Formal Verification

of the Security for Dual Connectivity in LTE

Noamen Ben Henda, Karl Norrman, Katharina Pfeffer
Ericsson Research Security, Sweden

outline

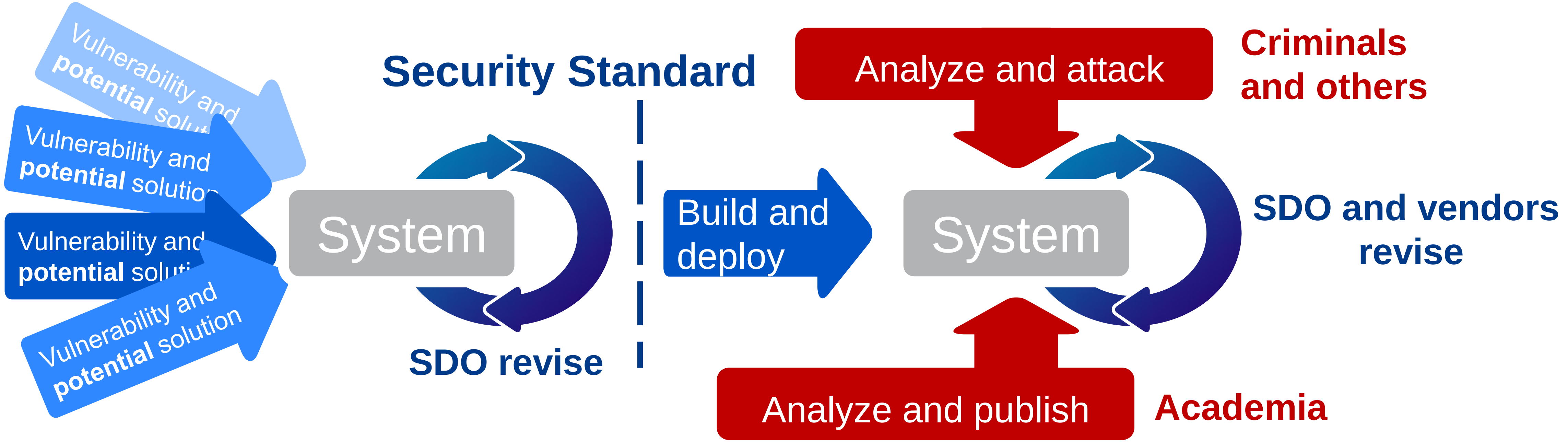
- Motivation
- Formal Verification of Security Protocols
- Dual Connectivity (DC)
- DC Modeling
- Results and Conclusion

outline

- **Motivation**
- Formal Verification of Security Protocols
- Dual Connectivity (DC)
- DC Modeling
- Results and Conclusion

motivation

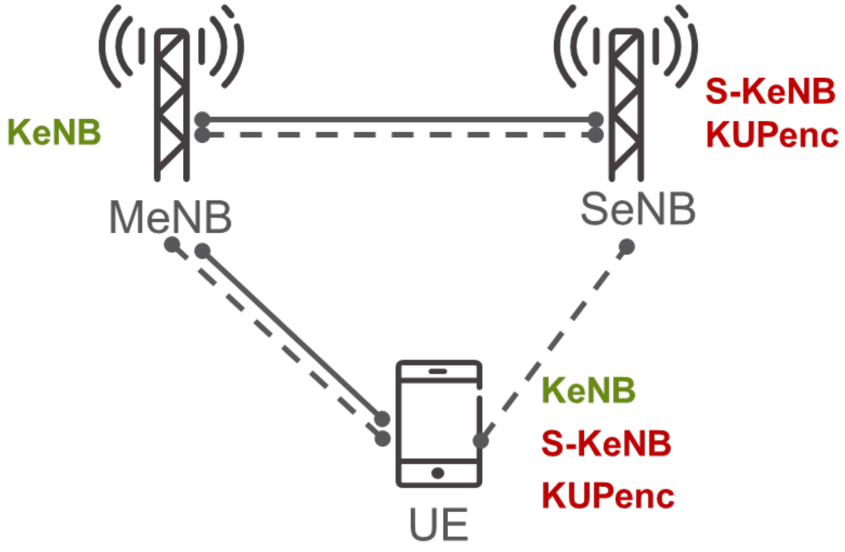
- Massively deployed Telecom protocols, design errors after deployment are **difficult and expensive to correct**


