Hermes: Unlocking Security Analysis of Cellular Network Protocols by Synthesizing Finite State Machines from Natural Language Specifications

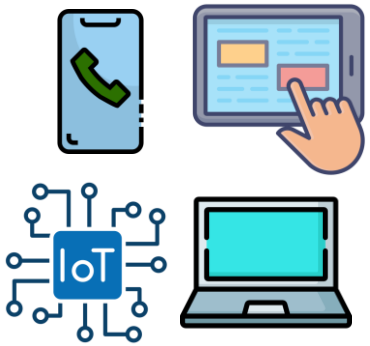
Abdullah Al Ishtiaq, Sarkar Snigdha Sarathi Das, Syed Md Mukit Rashid, Ali Ranjbar, Kai Tu, Tianwei Wu, Zhezheng Song, Weixuan Wang, Mujtahid Akon, Rui Zhang, Syed Rafiul Hussain

Systems and Network Security (SyNSec) Lab
Department of Computer Science and Engineering
Pennsylvania State University

PennState

SyNSec

Security Analysis of Cellular System

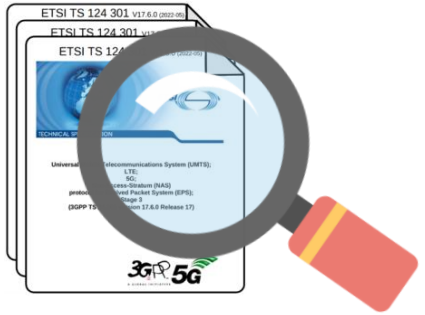


Billions of Devices

Security Sensitive Applications

Large Attack Surface

Security Analysis Ensures Secure Design

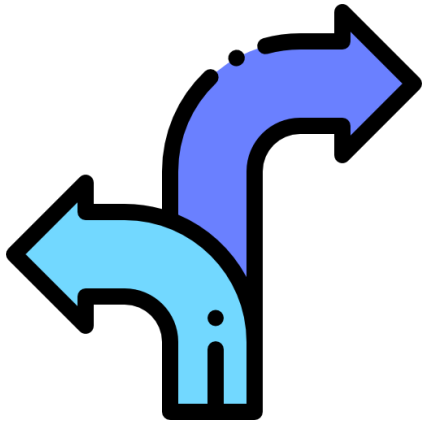


3GPP Design Specifications Needs Analysis

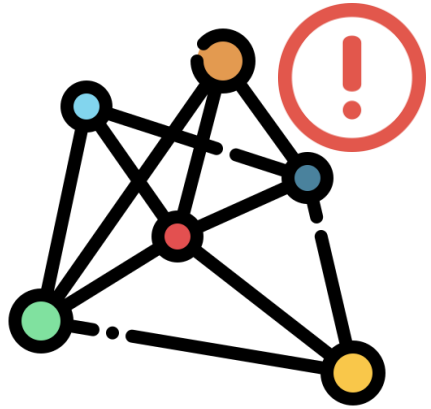
Issues in Natural Language Cellular Specifications

Hundreds of documents

Difficult to understand



Conflicts and underspecifications

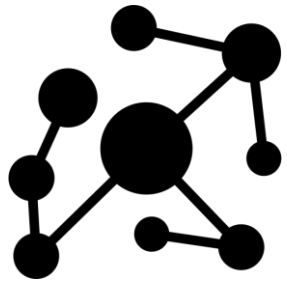


No formal Model

Formal Security Analysis of Cellular Systems

Specification Documents

Cellular Expert



Formal Model

Errors

Time consuming

Narrow scope

Specification Updates

Formal Security Analysis of Cellular Systems

Is it possible to automatically generate formal models from natural language specifications to aid security analysis of cellular systems?

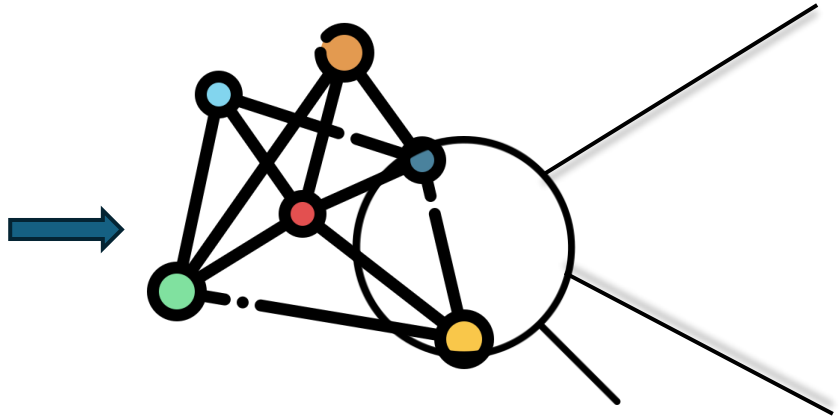
Formal Model Extraction from Specifications

The UE initiates … sending an ATTACH ACCEPT message.

The UE sends … with integrity protection.

Upon receiving … enter state EMM-REGISTERED.

