FORMAL ANALYSIS OF SECURITY PROCEDURES IN LTE

Noamen Ben Henda, Karl Norrman
Ericsson Research

OUTLINE

- Motivation
- Background
- LTE Security
- ProVerif Modeling
- Verification Results
- Conclusion

MOTIVATION

- Massively deployed wide variety of Telecom protocols
- Design errors detected after deployment are difficult and expensive to correct
- Standardization process could greatly benefit from use of formal verification
- Very active research area in academia leveraging many well established and supported tools

OUTLINE

- Motivation
- Background
- LTE Security
- ProVerif Modeling
- Verification Results
- Conclusion

SECURITY PROTOCOLS

- Security protocols are procedures based on message exchange between agents (peers) letting them share secrets over a public network
- They are intended to perform correctly even in the presence of a malicious intruder (attacker)
- Correctness requirements include secrecy and authenticity

ATTACKER MODEL

- The symbolic Dolev-Yao model: Full control over communication medium and perfect cryptography
- Ability to intercept all messages, forward, drop or replay old messages
- Cannot decrypt messages unless in possession of required keys

DIFFICULTIES

- *Unboundedness:*
  - prove correctness regardless of the number of agents and runs
- *Other sources of infiniteness:*
  - handle timestamps, counters, etc.
- *Scalability:*
  - handle large protocol models (more than just two agents) which are typical in Telecom networks
- *Usability:*
  - What can we do if we are stuck (unable to prove correctness)?
- *Testing:*
  - Wait for implementation? Simulate attacker?

