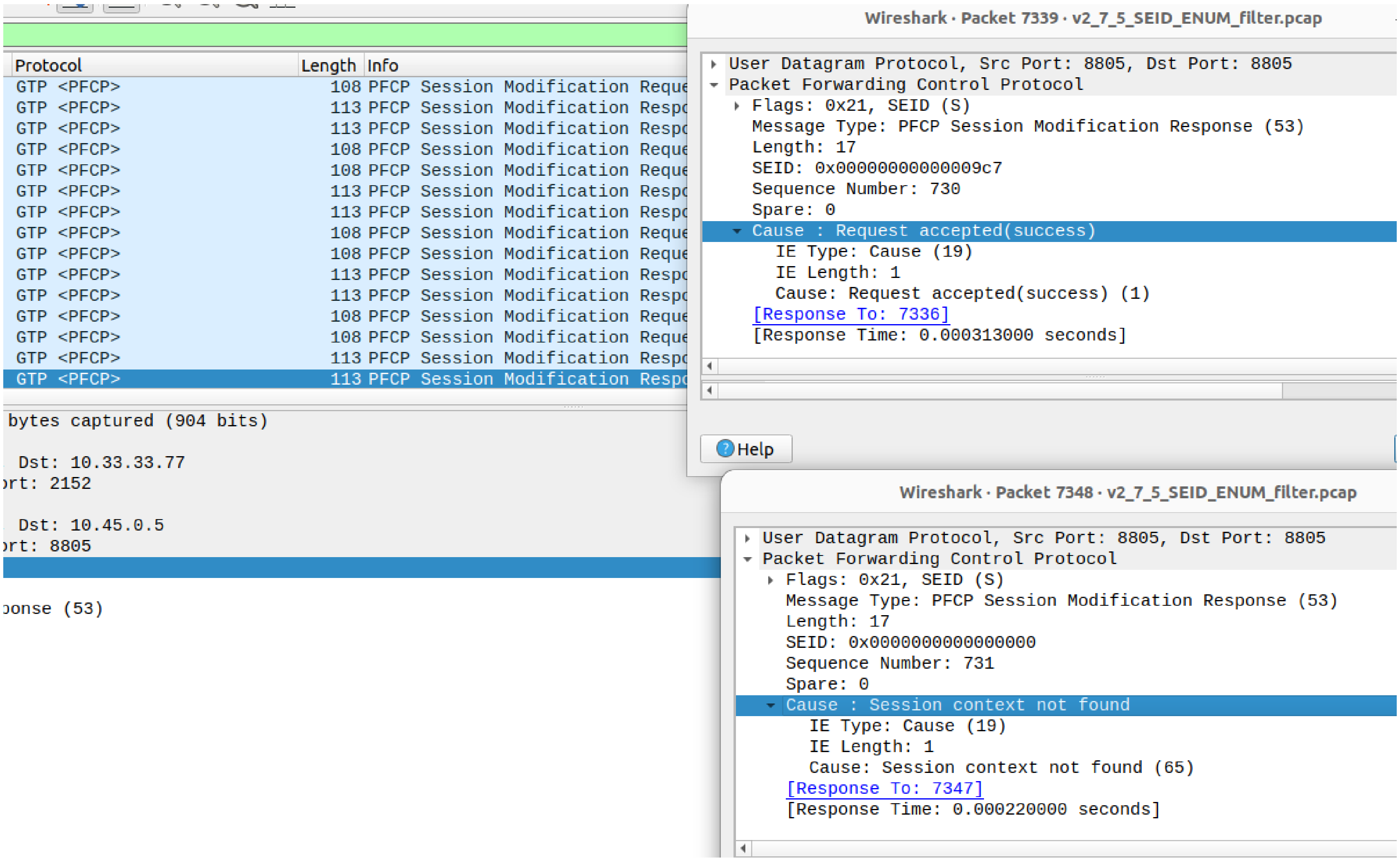
Two TEIDs: 1. Attacker radio connection
2. Forged TEID of a victim