…



start_state: EMM_REGISTERED_INITIATED

end_state: EMM_REGISTERED

condition: channel_MME_UE = attach_accept & attach_accept_integrity_protected

action: channel_UE_MME = attach_complete

Natural Language Specifications

FSM

Example Transition

Challenges of Automated Formal Model Extraction

On receipt of the SERVICE REJECT message, **if** the UE is in state EMM-SERVICE-REQUEST-INITIATED **and** the message is integrity protected **or** contains a reject cause other than EMM cause value #25, the UE shall reset the service request attempt counter.

```
start_state := emm_service_request_initiated

condition   :=
    (channel_MME_UE = service_reject) &
    (service_reject_integrity_protected |
    service_reject_emm_cause_present   &
    service_reject_emm_cause_value != cause_25)

action      := ue_service_req_attempt_counter = 0
```

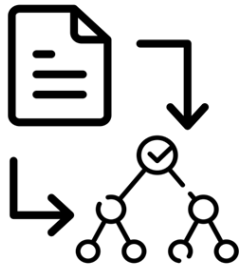
Input Text | **ML Model** | **Output Transition**

- Too complex of a task for existing NLP models
- No training data
- Need experts to annotate

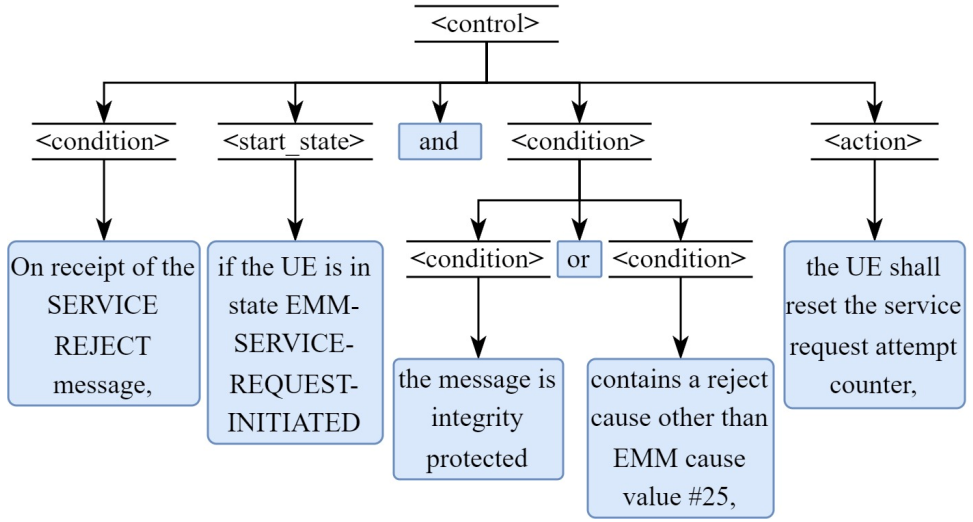
Workflow of Hermes

On receipt of the SERVICE REJECT message, if the UE is in state EMM-SERVICE-REQUEST-INITIATED and the message is integrity protected or contains a reject cause other than EMM cause value #25, the UE shall reset the service request attempt counter.

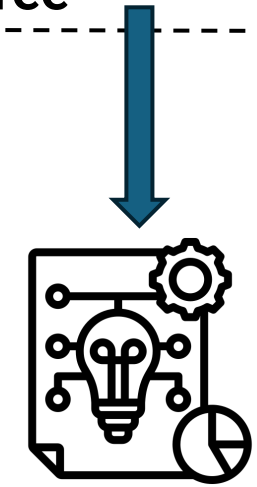
Input Text



NEUTREX

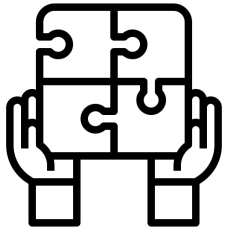


Constituency Parse Tree

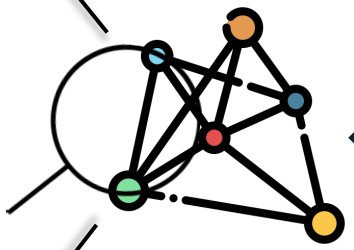


IR Synthesizer

Separate Logical formulas for transition components



FSM Synthesizer

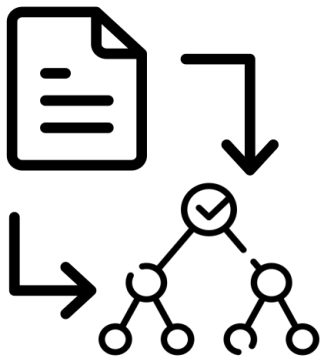


FSM

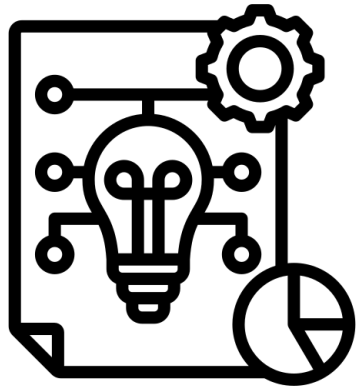
start_state := emm_service_request_initiated

condition :=
(channel_MME_UE = service_reject) &
(service_reject_integrity_protected |
service_reject_emm_cause_present &
service_reject_emm_cause_value != cause_25)

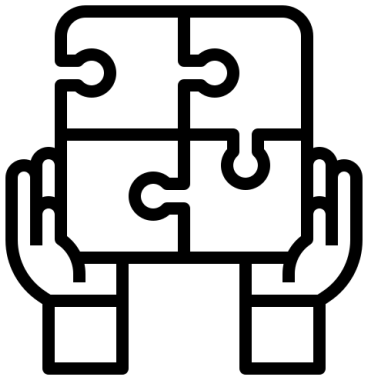
action := ue_service_req_attempt_counter = 0



