ACM SIGCOMM 2019 Student Research Competition

CellScope: Automatically Specifying and Verifying Cellular Network Protocols

Yinbo Yu[#*]

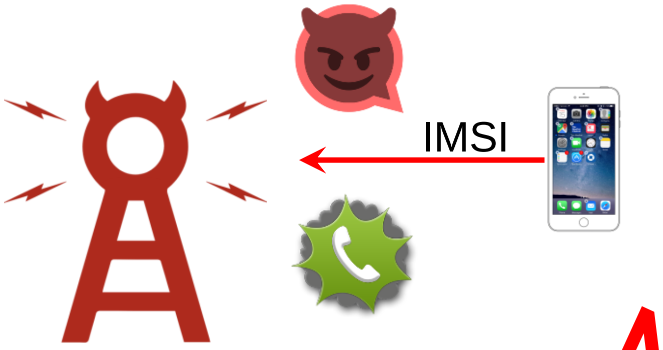
You Li[*], Kaiyu Hou[*], Yan Chen[*], Hai Zhou[*] and Jianfeng Yang[#]

[#]Wuhan University, [*]Northwestern University

WUHAN UNIVERSITY

Northwestern University

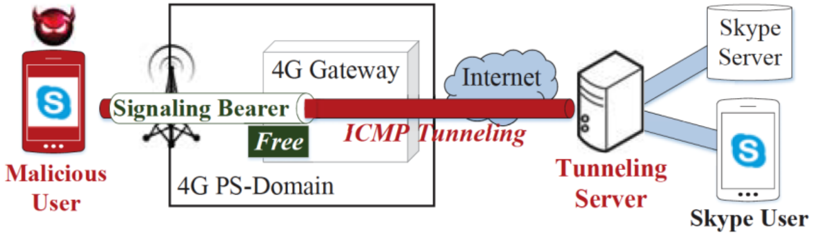
Cellular network



Fake base station

Location tracking

Not secure

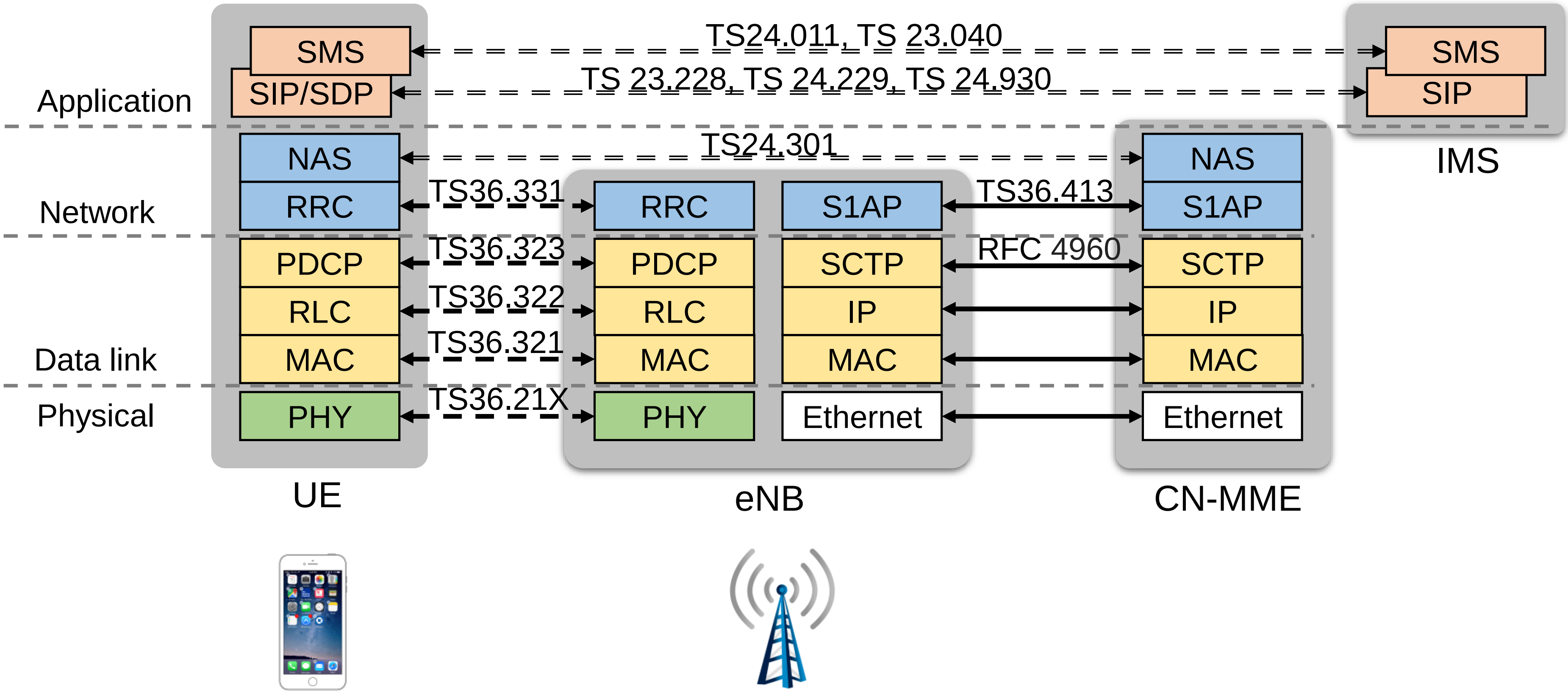


Free data access

No service

[1] A. Dabrowski etc. IMSI-Catch Me If You Can: IMSI-Catcher-Catchers. ACSAC'14
[2] Guan-Hua Tu etc. Control-Plane Protocol Interactions in Cellular Networks. SIGCOMM'14
[3] Chi-Yu Li etc. Insecurity of Voice Solution VoLTE in LTE Mobile Networks. CCS'15
[4] Altaf Shaik etc. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. NDSS'16
[5] Syed Rafiul Hussain etc. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. NDSS'18