Abonrmal behavior for PFCP

- Specification ambiguities
  - Undefined behavior when sessions are established without any rules
    - Resulting a **DoS**: All cores create dummy sessions and waste resources
    - Some cores crash after receiving 4096 requests, terminating all existing sessions
    - Some crash for empty requests: unexpected code flow
  - Implementation differences
    - Missing authentication of the SEID-IP tuple; allows for source authentication
    - Failure to do so allows attackers to manipulate sessions by replaying or guessing SEIDs
    - Majority cores did not implement this functionality; some ambiguity

SEID Enumeration - how

Exploiting standard compliant error responses in session management messages



Success factors for enumeration

- Speed: Depends on identifier space and allocation pattern
- Multiple smartphone connection paths – speed up enumeration
- **No rate limiting**
- One TEID-IP pair is sufficient for attack and can be cracked in seconds
- Ongoing connections are not interrupted - stealthy

TEID ->

Core	Allocation	Enumeration	Time
Open5GS	2B Random	Possible	seconds
Free5GC	4B Incremental	Possible	hours
OAI-5G	4B Random	Prohibited	infinte
SD-Core	4B Incremental	Possible	hours
CC1	4B Random	Prohibited	infinite
CC2	4B Incremental	Allowed	hours

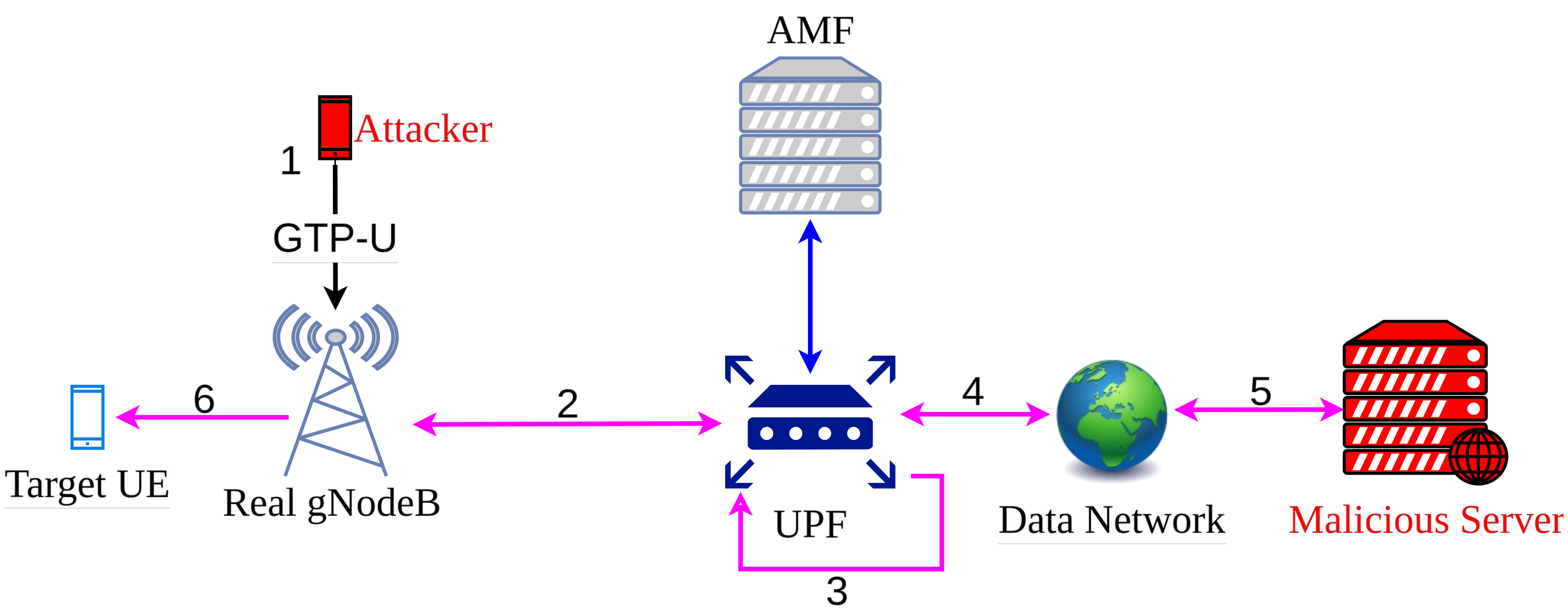
SEID ->

Core	Allocation	Enumeration	Time
Open5GS	12bit Random	Possible	seconds
Free5GC	8B Incremental	Possible	hours
OAI-5G	8B Incremental	Possible	hours
SD-Core	8B Random	Possible	infinite
CC1	8B Incremental	Possible	hours
CC2	8B Incremental	Possible	hours