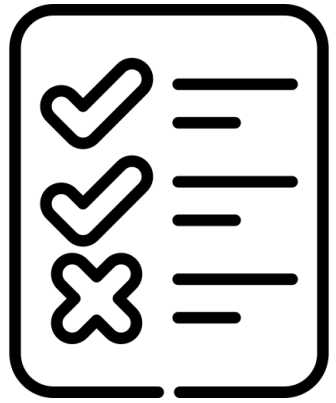
NEUTREX



IRSynthesizer



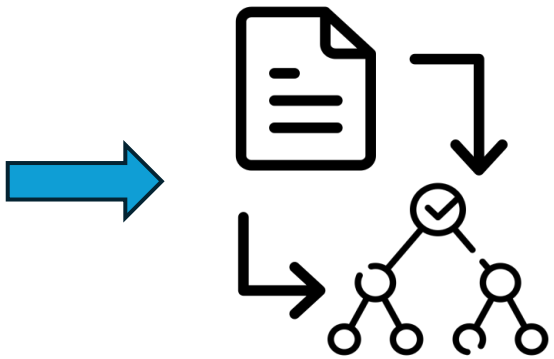
FSMSynthesizer

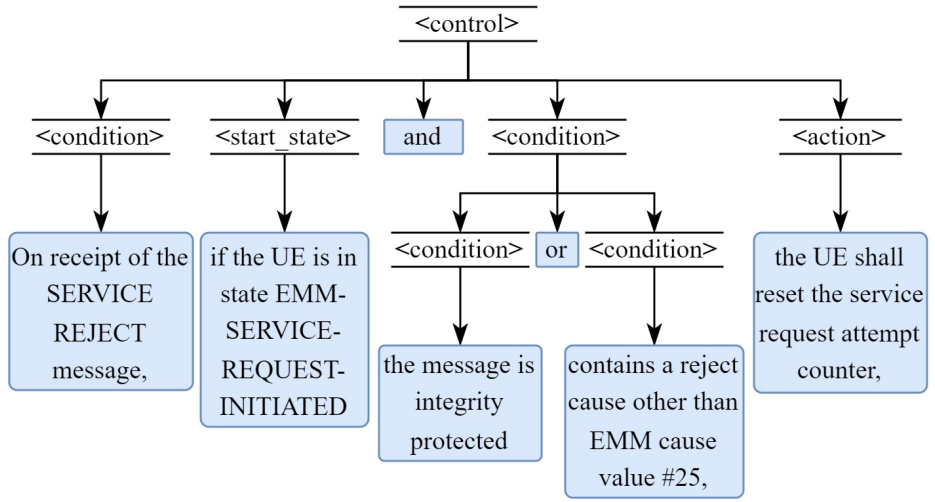


Findings

NEUTREX: Overview

On receipt of the SERVICE REJECT message, if the UE is in state EMM-SERVICE-REQUEST-INITIATED and the message is integrity protected or contains a reject cause other than EMM cause value #25, the UE shall reset the service request attempt counter.