[6] Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion. NDSS'19
[7] David Rupprecht etc. Breaking LTE on Layer Two. IEEE S&P'19

Protocol Stack



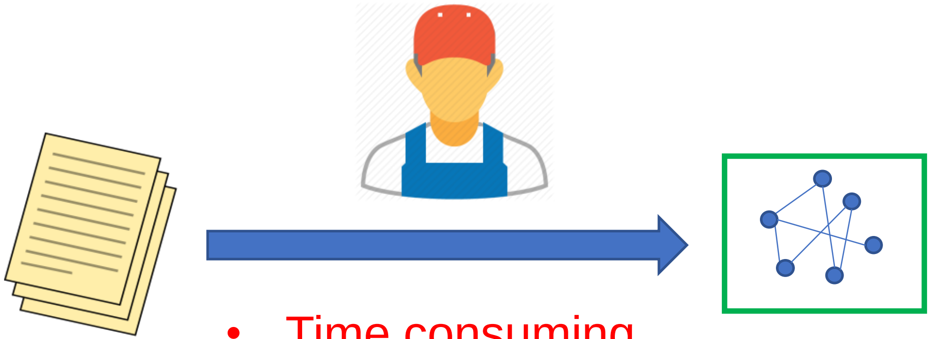
3GPP™

A GLOBAL INITIATIVE

Existing work

Formal Verification(SIGCOMM’14, NDSS’18,, NDSS’19**)**: specify protocols as formal models and verify with correctness properties

- Systematic and solid
- Manual specification



- Time consuming
- Error-prone

Challenges:

- Hundreds or thousands of pages of human language
- More standards specifying interaction behaviors among protocols
- Optional configurations

[1] Guan-Hua Tu etc. Control-Plane Protocol Interactions in Cellular Networks. SIGCOMM’14
[2] LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. NDSS’18
[3] Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion. NDSS’19

Our design

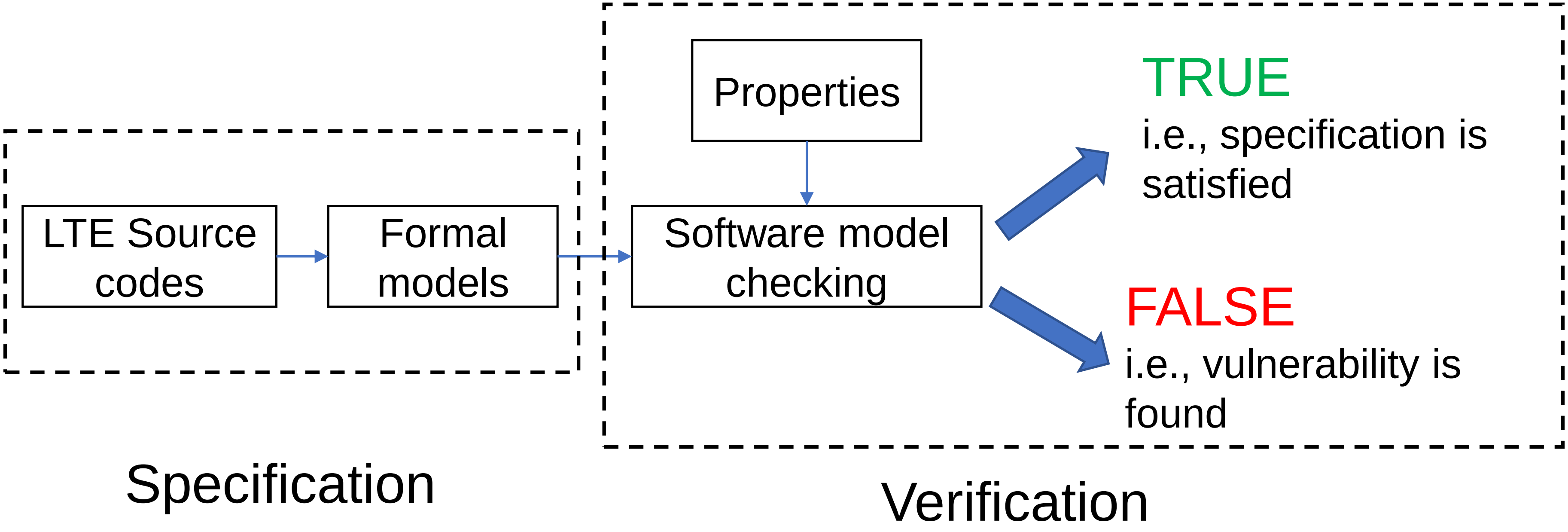
Is it possible to automatically specify and verify cellular network protocols?

