

Formal Analysis of 5G EAP-TLS Authentication Protocol Using Proverif

Published in: [IEEE Access](#) (Volume: 8) at 27 January 2020

Summarized by
Sangwon Lim (sangwonlim@snu.ac.kr)

Contents

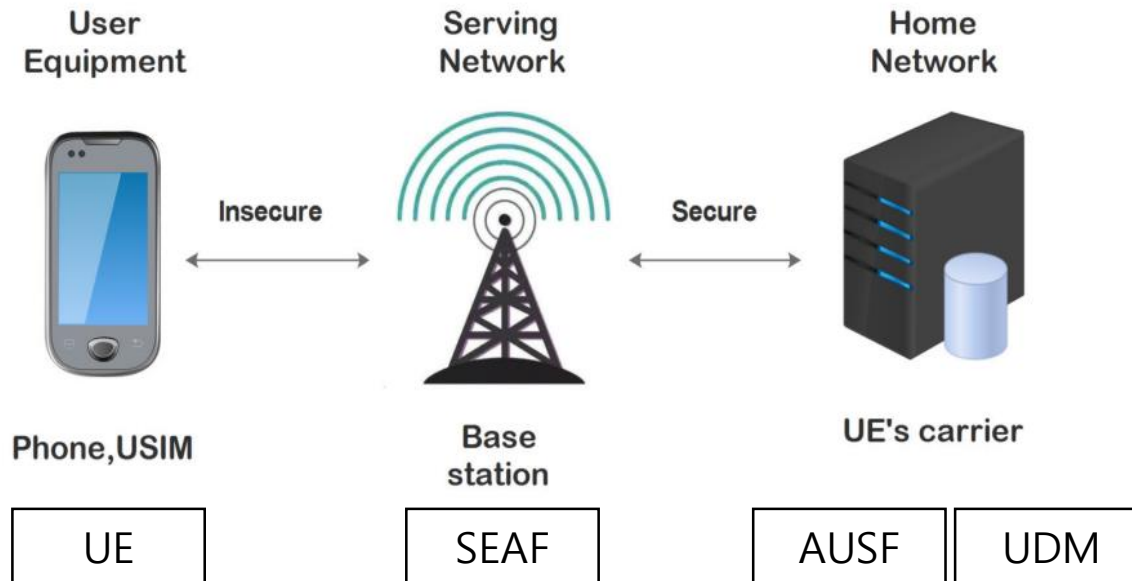
- Introduction & Background & Related works
- Pi calculus (a formal language)
- 5G EAP-TLS Protocol
- Formal model of the 5G EAP-TLS
- Verification results
- Conclusion & Critique

Introduction

- As a critical component of the security architecture of 5G network, the authentication protocol plays a role of the first safeguard in ensuring the communication security
- EAP-TLS was one of such protocols being defined in the 5G standards to provide key services in the specific IoT circumstances
- The authors present in this work a comprehensive formal analysis of the security related properties of the 5G EAP-TLS authentication protocol based on the symbolic model checking approach

Background

- The 5G network architecture



- **SEAF** (Security Anchor Function): acts as "middleman" during the authentication process between a UE and its home network
- **AUSF** (Authentication Server Function): makes the decision on UE authentication
- **UDM** (Unified data management): hosts functions such as the Authentication Credential Repository and Processing Function (**ARPF**), which selects an authentication method based on subscriber identity and configured policy and computes the **authentication data** and **Keys**

Background

- Techniques available in 5G for mutual authentication between the subscriber and the network
 - 5G-AKA: **A**uthentication and **K**ey **A**greement
 - EAP-AKA: **E**xtensible **A**uthentication **P**rotocol – **A**uthentication and **K**ey **M**anagement
 - EAP-TLS: **E**xtensible **A**uthentication **P**rotocol – **T**ransport **L**ayer **S**ecurity
- Software programs for formal verification of cryptographic protocols
 - **Scyther**: It has been used to analyse the IKEv1 and IKEv2 protocol suites
 - **Tamarin Prover**: It has been used to verify TLS1.3, and DNP3 Secure Authentication v5
 - **ProVerif**: It has been used to verify TLS 1.3, and Intel SGX