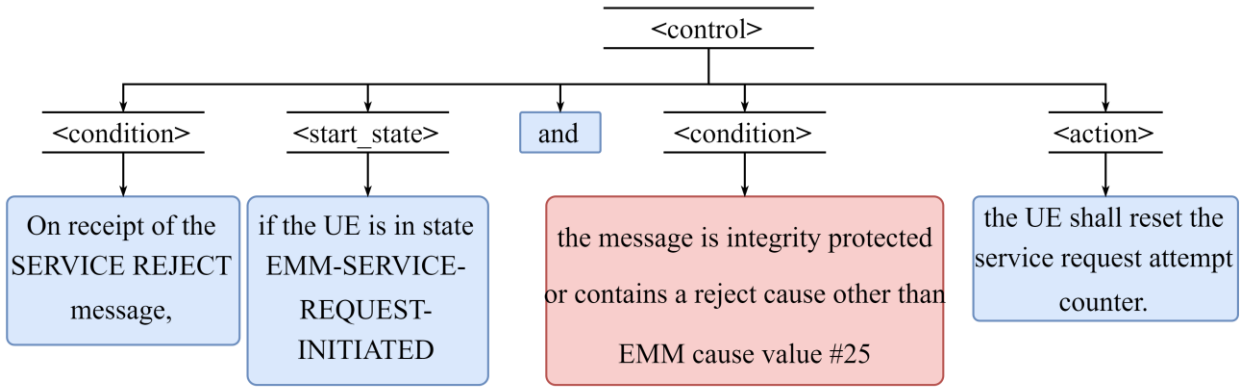


Input Text | **NEUTREX** | **Constituency Parse Tree**

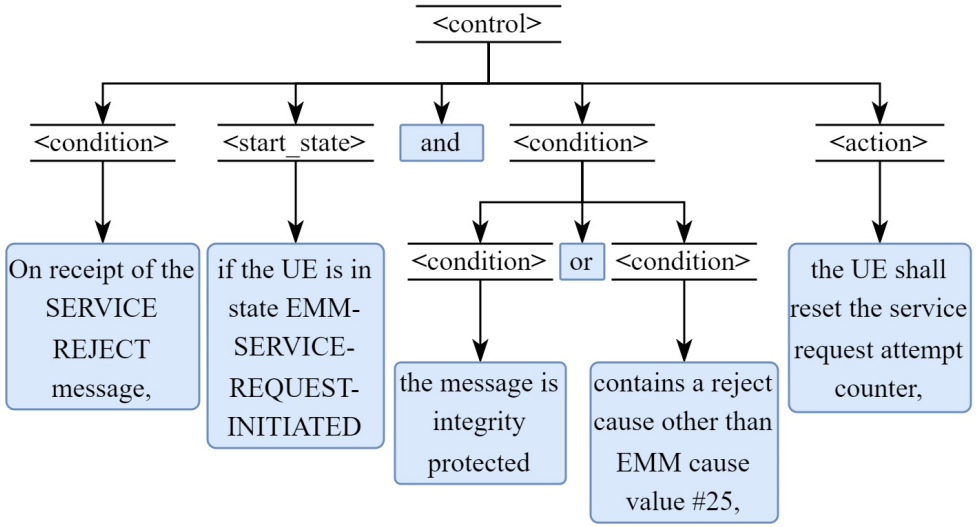
Challenges in NEUTREX

Existing entity tagger frameworks cannot identify complex relations

Develop a constituency parsing framework



Entity Tagger Parsing



Expected Parsing

Challenges in NEUTREX

Existing entity tagger frameworks cannot identify complex relations

Develop a constituency parsing framework

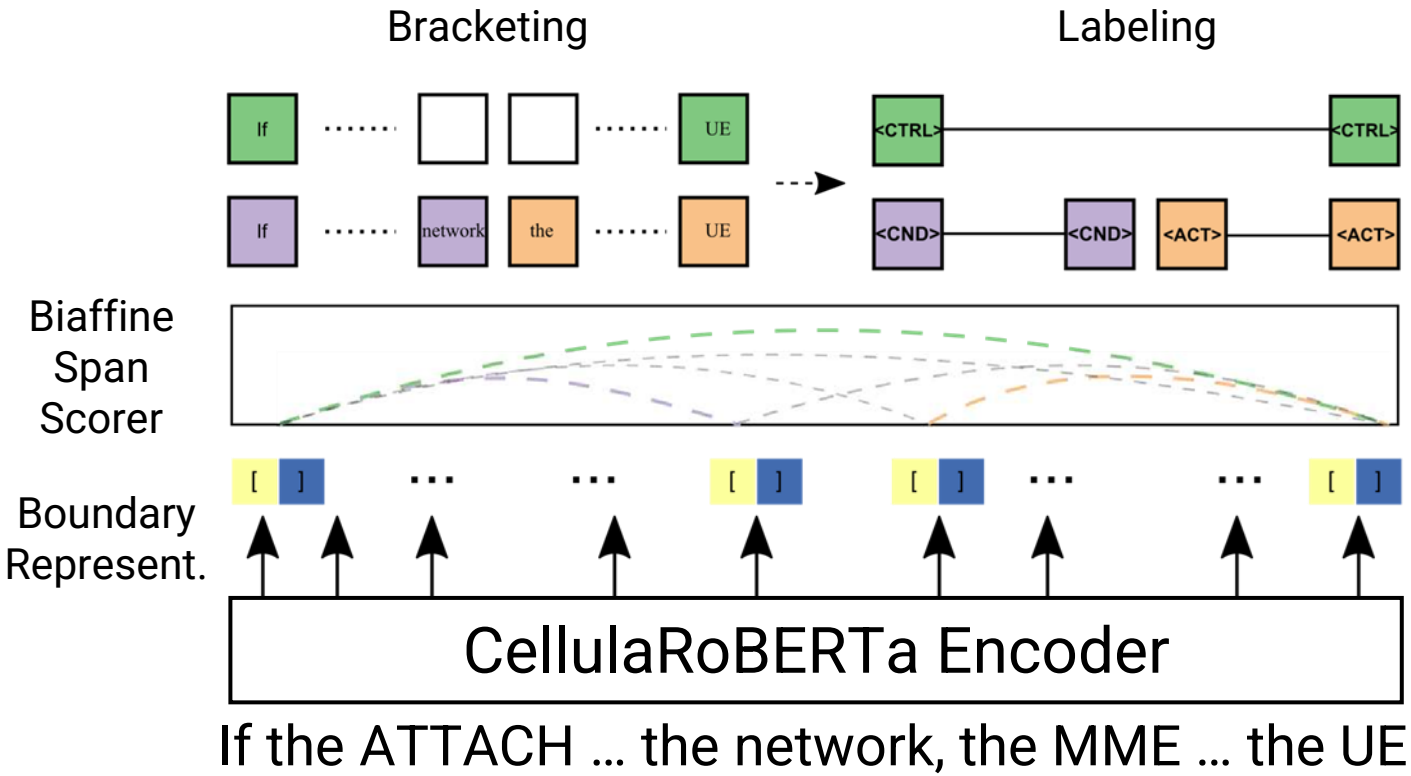
No annotation scheme available for training data

Develop a grammar suitable for cellular data

Existing embedding models cannot understand cellular data

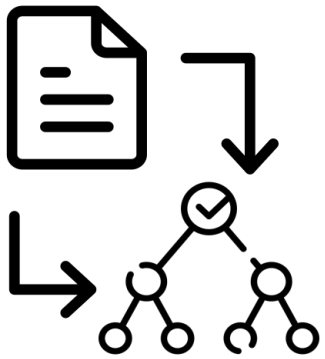
Pretrain an embedding model with cellular data