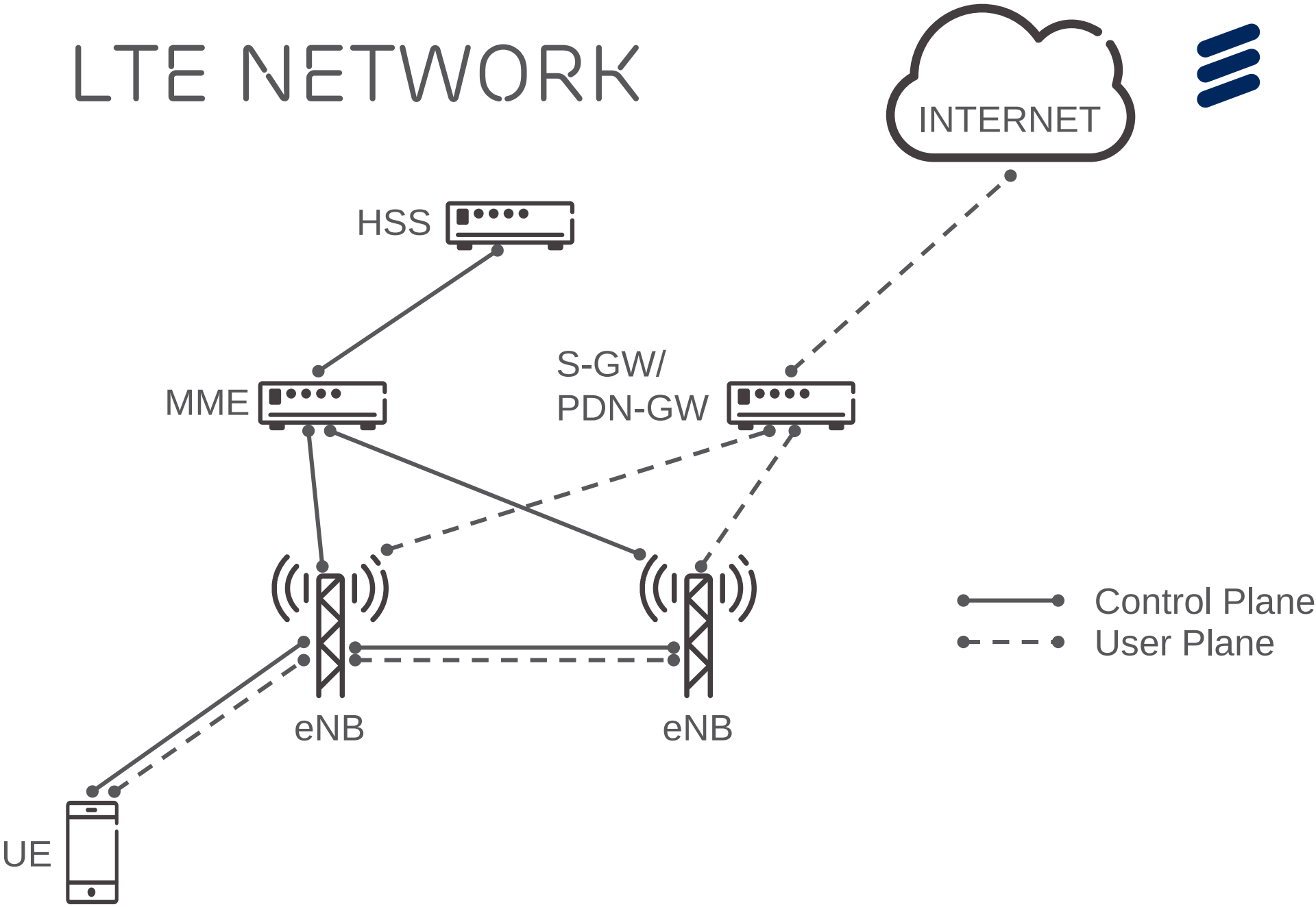
LTE FEASIBILITY STUDY

- Considered different security procedures in LTE (most of them never analyzed in this manner)
- Chosen one of the academic tools (ProVerif) offering a good compromise for ease-of-use and expressiveness of input language
- Used the tool to model and verify secrecy and different authentication properties