Vulnerability and potential solution
Vulnerability and potential solution
Vulnerability and potential solution
Vulnerability and potential solution
Security Standard
System
SDO revise
Build and deploy
Analyze and attack
Criminals and others
System
SDO and vendors revise
Analyze and publish
Academia

- Active research in academia
- **Usage in standardization still limited**

AIM of our work

- Evaluate **applicability** of formal verification tools for standardization of security protocols
  - Expressiveness
  - Usability
  - Performance



- Verify security of one selected feature
  - Dual Connectivity (DC)

- Formal verification of DC with three state-of-the-art academic tools:
  - Scyther, Tamarin, ProVerif

outline

- › Motivation
- › **Formal Verification of Security Protocols**
- › Dual Connectivity (DC)
- › DC Modeling
- › Results and Conclusion

Formal verification of Security protocols

- **Security protocols:**
  - procedures based on **message exchange** between **agents**
  - let agents share secrets over a public network
  - intended to perform correctly even in the presence of a **malicious intruder** (attacker)

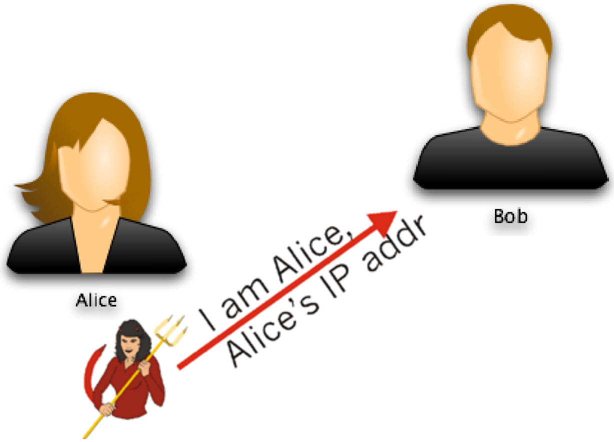
I am Alice,
Alice's IP addr
Alice
Bob
Trudy

  - rely heavily on **cryptographic primitives**

Attacker model

- In the **Symbolic Dolev-Yao Model** the attacker
  - has full control over communication medium
    - ability to intercept all messages, forward, drop or replay old messages
  - cannot decrypt messages unless in possession of required keys