Software model checking

Open-source implementations of Cellular network:

OPEN AIR INTERFACE 5G openLTE SRSLTE

Basic Idea

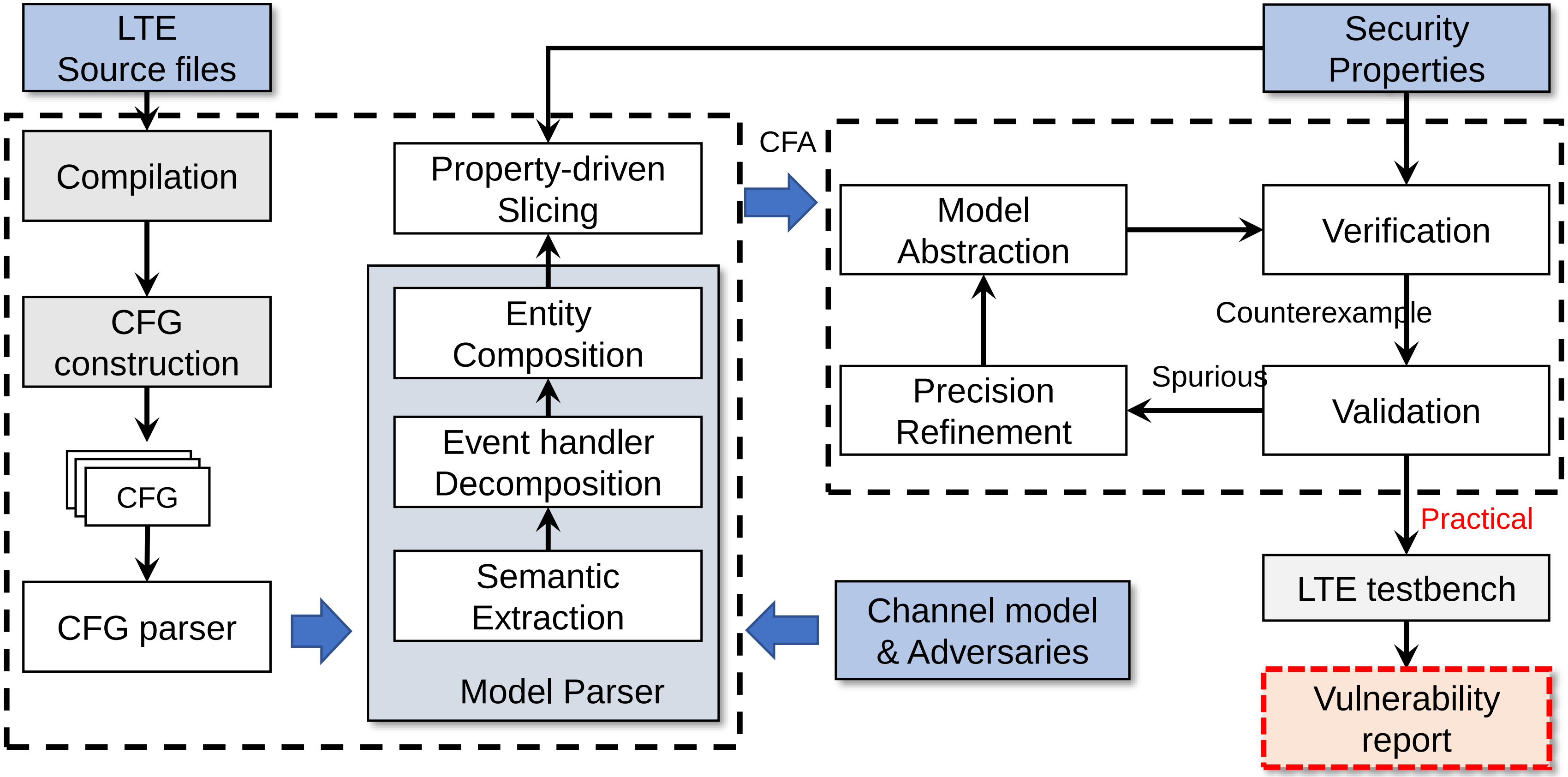


Properties
TRUE
i.e., specification is satisfied
LTE Source codes
Formal models
Software model checking
FALSE
i.e., vulnerability is found
Specification
Verification