Using this
in the real world

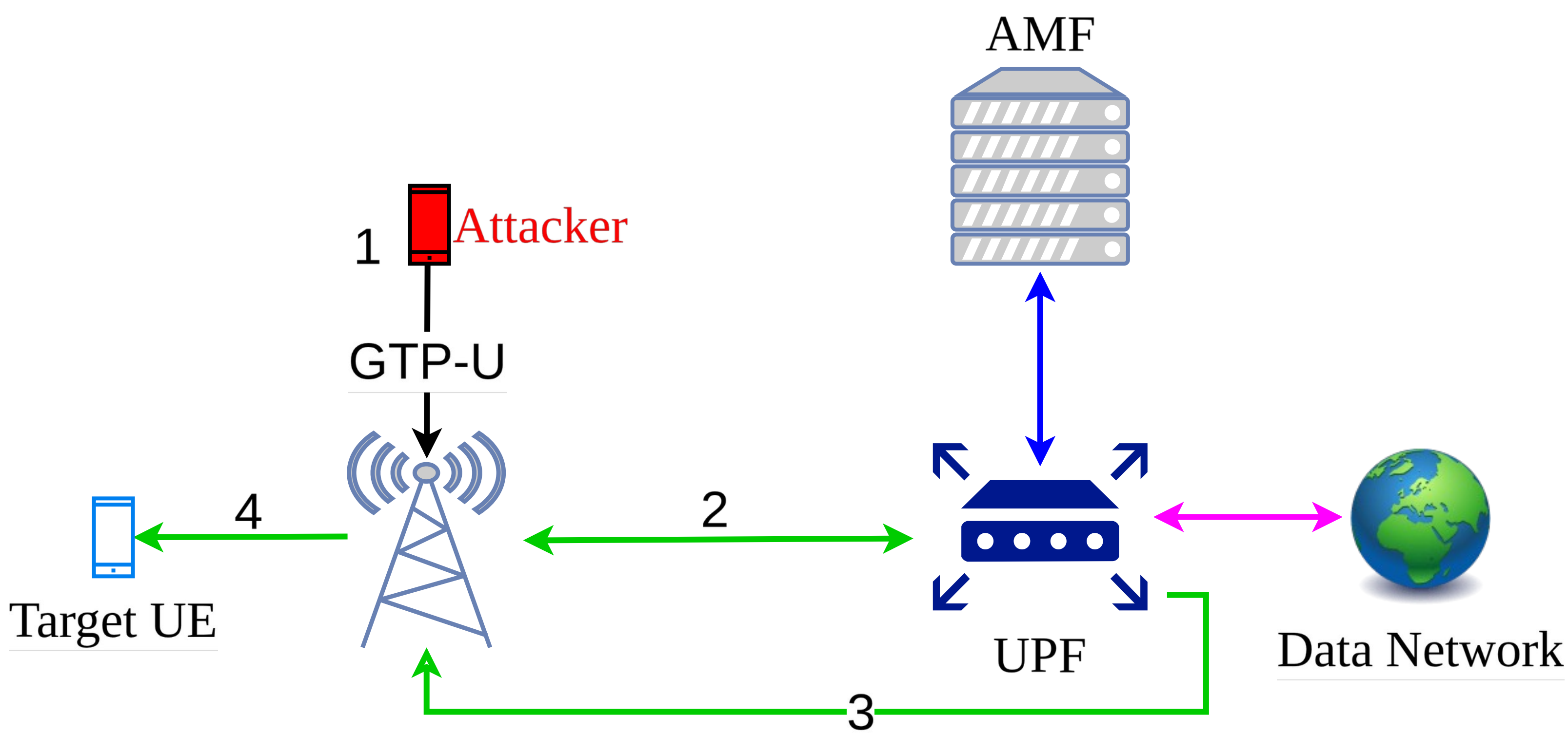
Reflective injection

- redirect traffic through a victim UE's uplink, enabling reflective delivery of unsolicited traffic to UEs
  - charging fraud where billing system attributes traffic volume to victim
  - bypass inbound filtering to otherwise unreachable UEs
- Amplified reflection: small spoofed query can trigger a large response
  - exhaust both uplink and downlink quotas