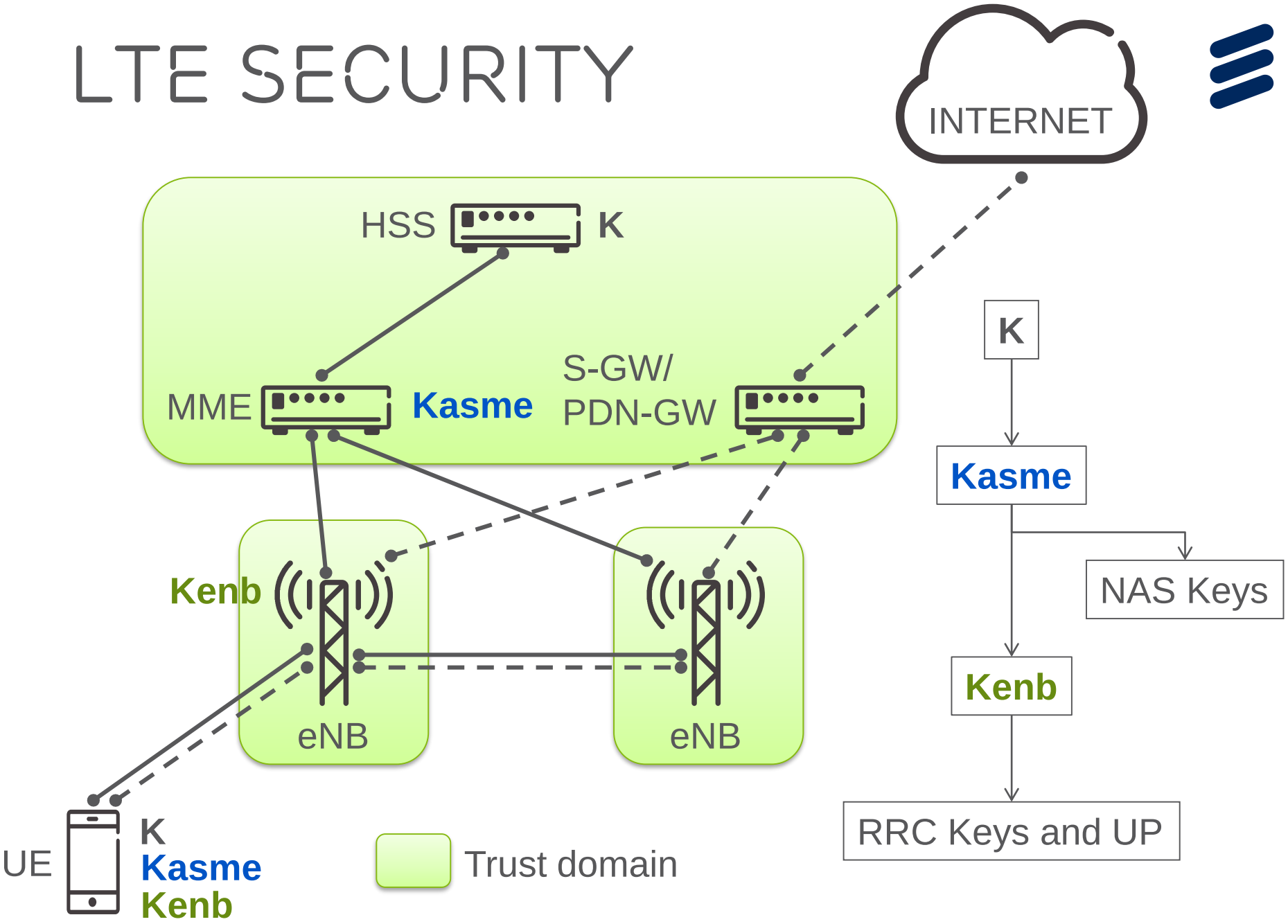
OUTLINE

- Motivation
- Background
- LTE Security
- ProVerif Modeling
- Verification Results
- Conclusion

LTE NETWORK


INTERNET
HSS
MME
S-GW/
PDN-GW
eNB
eNB
UE
Control Plane
User Plane

LTE SECURITY



OUTLINE

- Motivation
- Background
- LTE Security
- ProVerif Modeling
- Verification Results
- Conclusion