NEUTREX: Workflow

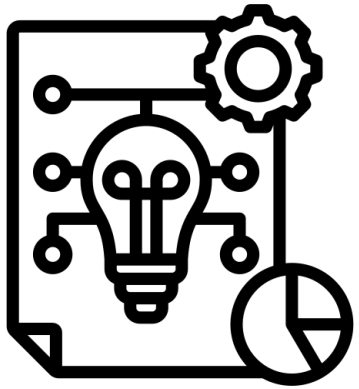


Learning both tasks together is difficult for the constituency parser

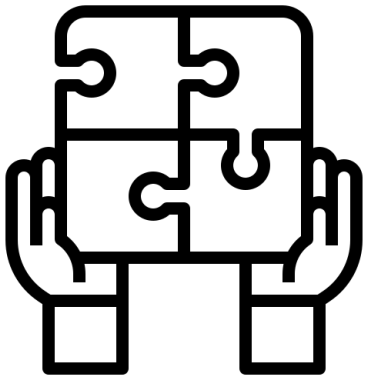
Initially learn labeling, then start bracketing



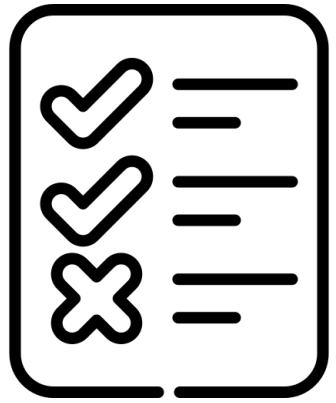
NEUTREX



IRSynthesizer



FSMSynthesizer

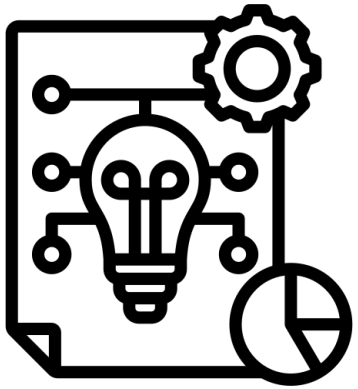


Findings

IRSynthesizer: Overview

Action

The UE shall reset the service request attempt counter



ue_service_req_attempt_counter = 0

Natural Language Transition Component

IRSynthesizer

Intermediate Representation

Challenges in IRSynthesizer

No model or dataset available for NL to logic translation in cellular domain

Design a Domain-Specific Language DSL for translation

```
m:= receive(type, src, msg, dst)
    def m.handleReceive (type, src, msg, dst)
        if type == condition:
            assert σ[chan_src_dst] == msg
        else:
            σ[chan_src_dst] := msg
```

Example DSL Rule

IRSynthesizer: Workflow

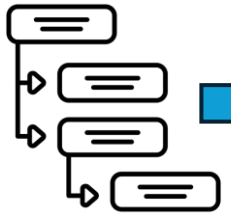
Action

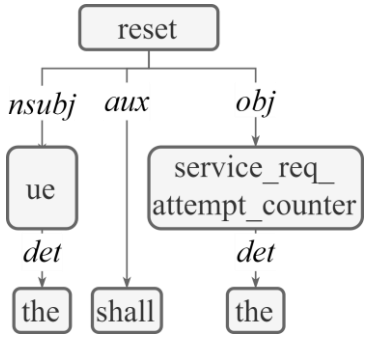
The UE shall reset the service request attempt counter

NL Transition Component

```
c:= reset(type, agent, counter)
    def c.handleReset (type, agent, counter)
        if type == action:
            σ[agent_counter] := 0
```

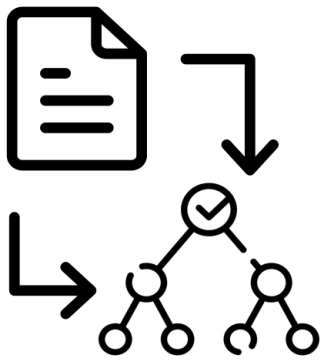
DSL Rule



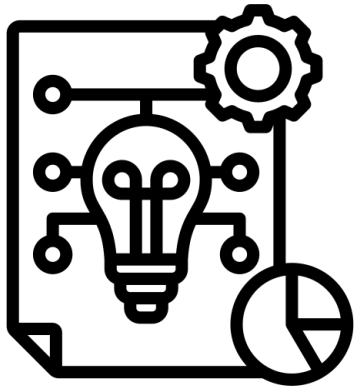


ue_service_
req_attempt
_counter = 0

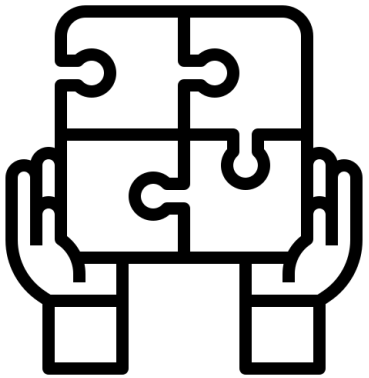
Specification Document | **Keyword Extractor** | **Dependency Tree Generator** | **Dependency Tree** | **IR Translator** | **Logical Formula**