Related works

- Comparison of the formal models of 5G authentication protocols

Protocol	5G AKA				5G AKA'	5G EAP-TLS	
Article	[19]	[20]	[27]	[24]	[19]	[28]	This paper
Cryptographic primitives	Shared key cryptography	Shared key cryptography	Shared key cryptography	Shared key cryptography	Shared key cryptography	Public key cryptography	Public key cryptography
Modeling entities	UE,SEAF, AUSF	UE,SEAF, AUSF,ARPF	UE,SEAF, AUSF	UE,AUSF	UE,SEAF, AUSF	UE,SEAF, AUSF	UE,SEAF, AUSF,ARPF
Model checker being used	TAMARIN	TAMARIN	TAMARIN	-	TAMARIN	Scyther	ProVerif
Modeling language	Multiset rewriting rules	Multiset rewriting rules	Multiset rewriting rules	Bana-comon logic	Multiset rewriting rules	Role scripts	Applied pi calculus
Security Properties	Confidentiality of session key, SUPI and SQN; Authentication of each entity	Confidentiality of session key, SUPI and SQN; Authentication of each entity	Confidentiality of SQN	Unlinkability between UE and AUSF	Confidentiality of session key, SUPI and SQN; Authentication of each entity	Confidentiality of session key and SUPI; Authentication of each entity	Confidentiality of session key and SUPI; Authentication of each entity and session key
Threat model	Dolev-Yao model and compromised components	Dolev-Yao model and compromised components	Dolev-Yao model	Customize model	Dolev-Yao model and compromised components	Dolev-Yao model	Dolev-Yao model

- SUPI: Subscription Permanent Identifier; SQN: Sequence number
- Dolev-Yao model: considers only adversaries that can compose and replay messages, and decipher them with known keys

Pi calculus

- Pi calculus is a formal language for security protocol modeling and popularized by the ProVerif model checker

- Syntax

Terms

$M, N ::=$	terms
a, b, c, k, m, n, s	names
x, y, z	variables
(M_1, \dots, M_k)	tuple
$h(M_1, \dots, M_k)$	constructor/destructor
$M = N$	term equality
$M <> N$	term inequality
$M \&\& M$	conjunction
$M M$	disjunction
$\text{not}(M)$	negation

Pattern matching

$T ::=$	patterns
$x : t$	typed variable
x	variable without explicit type
(T_1, \dots, T_n)	tuple
$= M$	equality test

Process

$P, Q, R ::=$	processes
0	null process
$P Q$	parallel composition
$!P$	replication
$\text{new } n : t; P$	name restriction
$\text{in}(M, x : t); P$	message input
$\text{out}(M, N); P$	message output
$\text{if } M \text{ then } P \text{ else } Q$	conditional
$\text{let } x = M \text{ in } P \text{ else } Q$	term evaluation
$R(M_1, \dots, M_n)$	macro usage

Pi calculus

- Examples

Symmetric enc/dec

type key.

```
fun senc(bitstring, key): bitstring.  
  reduc forall m: bitstring, k: key; sdec(senc(m, k), k) = m.
```

Asymmetric enc/dec

type skey.

type pkey.

```
fun pk(skey): pkey.  
fun aenc(bitstring, pkey): bitstring.  
  reduc forall m: bitstring, sk: skey; adec(aenc(m, pk(sk)), sk) = m.
```

Digital signature/check sign

type sskey.