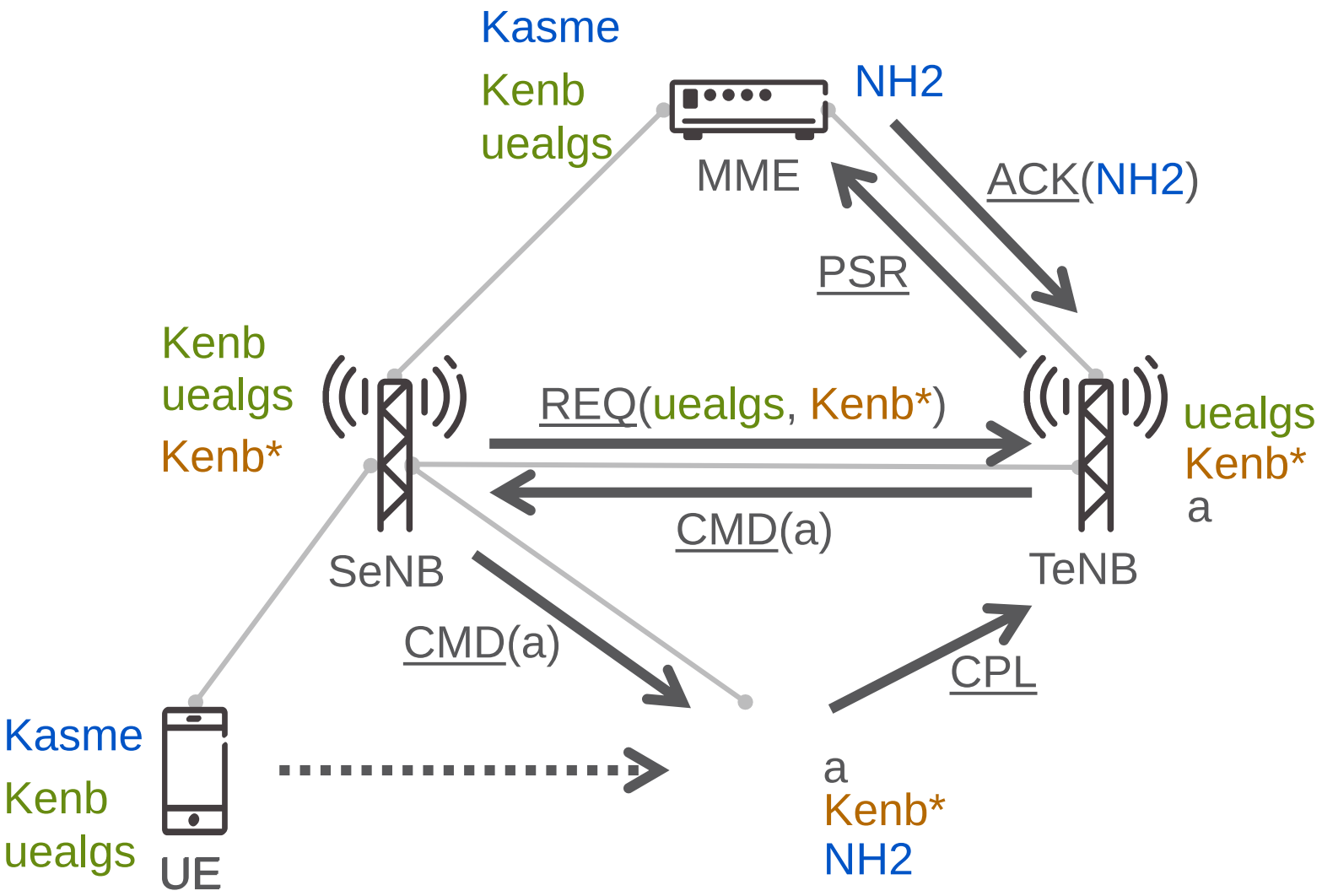
PROVERIF LANGUAGE

- Typed variant of the pi calculus
- Messages are terms and cryptographic primitives are rewrite rules
- Processes are sequence of events
- Security properties are assertions.

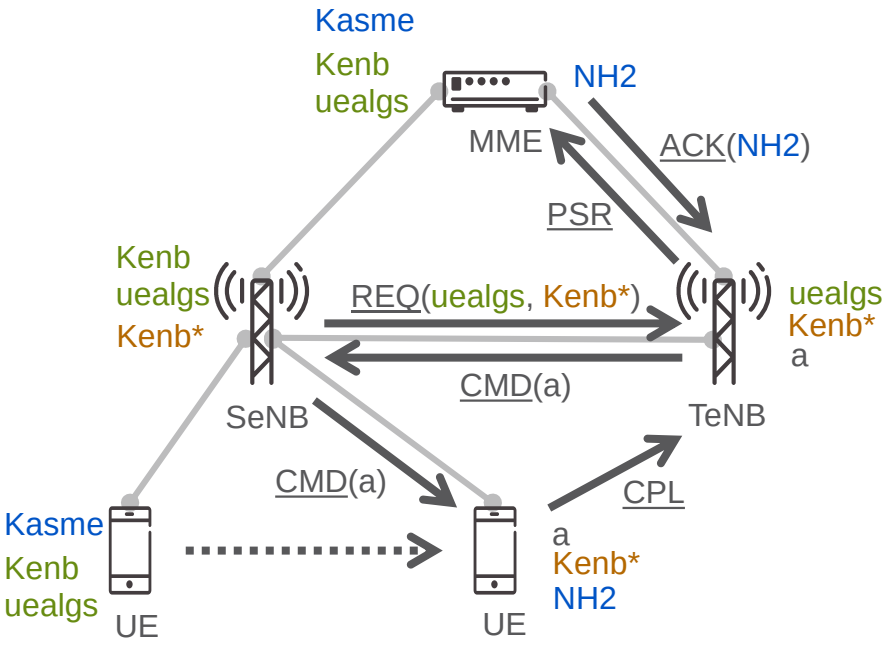
PROVERIF MODEL

- *Declarations*
  - User types
  - Communication channels
  - Constants
  - Cryptographic primitives
  - Queries
- *Process Macros*
  - Parameterized process definition
  - List of events (send event, receive event, etc.)
- *Main Process*
  - Declarations
  - Process macro instantiation