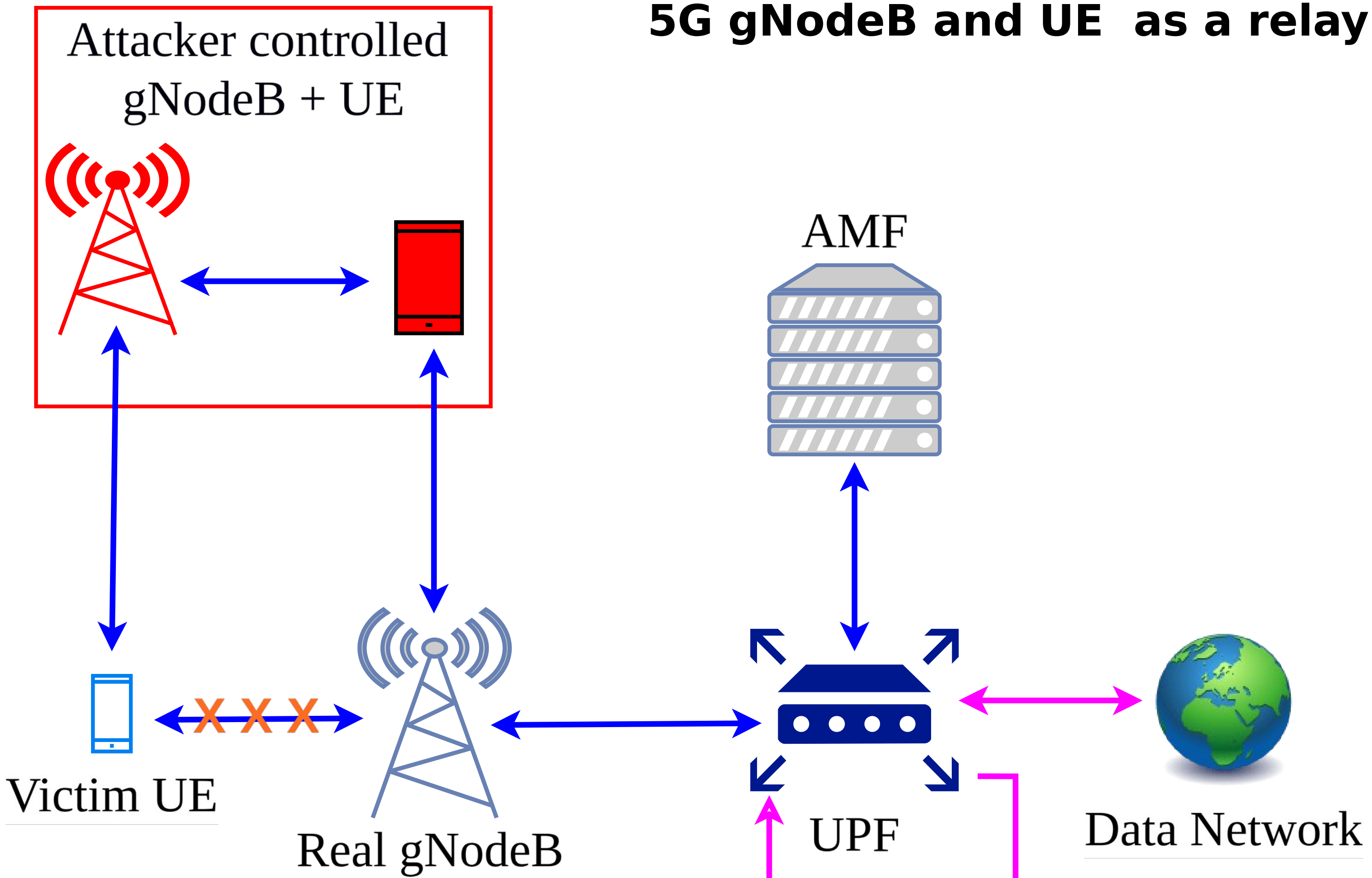
Direct routes to target UEs

- Direct and covert data injection into a UE, bypassing standard data path potentially evading any network layer defenses at the UPF preventing east-west traffic
- Bypassing the standard uplink–core–downlink data path and avoiding involvement of the external data network.