Alice
I am Alice,
Alice's IP addr
Bob
Trudy



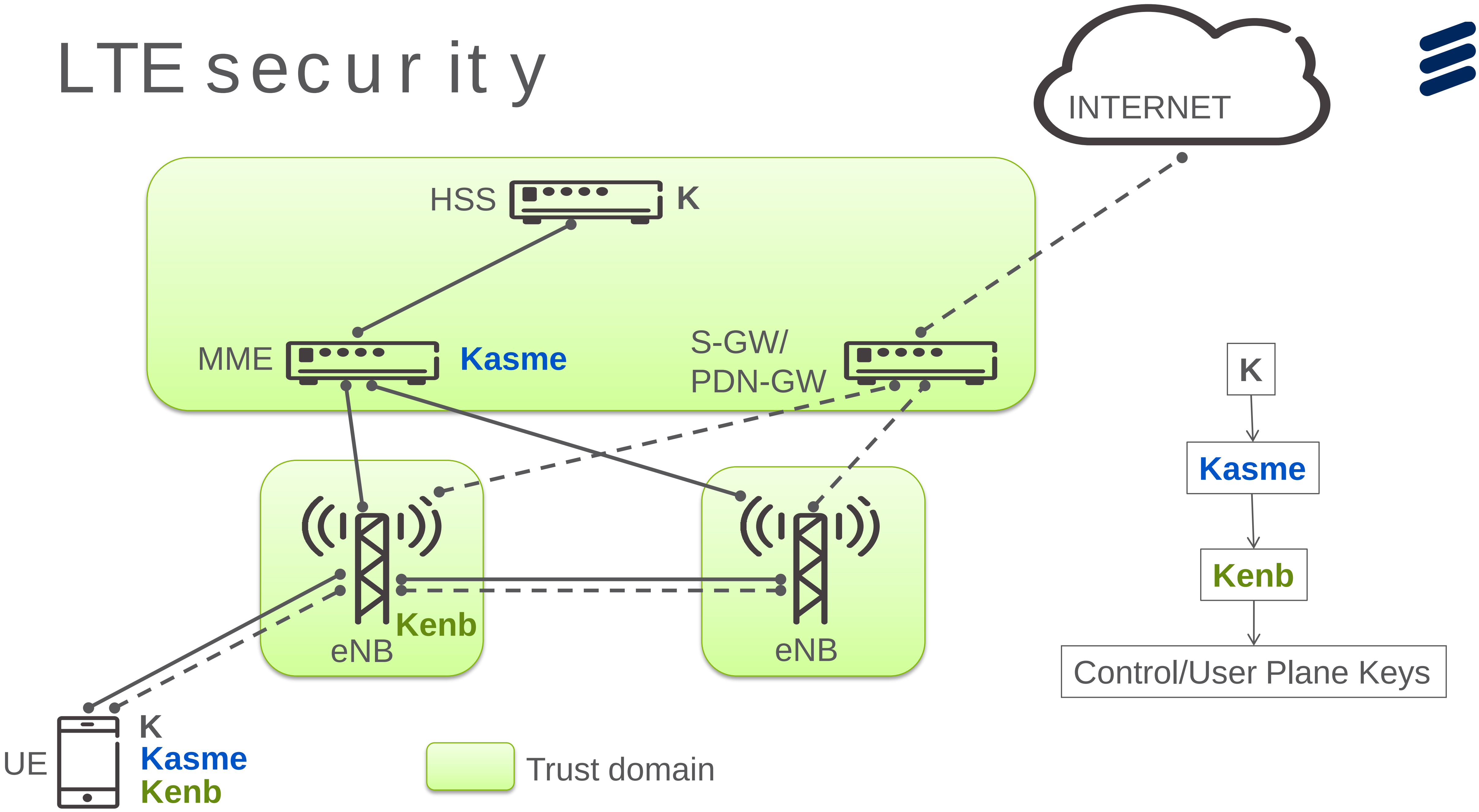
Security properties

- Key establishment security properties:
  - **Agreement** (involved agents obtain same parameter/s, e.g. key)
  - **Secrecy** (no other than the involved agents obtains key)
  - **Freshness** (prevents key re-use)
- Aim: Proof that security properties hold for unbounded number of agents and protocol runs

outline

- Motivation
- Formal Verification of Security Protocols
- **Dual Connectivity (DC)**
- DC Modeling
- Results and Conclusion