X2-HANDOVER

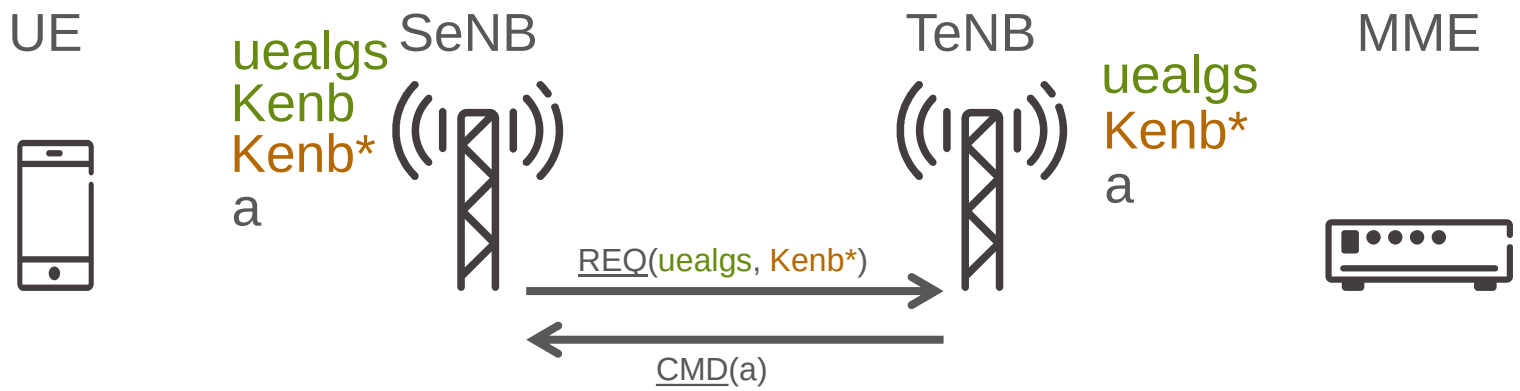

Kasme
Kenb
uealgs
MME
NH2
ACK(NH2)
PSR
Kenb
uealgs
Kenb*
REQ(uealgs, Kenb*)
uealgs
Kenb*
a
CMD(a)
SeNB
TeNB
CMD(a)
CPL
Kasme
Kenb
uealgs
UE
a
Kenb*
NH2

DECLARATIONS



- **free** pubch: **channel**.
- **free** secch: **channel** [**private**].
- **type** key.
- **type** alg.
- **type** algs.
- **fun** consset(alg, algs): algs [**data**].
- **type** msgheader.
- **const** REQ: msgheader.
- **const** CMD: msgheader.
- **fun** psenc(alg, **bitstring**, key): **bitstring**.

PROCESS MACROS



› SeNB(uealgs: caps, Kenb: key, cellid: **bitstring**) =

let Kenb*:key = kdf(cellid, Kenb) **in**

out(secch, (REQ, Kenb*, uealgs));

in(secch, (=CMD, a: alg));

…

› TeNB() =

in(secch, (=REQ, Kenb*: key, uealgs: caps));

let a: alg **suchthat** mem(a, uealgs) **in**

out(secch, (CMD, a));

…

MAIN PROCESS

› **process**

(---------- context setup ---------- *)*

new a1: alg; new a2: alg;

let uealgs = consset(a1, consset(a2, emptyset)) **in out**(pubch, uealgs);

new Kasme: key; **new** nasulcount: **bitstring**; **out**(pubch, nasulcount);