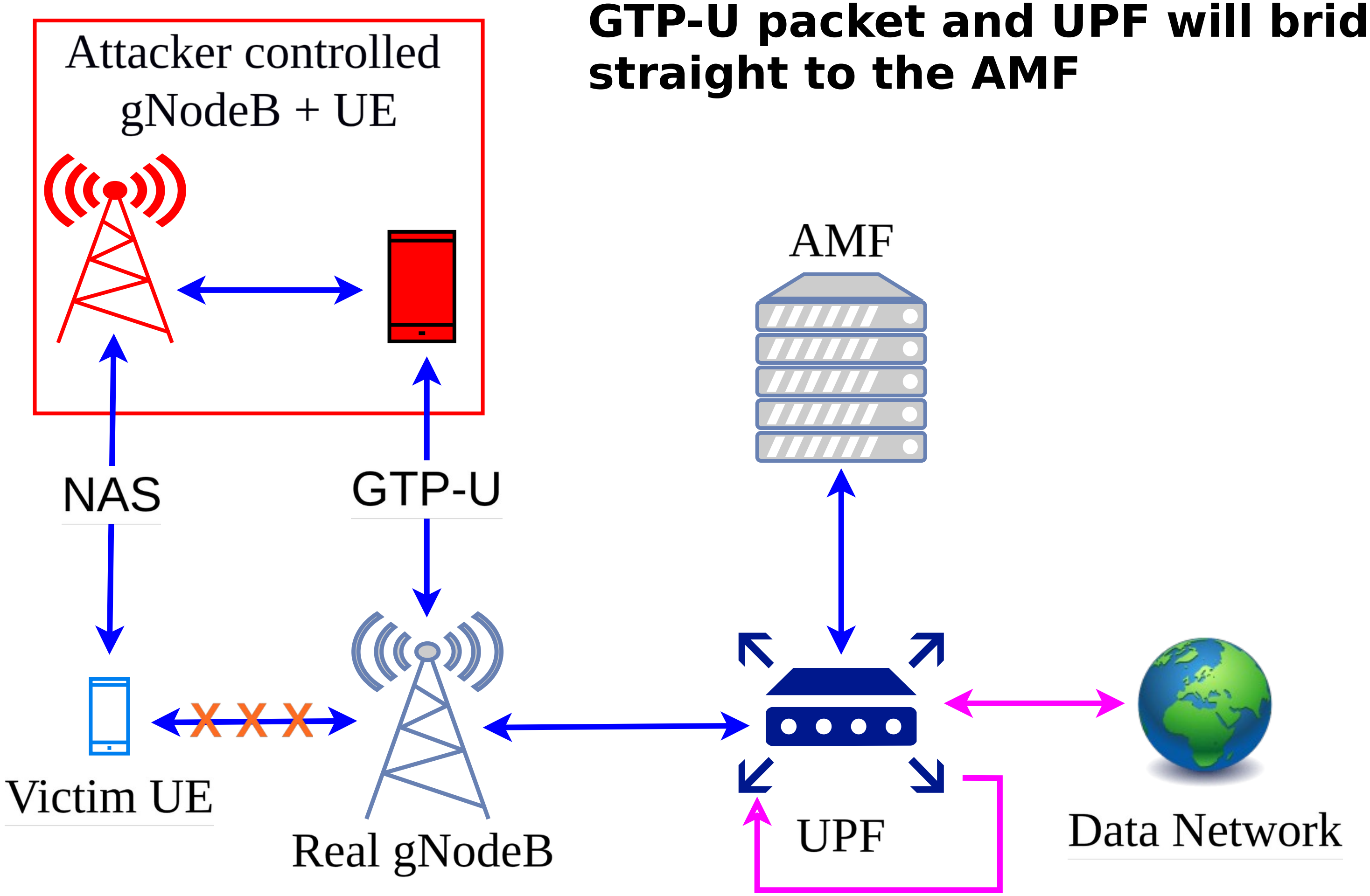
A legitimate MITM

Operating a legitimate rogue 5G gNodeB and UE as a relay



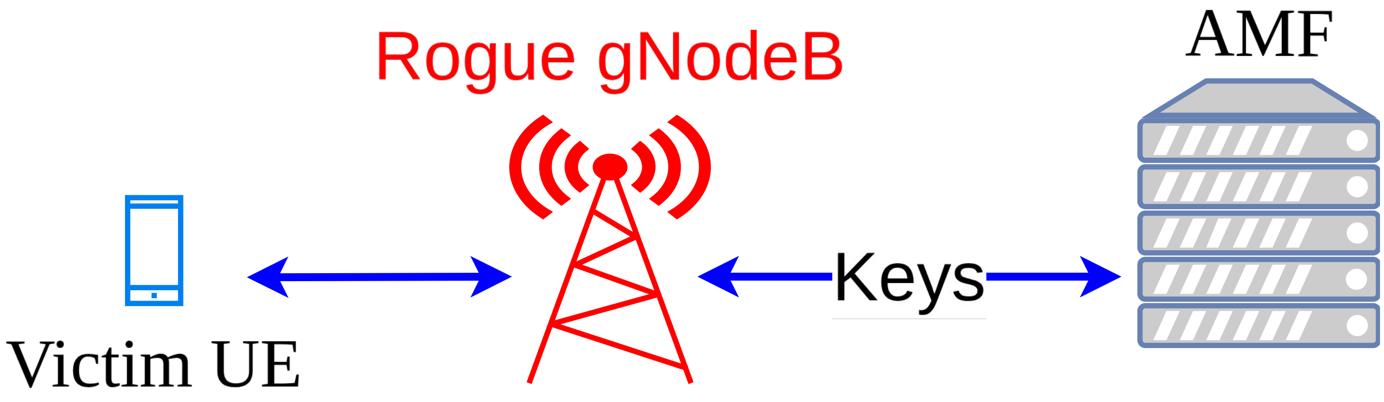
A legitimate MITM

Attacker tunnels NGAP/NAS traffic in a GTP-U packet and UPF will bridge it straight to the AMF



A legitimate MITM

Encryption and intergerity protection keys are directly handed over to attacker controlled gNodeB