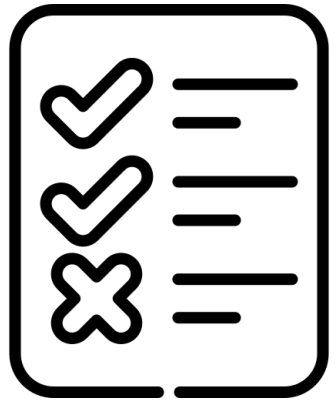
NEUTREX



IRSynthesizer

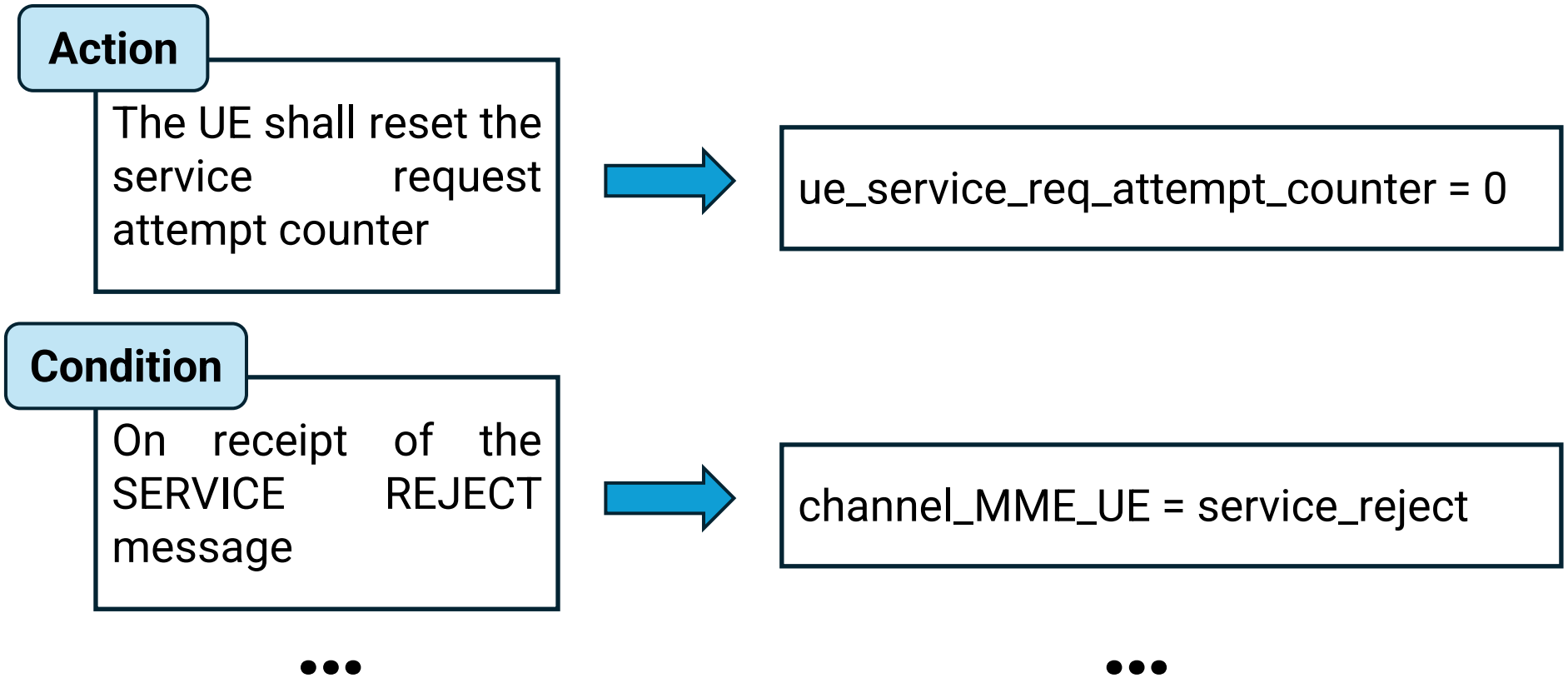


FSMSynthesizer

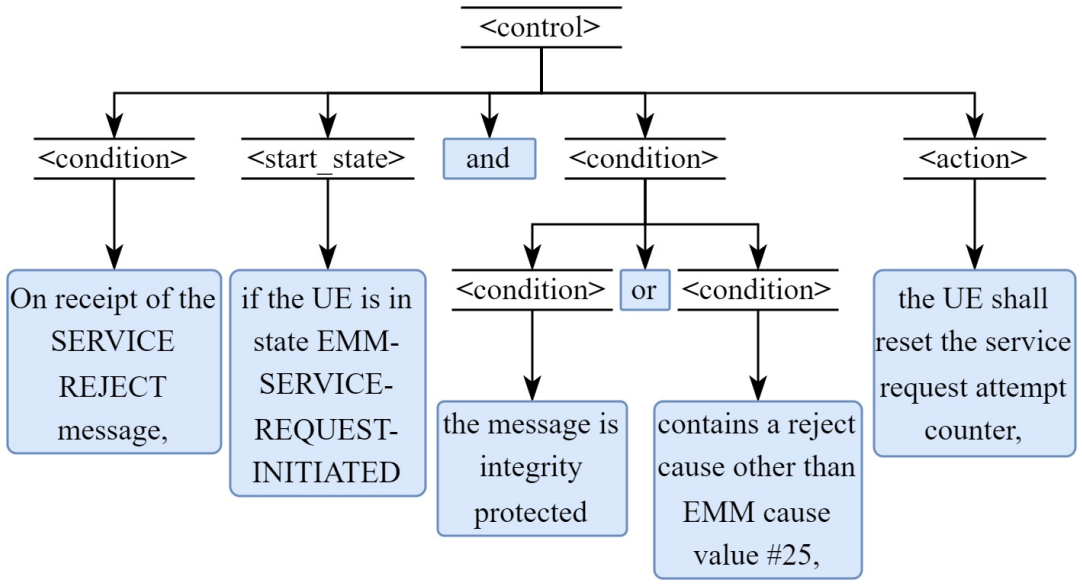


Findings

IRSynthesizer Creates Sperate IR for Transition Components



FSMSynthesizer: Combining Transition Components

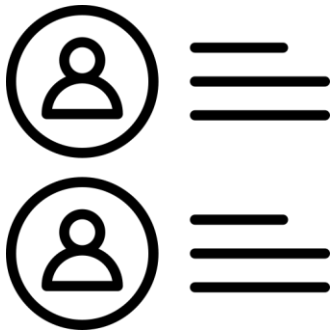


Constituency Parse Tree

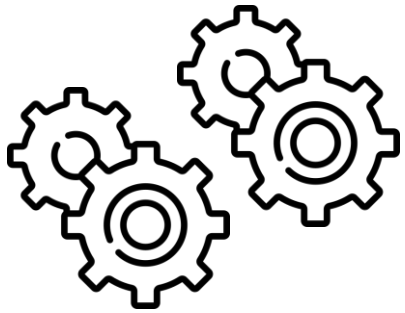
```
start_state := emm_service_request_initiated

condition   :=
    (channel_MME_UE = service_reject) &
    (service_reject_integrity_protected |
    service_reject_emm_cause_present   &
    service_reject_emm_cause_value != cause_25)

action      := ue_service_req_attempt_counter = 0
```

Output Transition

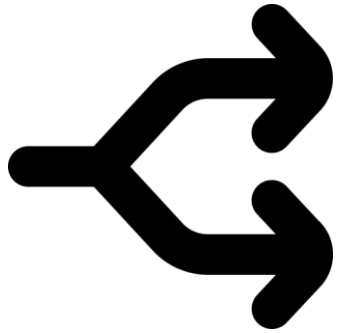
FSMSynthesizer: Compiling and Checking Transitions



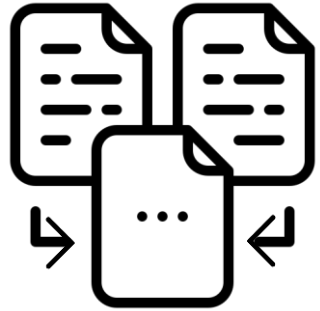
IRSynthesizer Generates Transitions for Multiple Entities



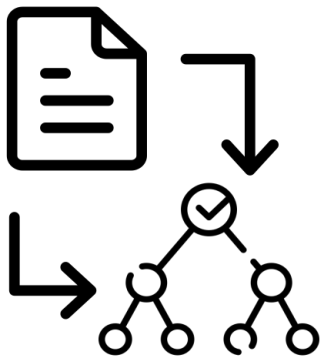