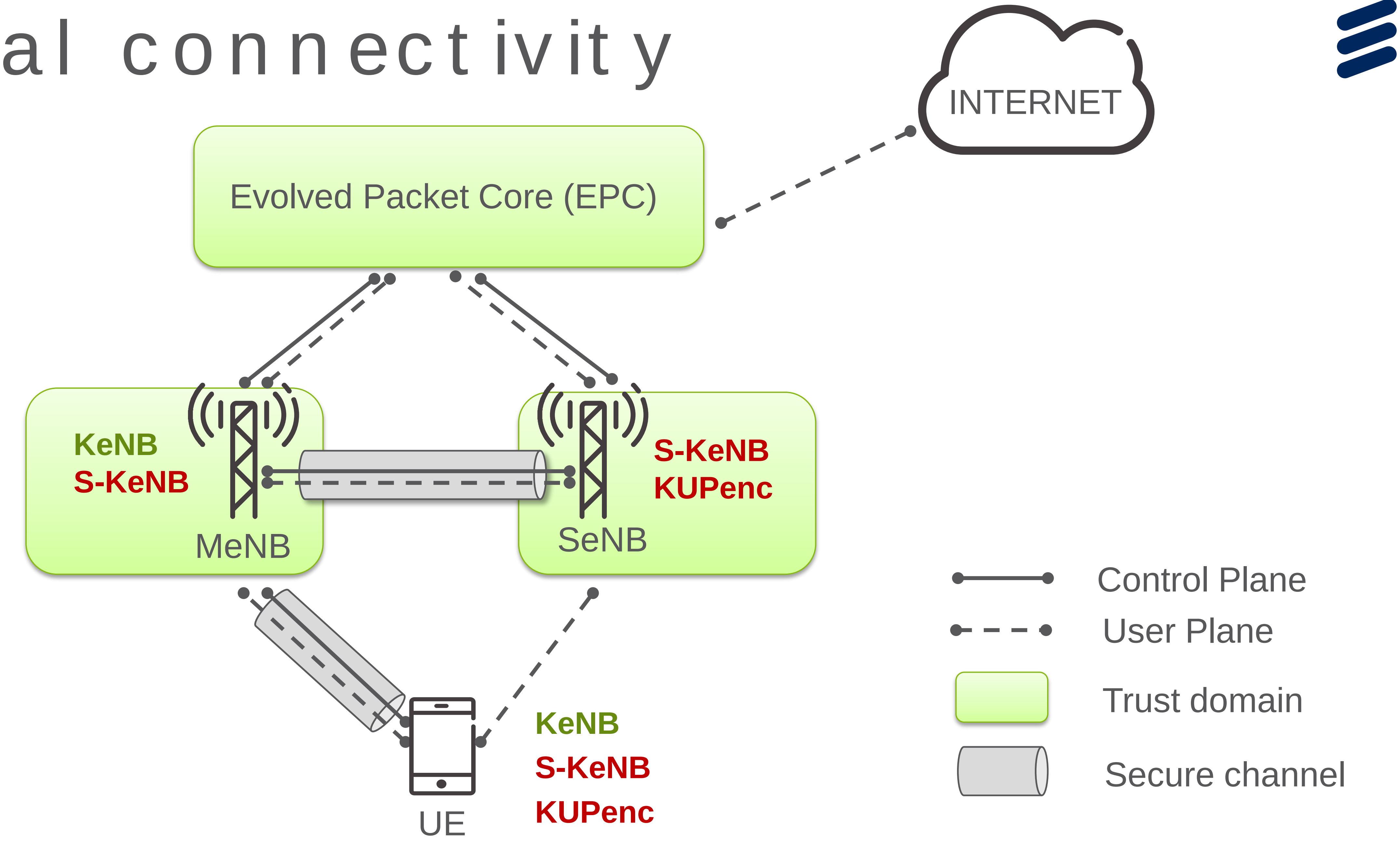
LTE security