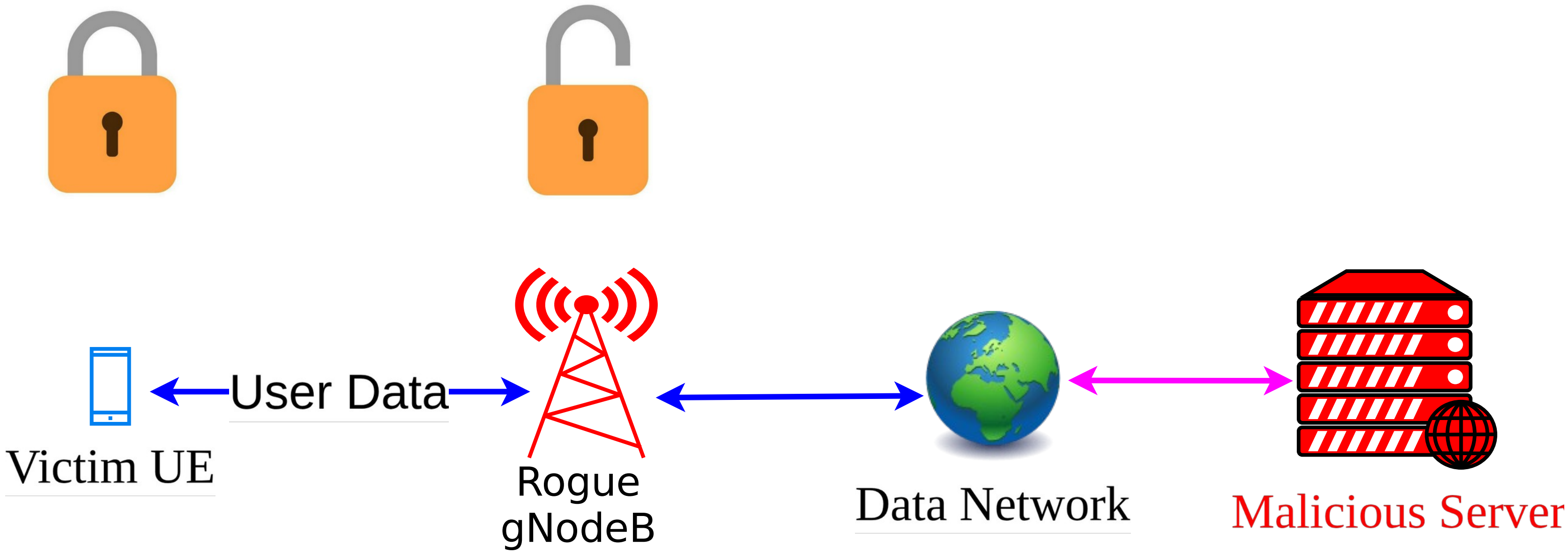
NGAP tunneled inside GTP-U

SCTP and NGAP encapsulated inside attacker’s GTP session

Protocol	Length	Info
GTP <SCTP>	132	INIT
GTP <SCTP>	356	INIT_ACK
GTP <SCTP>	328	COOKIE_ECHO
GTP <SCTP>	100	COOKIE_ACK
GTP <NGAP>	184	NGSetupRequest
NGAP	128	NGSetupResponse
GTP <SCTP>	172	DATA (TSN=0) (retransmission)
GTP <SCTP>	156	HEARTBEAT
GTP <SCTP>	156	HEARTBEAT_ACK
GTP <NGAP/NAS-5GS>	188	InitialUEMessage, Registration request
NGAP/NAS-5GS	152	SACK (Ack=1, Arwnd=106496) , DownlinkNASTransport, Authentication request
GTP <NGAP/NAS-5GS>	196	SACK (Ack=1, Arwnd=106496) , UplinkNASTransport, Authentication response
NGAP/NAS-5GS	132	SACK (Ack=2, Arwnd=106496) , DownlinkNASTransport, Security mode command
GTP <SCTP>	176	SACK (Ack=2, Arwnd=106496) DATA (TSN=2) (retransmission)
GTP <NGAP/NAS-5GS/NAS-5GS>	240	SACK (Ack=2, Arwnd=106496) , UplinkNASTransport, Security mode complete, Registration
NGAP/NAS-5GS	248	SACK (Ack=3, Arwnd=106496) , InitialContextSetupRequest, Registration accept
GTP <SCTP>	292	SACK (Ack=3, Arwnd=106496) DATA (TSN=3) (retransmission)
GTP <NGAP/NAS-5GS>	292	UplinkNASTransport, Registration complete, UplinkNASTransport, UL NAS transport, PDU
NGAP/NAS-5GS	148	SACK (Ack=6, Arwnd=106496) , DownlinkNASTransport, Configuration update command
GTP <SCTP>	192	SACK (Ack=6, Arwnd=106496) DATA (TSN=4) (retransmission)
NGAP/NAS-5GS	256	PDUSessionResourceSetupRequest, DL NAS transport, PDU session establishment accept
GTP <SCTP>	300	DATA (TSN=5) (retransmission)
GTP <NGAP>	152	PDUSessionResourceSetupResponse
GTP <SCTP>	156	HEARTBEAT
GTP <SCTP>	156	HEARTBEAT_ACK