FSMSynthesizer Creates Separate FSMs



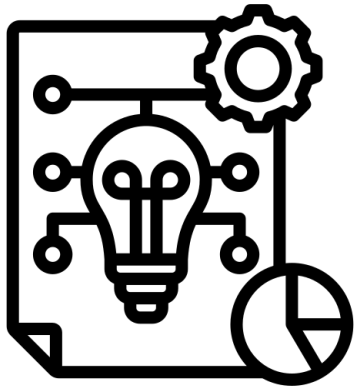
Split Transitions Lead to Inaccurate Security Analysis



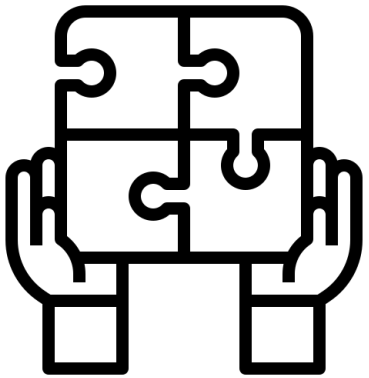
FSMSynthesizer Merges Co-inciding Transitions



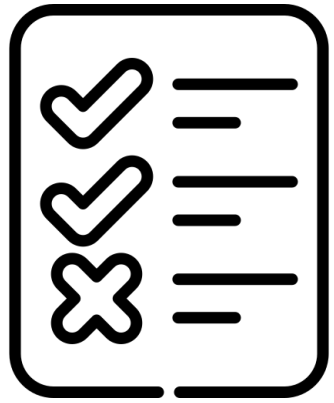
NEUTREX



IRSynthesizer



FSMSynthesizer



Findings

Evaluation of Extracted Transition Components

Evaluation Data	Metric	RFCNLP[1]	Hermes
4G NAS	Unlabeled F-1	39.12	71.33
	Labeled F-1	38.52	68.20
5G NAS	Unlabeled F-1	12.66	67.82
	Labeled F-1	12.54	65.20
5G RRC	Unlabeled F-1	12.01	73.62
	Labeled F-1	10.22	68.69
TCP	Unlabeled F-1	57.43	59.73
	Labeled F-1	47.76	57.06
DCCP	Unlabeled F-1	38.88	56.71
	Labeled F-1	33.91	55.06

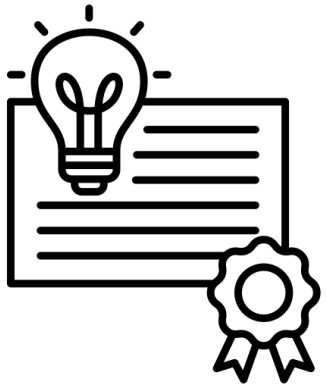
[1]Pacheco, Maria Leonor, et al. "Automated attack synthesis by extracting finite state machines from protocol specification documents." *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022.