Challenges

- **Size Explosion**: millions lines of code
- **Independent software entity**: multiple software entities (UE, eNB, and CN)
- **Multi-Agent Interaction**: each of the entity is driven by messages sent by each other.

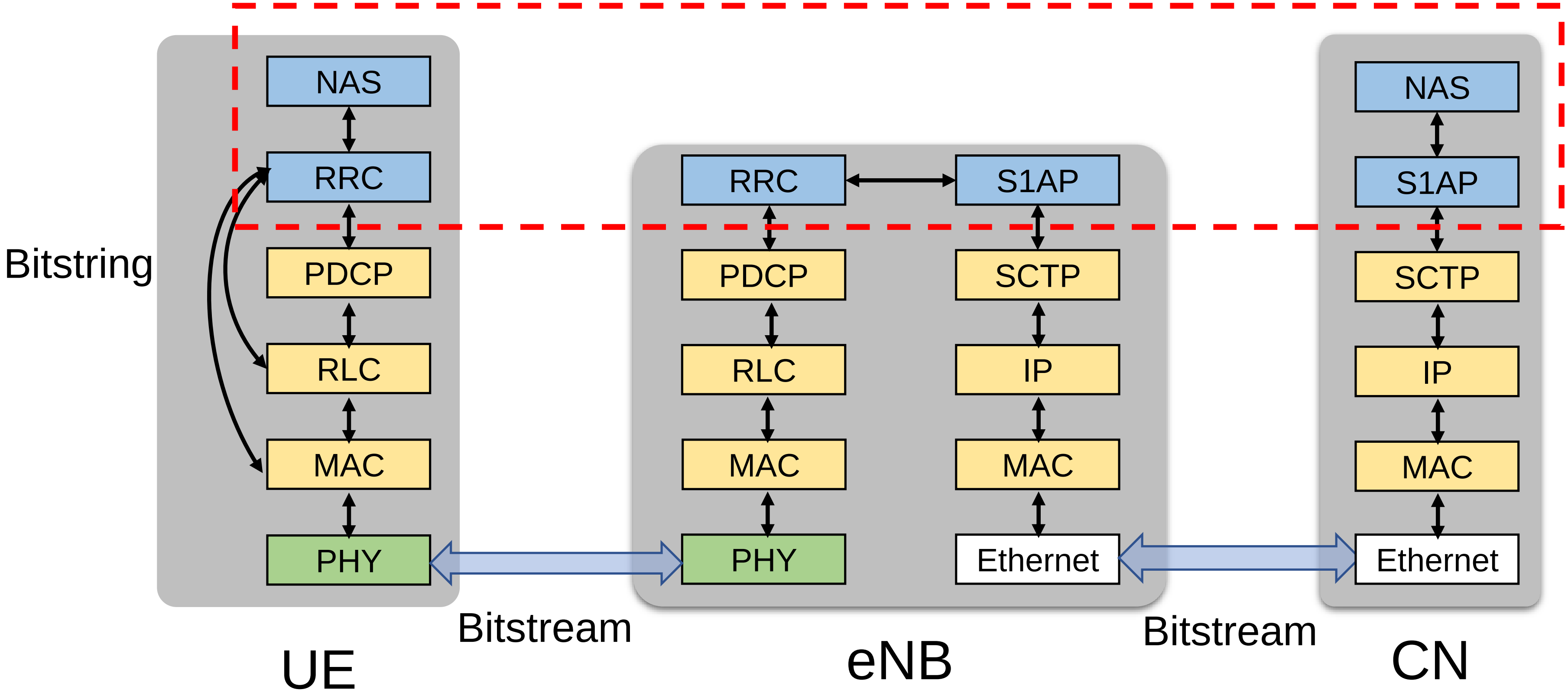
CellScope



Parser Frontend