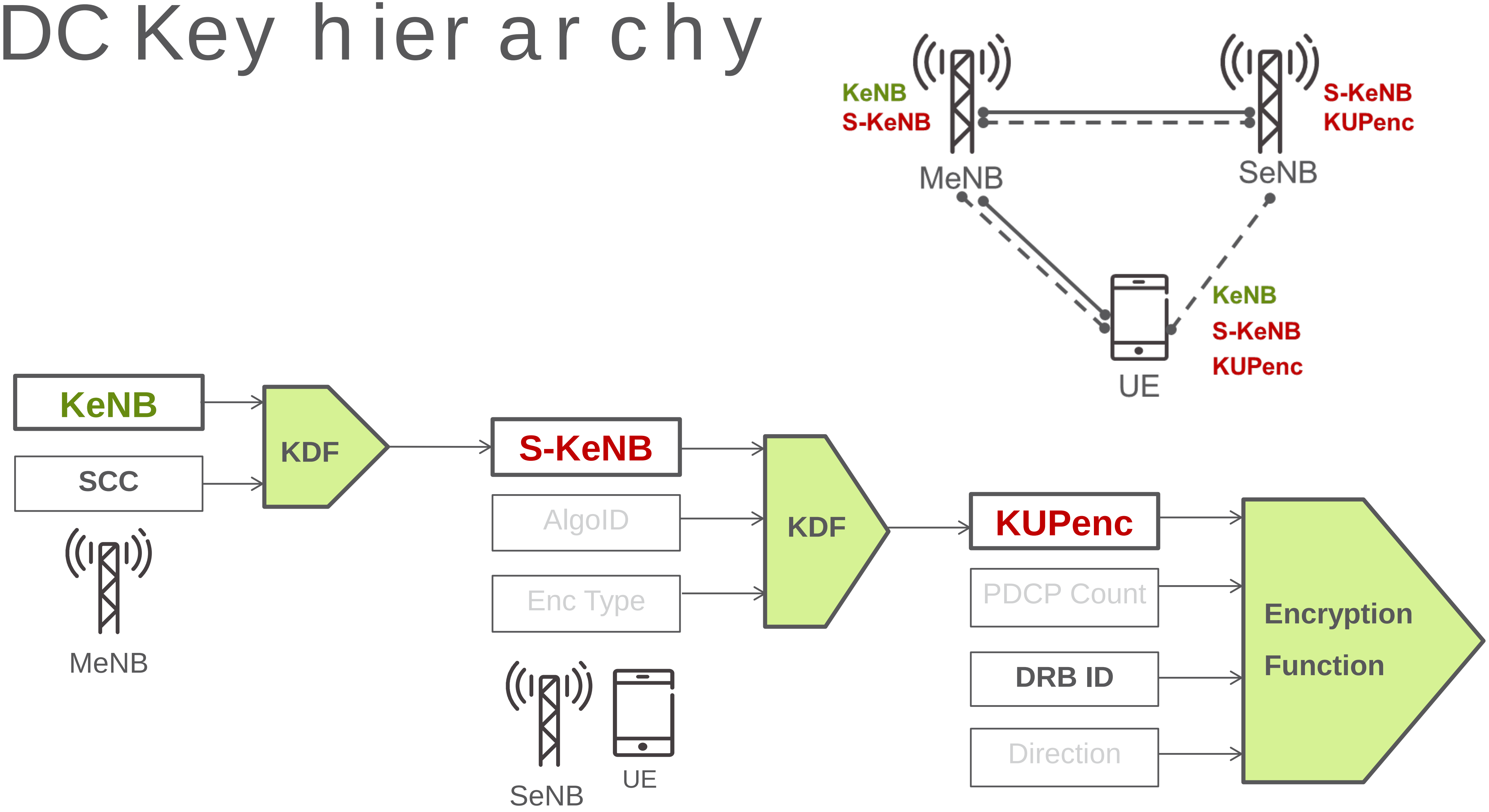
K → Kasme → Kenb → Control/User Plane Keys