Evaluation of Extracted FSM

Protocol	Accuracy of IRSynthesizer and FSMSynthesizer		Accuracy of Hermes	
	Action	Condition	Action	Condition
4G	92.23	92.24	81.14	87.21
5G	93.86	94.45	81.39	86.40

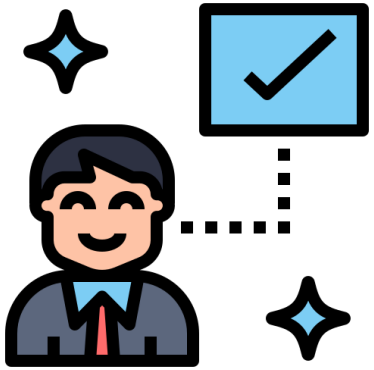
Application in Security Analysis

Translate FSM to Formal Model



Check Security Properties

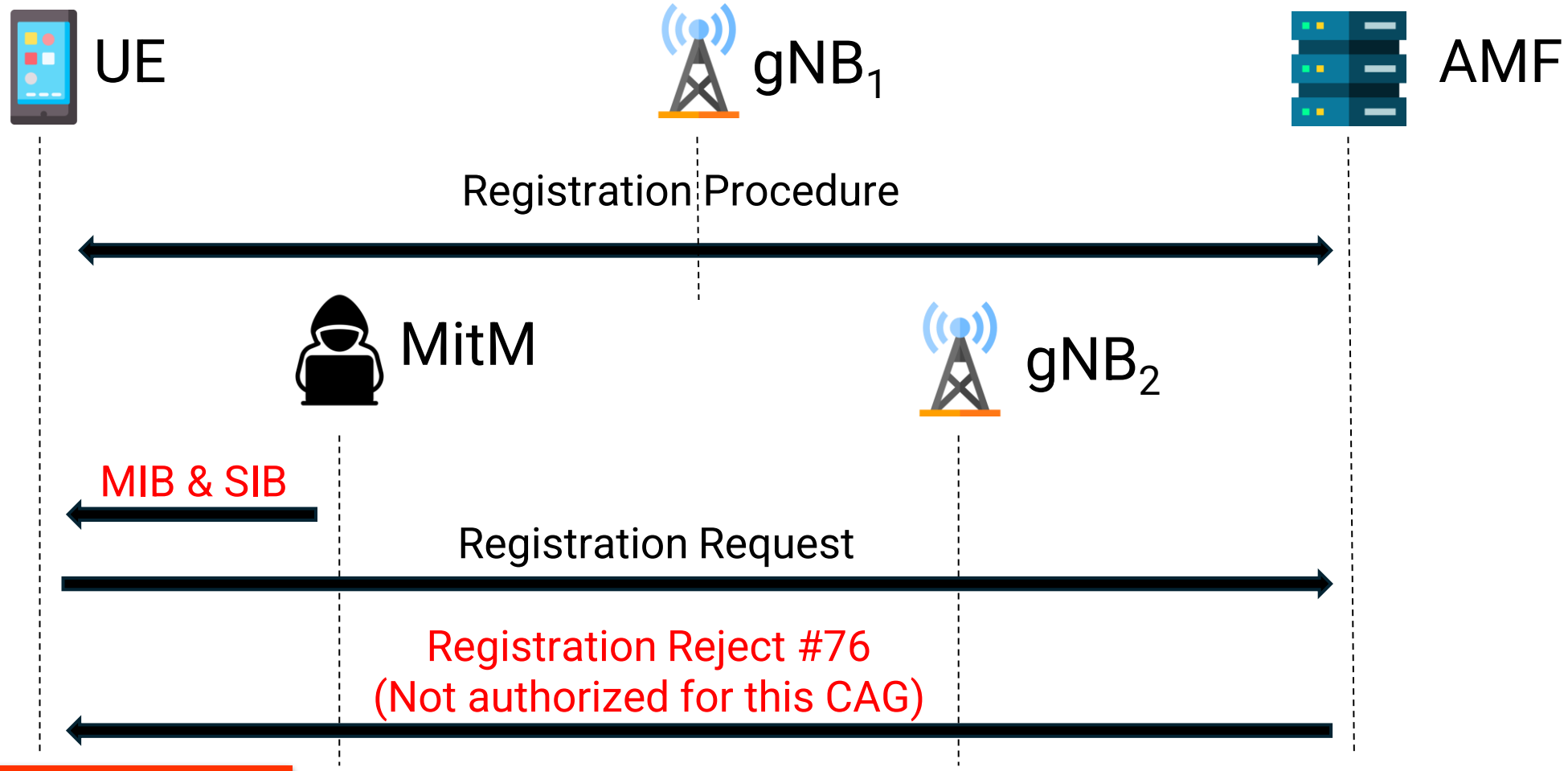
Identify 19 Previous and 3 New Vulnerabilities



Acknowledgement from GSMA

GSMA™	CVD-2023	0071	Abdullah Al Ishtiaq, Kai Tu, Syed Md Mukit Rashid, Syed Rafiul Hussain	Pennsylvania State University

Deletion of Allowed CAG List



- **Delete Allowed CAG List**
- **Cannot connect to network**

Summary

Automated FSM extraction framework

State-of-the-art in transition component extraction

3 new and 19 previous vulnerabilities in 4G and 5G

Released annotated datasets and trained models

GitHub: https://github.com/SyNSec-den/hermes-spec-to-fsm

Hermes: Unlocking Security Analysis of Cellular Network Protocols by Synthesizing Finite State Machines from Natural Language Specifications

Abdullah Al Ishtiaq

Department of Computer Science and Engineering

Pennsylvania State University

Website: abdullahishtiaq.net

GitHub: github.com/ishtiaqniloy