Legitimate interception

- GnodeB receives crypto keys from AMF for security setup with UE
  - Full visibility to authentication and registration process
  - Custom UPF or forward traffic directly to external networks, bypassing the legitimate UPF
  - Bi-directional IP traffic to flow through the rogue gNodeB as if the connection were legitimate



Impact

- Full interception & redirection of user traffic by a attacker-controlled gNodeB
  - Attacker gains control over critical functions such as user data paths, DNS resolution, handovers, and service availability
  - All inside an legitimate and encrypted session
- Voice call (VoNR) can be intercepted, SMS delivery can be controlled
- **Cannot defend: existing 5G security mechanisms—such as mutual authentication, encryption, integrity protection, and downgrade prevention**
- Previously required sophisticated setups in 4G can now be executed over a simple data connection, significantly lowering the barrier to exploitation.
- Stingray detectors and all UE-side security solutions will fail

The root problem

Long sustained protocol

- GTP-U: Notorious protocol from 2G still used in 5G and maybe in 6G too
  - Due to simple forwarding, low performance overhead
  - Inherently suitable for tunneling
  - lacks built-in integrity checks or source authentication
  - forwarding based solely on the destination IP and TEID
  - design does not inspect header and payload contents

- Modern UPFs are processing tunneled or encapsulated packets
  - Permits control plane protocol payloads and bridge them to AMF/SMF

Rethinking trust in the user plane

No easy solution

- Tunneling is well exploited over roaming interfaces
- Complex infrastructures to be seen with 5G slicing, virtualized, private cores, edge computing.
  - Privately controlled UPFs – prone to misconfigurations
  - Skills in understanding the attacks, abnormal protocol flows
- Expensive solutions from vendors – limited budget, no monitoring (takeaways from latest telco incidents)
- GTP exploited by Liminal panda to tunnel C2 traffic
  - security solutions less likely to inspect and restrict GTP-encapsulated traffic [ref]
- Regulations and restrictions around GTP and user plane data inspection

Recommendations & way forward

Disclosure

- All open source developers and commercial vendors are notified
- Some fixed it and some require budget approvals and more scrutiny
- CVEs in progress
- Disclosed to GSMA in their FSAG meeting
  - Work in progress to include the attacks in this research to GTP security guidelines and recommendations

Fixing it

- Firewalls recommended, extensive guidelines from GSMA (IR.88, FS.37)
- Underlying root cause fixes need systemic level changes
  - Handling GTP-U and its malicious mutations
- Tackling the protocol design
  - Encapculation depth, rate limiting, TEIS/SEID allocation & management
- Routing security into UPF
  - security into packet-processing frameworks
- Misconfigurations: segmentation, routing awareness, isolation enforcement
- Dropping encapsulated GTP packets – already GMSA marks them fradulent
  - Not only packets from external GRX (or IPX) but packets from RAN too

Takeaways

- Modern UPFs still vulnerable to encapsulated GTP-U attacks
  - Opens door for tunneling and bridging attacks
- Insecure practices inside UPFs
  - Identifier allocation, management and rate limiting
- Six different 5G core networks tested and more than 80% of them are affected including commercial cores
- Vulnerable UPFs plus relaxed security setting inside core
  - New, powerful, and undetecteable attacks on subscribers and core
  - Billing fraud and legitimate MITM doing interception
- Insufficient guidelines on UPF secure design practices
- Full research will be published in ACM CCS this October and a preprint is here

The analogy: Titanic and 5G

- Titanic’s compartments = 5G’s isolated trust boundaries (control/user planes, network slices, interfaces).
- Iceberg impact = malicious UE traffic
- Water flowing over boundaries = protocol tunneling + boundary bridging.
- Overconfidence in “unsinkable” architecture = misplaced trust in standard 5G isolation.

Thank You!

Questions/Comments/Concerns?

altaf.shaik@fastiot.org