🟩 Trust domain

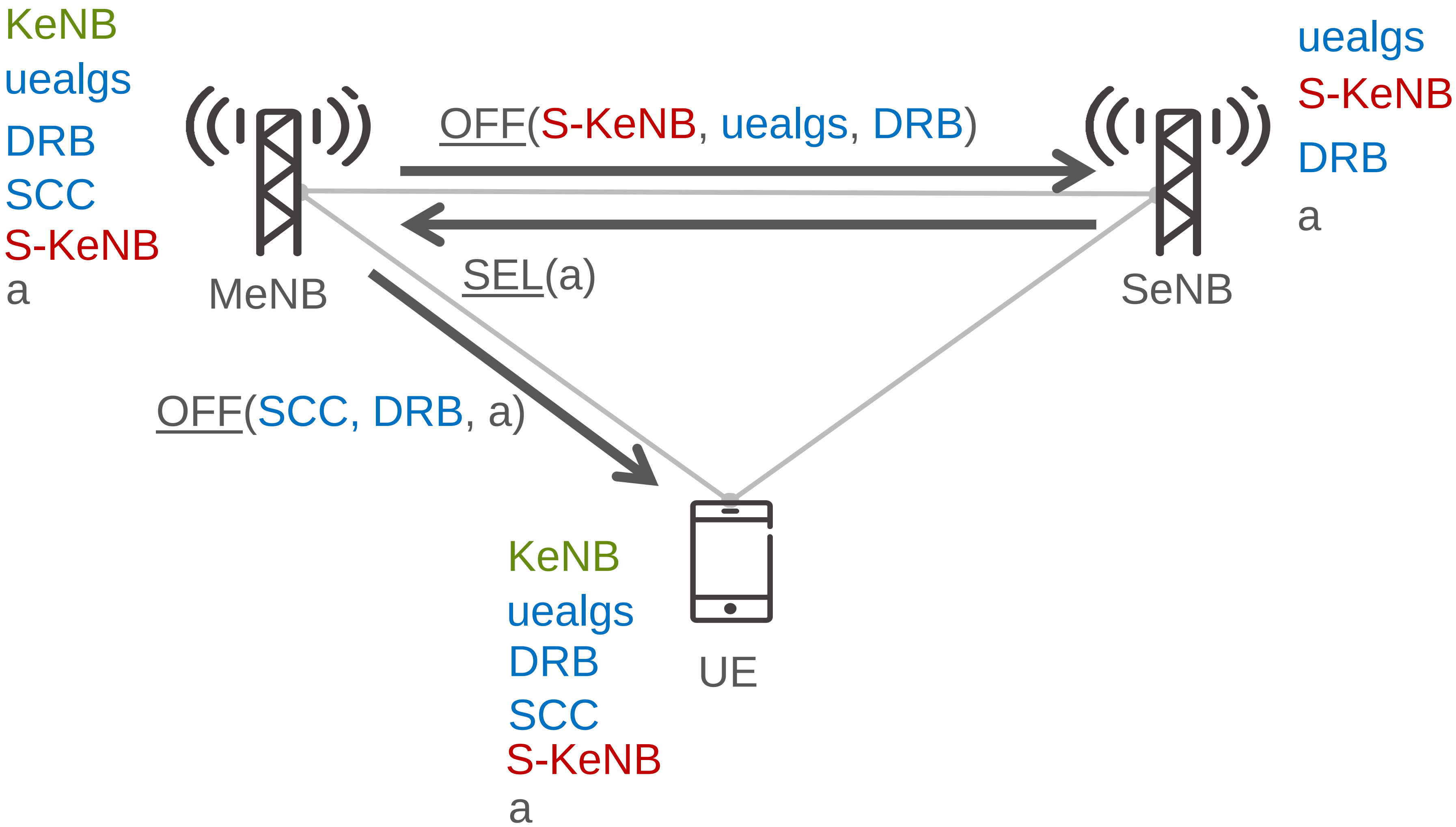
Dual connectivity



DC Key hierarchy



Dual Connectivity
initial offload



outline

- Motivation
- Formal Verification of Security Protocols
- Dual Connectivity (DC)
- **DC Modeling**
- Results and Conclusion

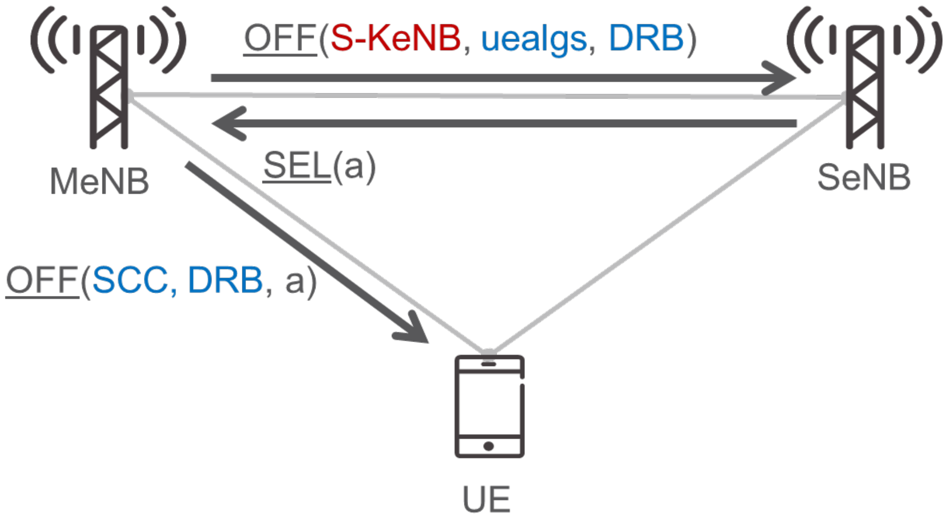
DC modeling

- Used automatic model checking tools: Scyther, Tamarin, ProVerif

- Different input languages and abstraction levels

- Goal:
  - verify **secrecy** and **freshness** of KUPenc
  - verify **agreement** on KUPenc and algorithm between terminal and SeNB