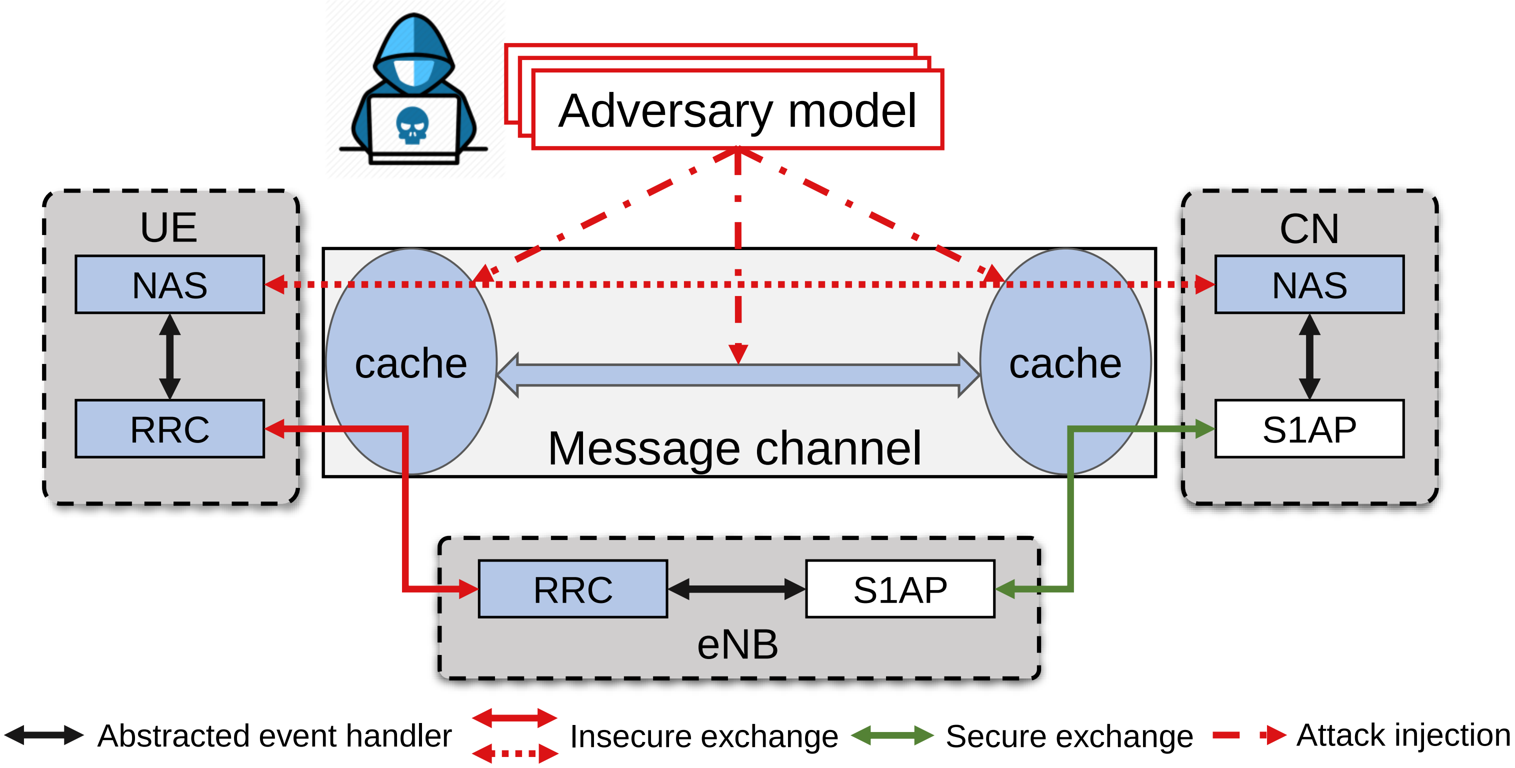
Verification Backend

Message Deliver



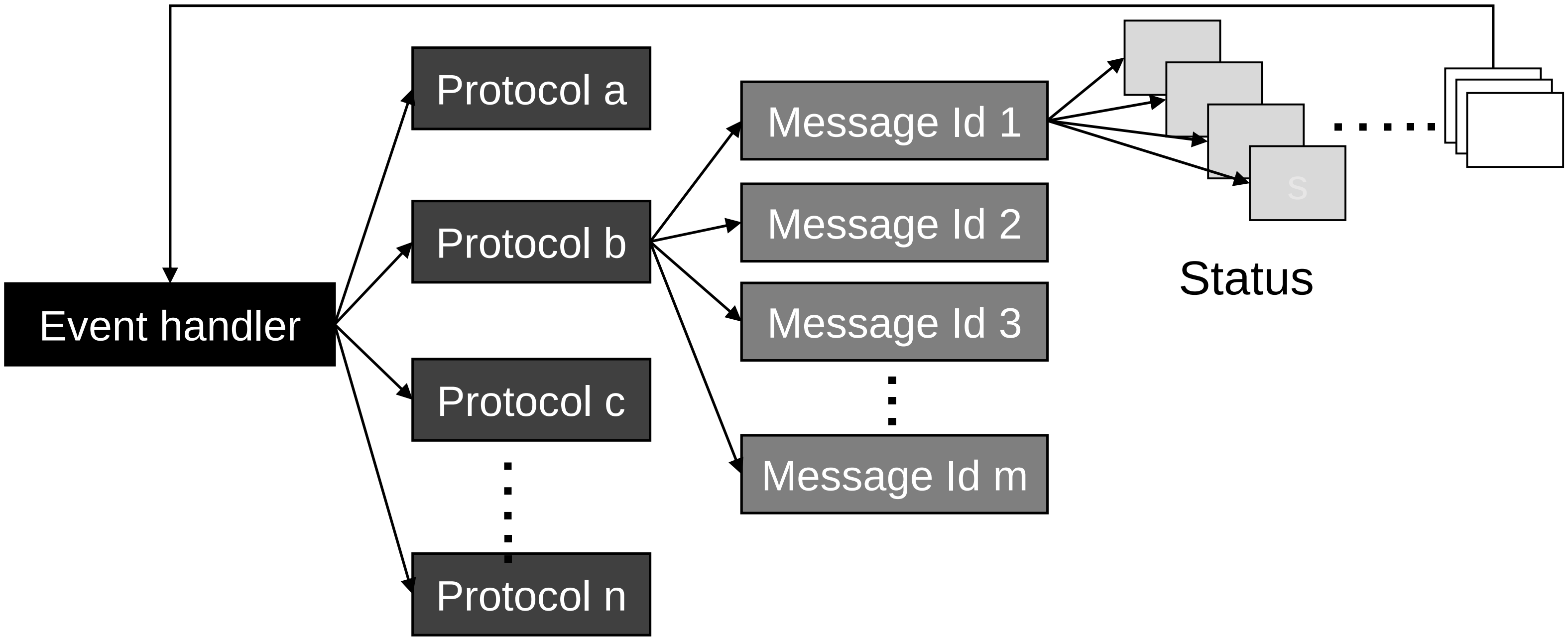
Communication channel

Formal message channel model



1. Mock up program behaviors in low layers
2. Formal message exchange models among software entities
3. Dolev-Yao style adversaries