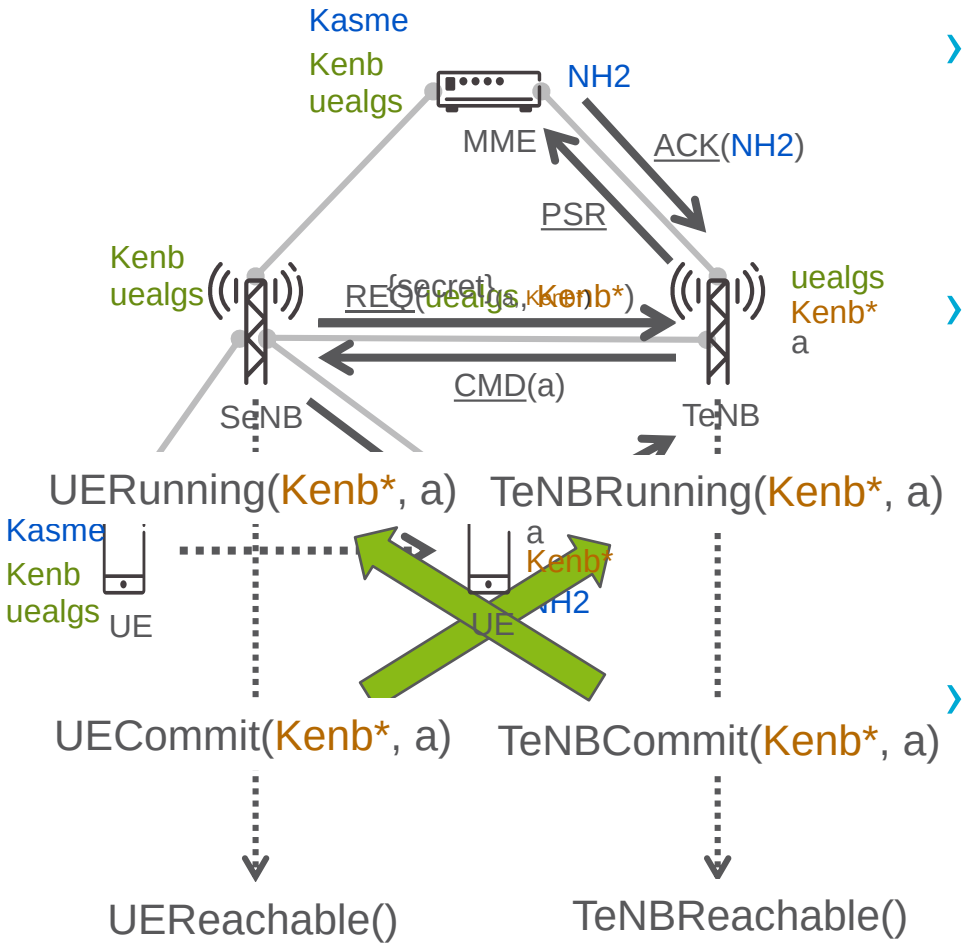
let Kenb: key = kdf(nasulcount, Kasme) **in new** cellid: bitstring;

out(pubch, cellid);

(---------- instantiation ---------- *)*

!UE(uealgs, Kasme, Kenb, cellid) **|** **!**SeNB(uealgs, Kenb, cellid) **|** **!**TeNB() **|** **!**MME(uealgs, Kasme , Kenb)

SECURITY PROPERTIES



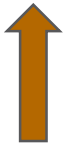
- *Secrecy*
  - **query attacker**(secret).
- *Authentication*
  - **query** a: alg, k: key; **event**(TeNBCommit(a, k)) ==> **event**(UERunning(a, k)).
- *Sanity*
  - **query** event(UEReachable()).