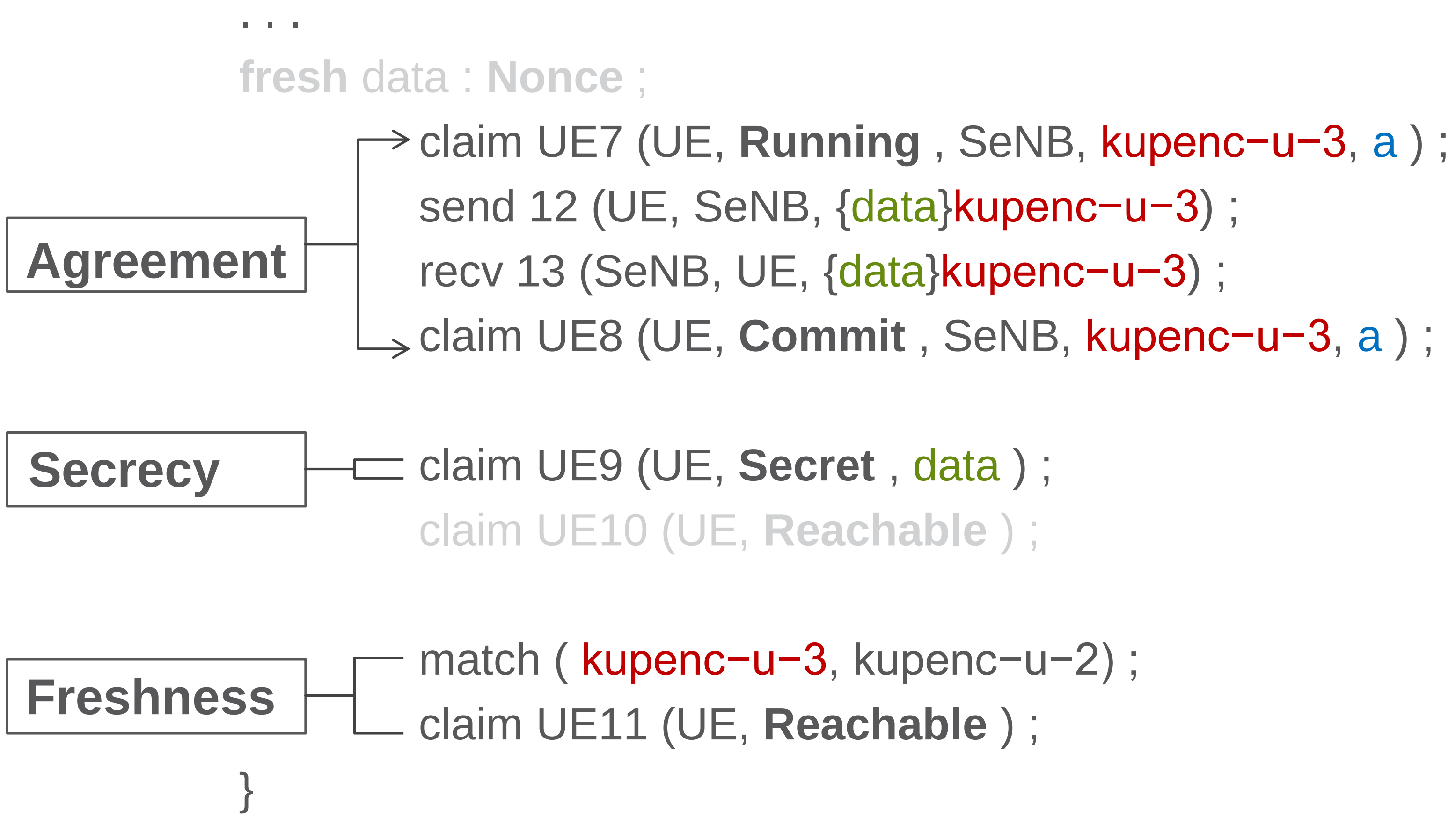
scyther



```
protocol dc (MeNB, SeNB, UE) {
          role MeNB {
                    var a : Alg ;
                    macro skenb−m−1 = kdf ( k (MeNB, UE) , scc−1) ;
                    send 1 (MeNB, SeNB, {skenb−m−1, ( alg−1, alg−2) , drb−1}k (MeNB, SeNB) ) ;
                    recv 2 (SeNB, MeNB, {a}k (MeNB, SeNB) ) ;
                    send 3 (MeNB, UE, {scc−1, a , drb−1}k (MeNB, UE) ) ;
                    claim MeNB1 (MeNB, Reachable ) ;
                    …
          }
          role SeNB { …
```