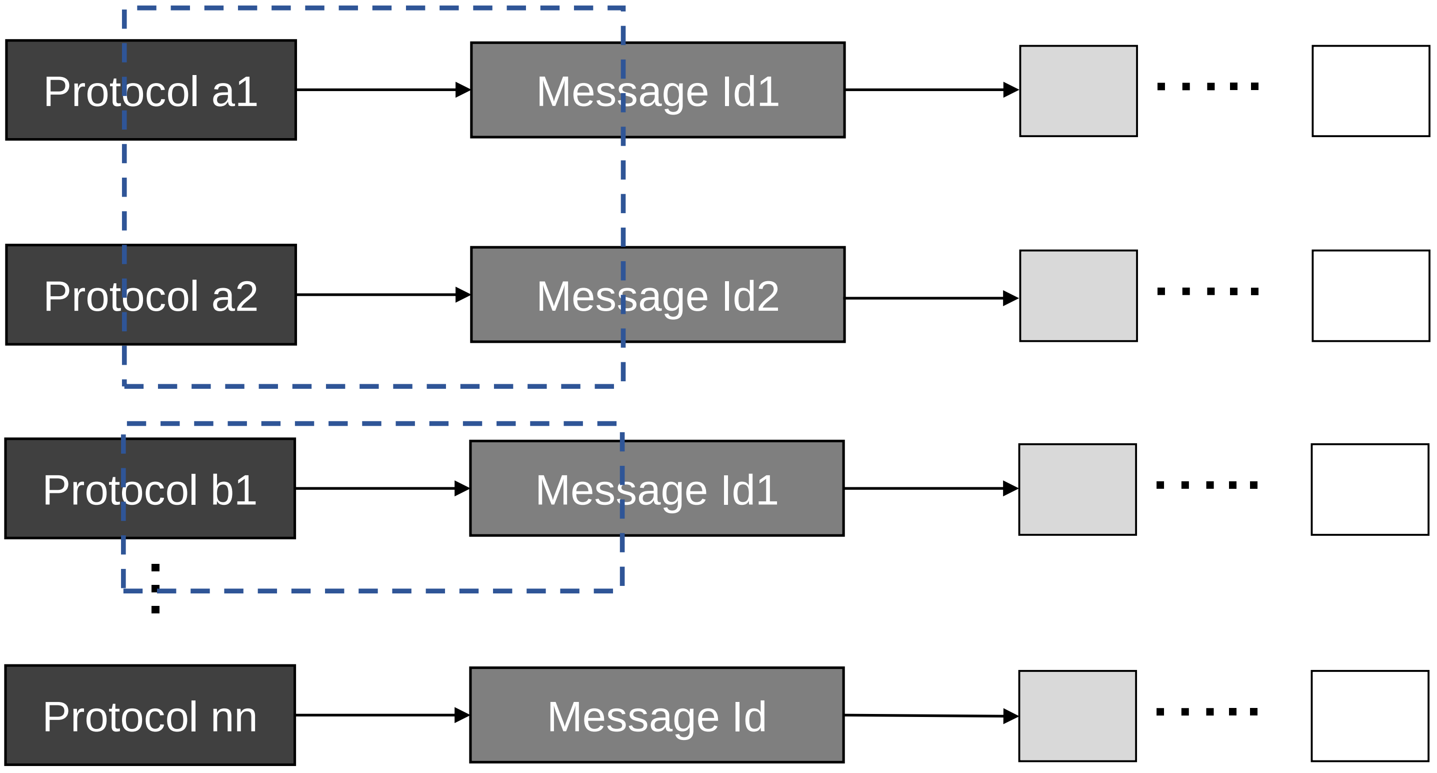
Decomposition of event handler



Tree mode

State space reduction

Decomposition of event handler



Linear mode

A lot of infeasible paths are removed

State space reduction

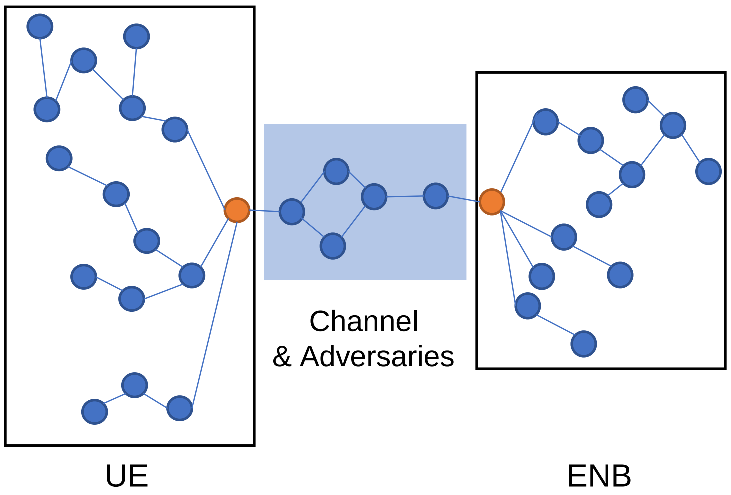
Property-driven slicing

Property

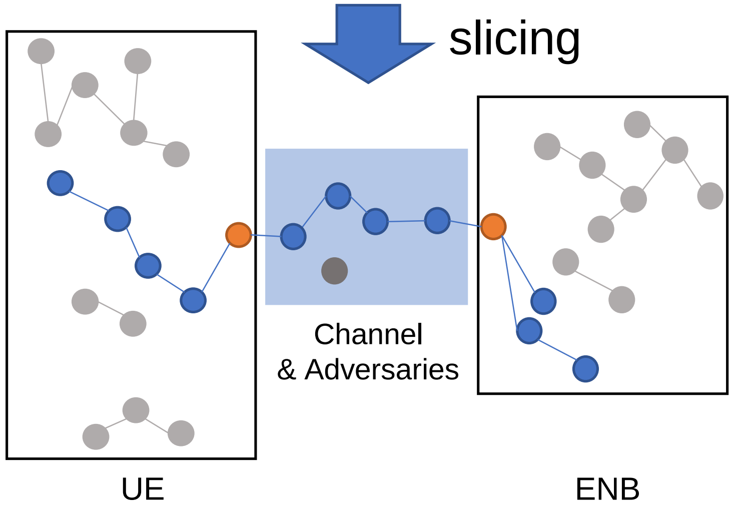
Adversary assumption

Slicing criteria

- Channel message
- Configurations