OUTLINE

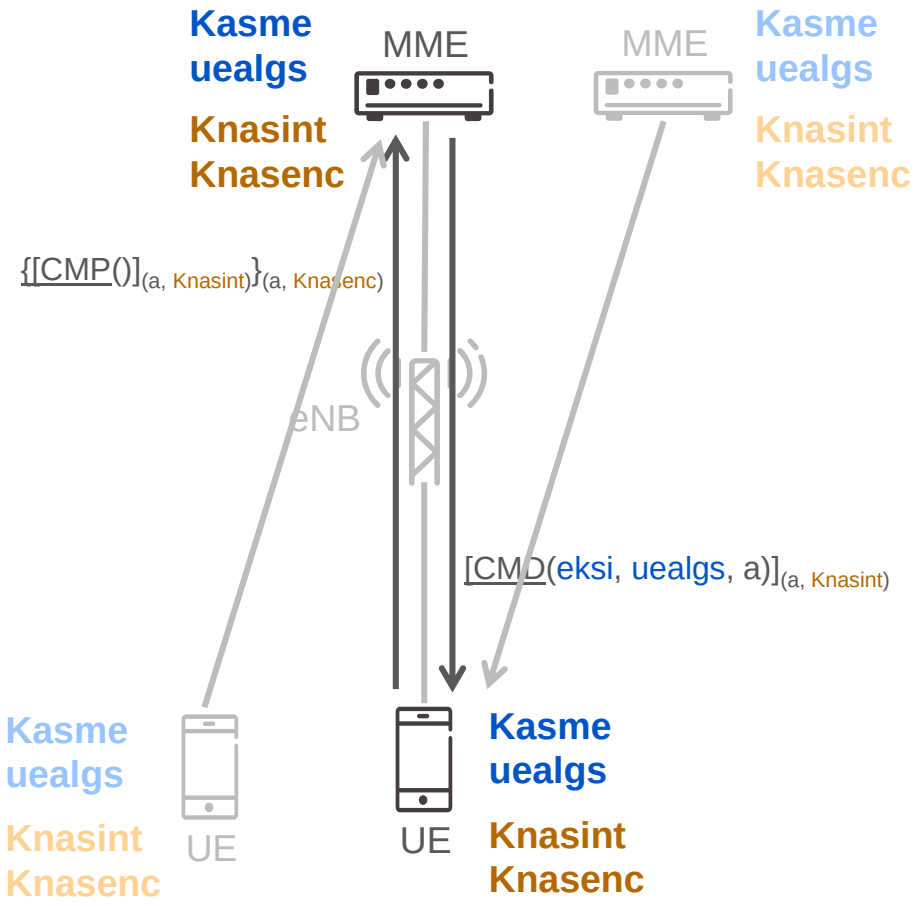
- Motivation
- Background
- LTE Security
- ProVerif Modeling
- Verification Results
- Conclusion

RESULTS

Property	AKA	NAS SMC	RRC SMC	X2 HO	S1 HO
Secrecy	TRUE	TRUE	TRUE	TRUE	TRUE
Weak auth. Node to UE	TRUE	TRUE	TRUE	TRUE	TRUE
Weak auth. UE to Node	TRUE	TRUE	TRUE	TRUE	TRUE
Strong auth. Node to UE	TRUE	False	TRUE	TRUE	?
Strong auth. UE to Node	?	False	?	?	?



FALSIFIED PROPERTIES



- Parallel sessions by honest agents are possible
- Can be mitigated by a transaction identifier
- Is strong authentication a requirement?

OUTLINE

- Motivation
- Background
- LTE Security
- ProVerif Modeling
- Verification Results
- Conclusion

EVALUATION

› *Upside*

- Better understanding of the design
- Competence development
- Increased assurance
- Formal specifications (models)

› *Downside*

- Model protocol in isolation
- Limited modeling capabilities (stateless)
- Difficult to handle non-termination
- Too powerful attacker model

CONTINUATION

- Analyze remaining parts of LTE: other types of handovers
- State of the art evaluation: Tool survey and comparison
- Looking into statefull verification: StatVerif, Tamarin, etc

WHY STATEFULL

- Keys are derived from higher level keys and other parameter of the systems
- The parameters used in the key derivation must always be unique to avoid key-stream re-use
- The parameters can be state related (identities, counters, sequence numbers, etc.)

ERICSSON