Initial Offload

scyther



Agreement

```
. . .
fresh data : Nonce ;
    claim UE7 (UE, Running , SeNB, kupenc−u−3, a ) ;
    send 12 (UE, SeNB, {data}kupenc−u−3) ;
    recv 13 (SeNB, UE, {data}kupenc−u−3) ;
    claim UE8 (UE, Commit , SeNB, kupenc−u−3, a ) ;
```

Secrecy

```
    claim UE9 (UE, Secret , data ) ;
    claim UE10 (UE, Reachable ) ;
```

Freshness

```
    match ( kupenc−u−3, kupenc−u−2) ;
    claim UE11 (UE, Reachable ) ;
}
```

outline

- Motivation
- Formal Verification of Security Protocols
- Dual Connectivity (DC)
- DC Modeling
- **Results and Conclusion**

Results and tool evaluation

- **Scyther** showed several restrictions while trying to model DC.
  - No support for modeling
    - sets/lists
    - control flows (loops, conditionals)
    - secure channels
    - choice

- **Tamarin** supports modeling of sets, control flows and choice
  - No support for secure channels

- **ProVerif** supports modeling of sets, choice and secure channels
  - No support for control flows (i.e. counters)

Results and tool evaluation

Tool	Scyther	Tamarin	ProVerif
Secrecy	+	+	++
Freshness	+	+	-
Agreement	-	-	++

Tool	Scyther	Tamarin	ProVerif
Usability	++	+	+
Expressiveness	-	++	-
Performance	+	-	++

conclusion

- Our initial goal was unbounded verification of the security properties secrecy, agreement, and key freshness.

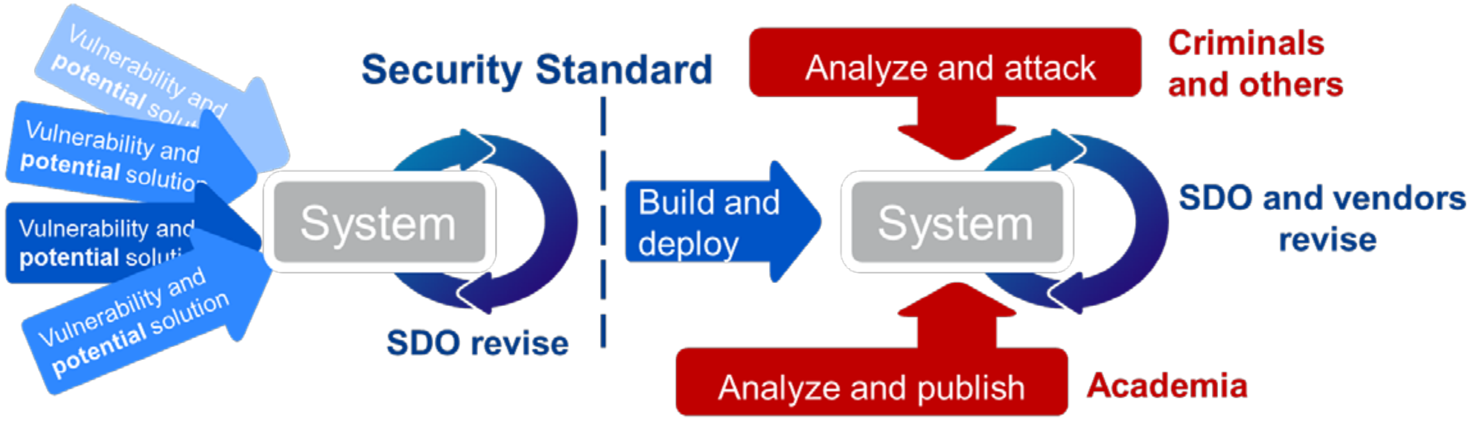
- None of the tools could verify **freshness** in the unbounded model
  - either modeling of required features was not supported or the tool did not terminate

- None of the tools alone provides full support for all the required features
  - combination possible, but not enough

Applicability during standardization

- › Modeling low level details and state changes during runs is often not supported.

- › Process of formal modeling can enrich standardization process.
  - – Reflect on design choices
  - – Formulate security goals



ERICSSON