- Adversary variables
- …



Before slicing

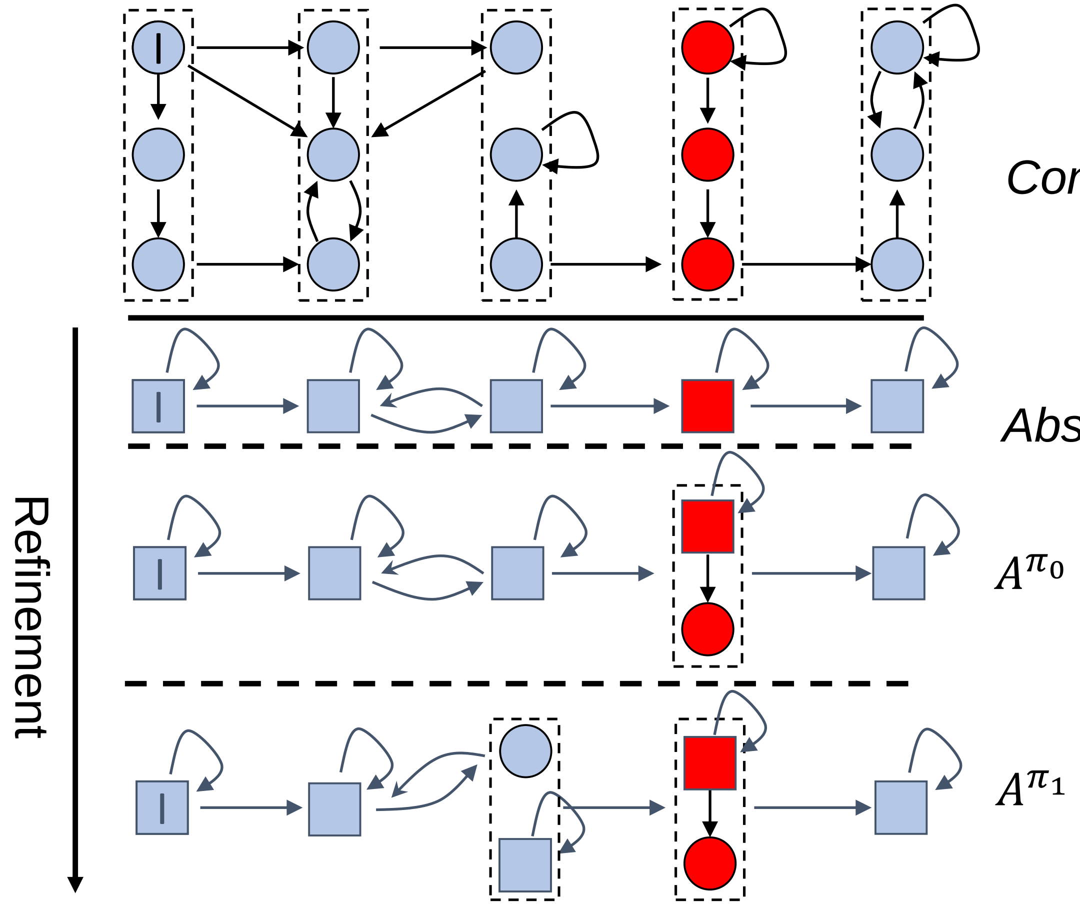


After slicing

Verification

CPAchecker: https://cpachecker.sosy-lab.org/

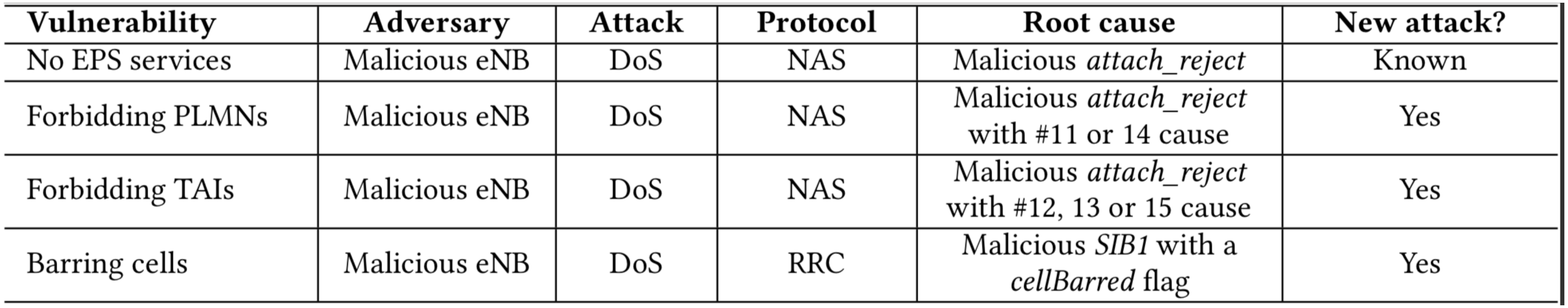
Priority counter-example guided abstraction refinement (P-CEGAR)



Variable:

- Protocol-related
- Dummy adversary
- Program control related

Primary results



Vulnerability	Adversary	Attack	Protocol	Root cause	New attack?
No EPS services	Malicious eNB	DoS	NAS	Malicious *attach_reject*	Known
Forbidding PLMNs	Malicious eNB	DoS	NAS	Malicious *attach_reject* with #11 or 14 cause	Yes
Forbidding TAIs	Malicious eNB	DoS	NAS	Malicious *attach_reject* with #12, 13 or 15 cause	Yes
Barring cells	Malicious eNB	DoS	RRC	Malicious *SIB1* with a *cellBarred* flag	Yes

Thanks

Q & A

Yinbo Yu: yyb@whu.edu.cn