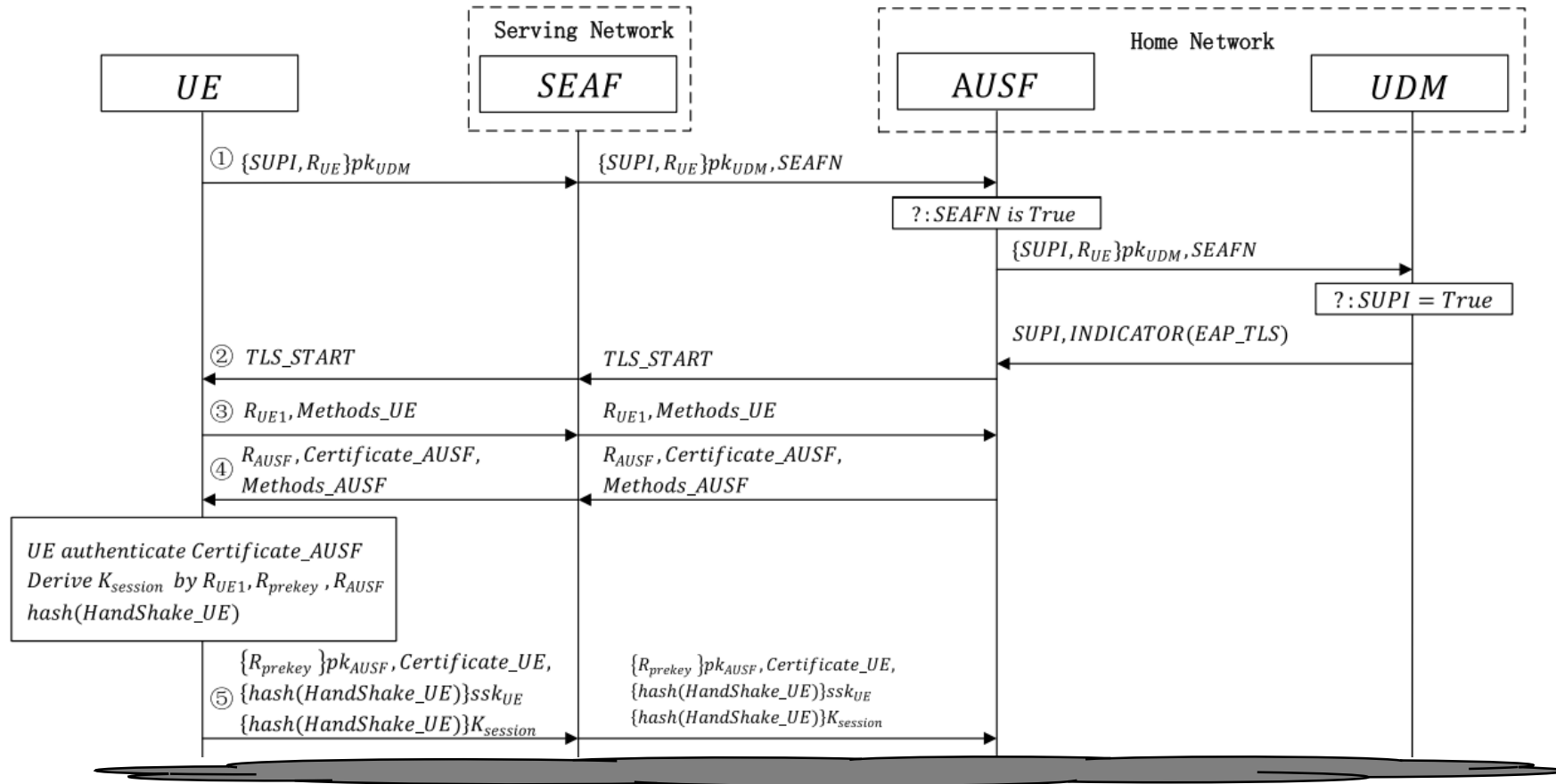
type spkey.

```
fun spk(sskey): spkey.  
fun sign(bitstring, sskey): bitstring.  
  reduc forall m: bitstring, k: sskey;  
    checksign(sign(m, k), spk(k)) = m.
```

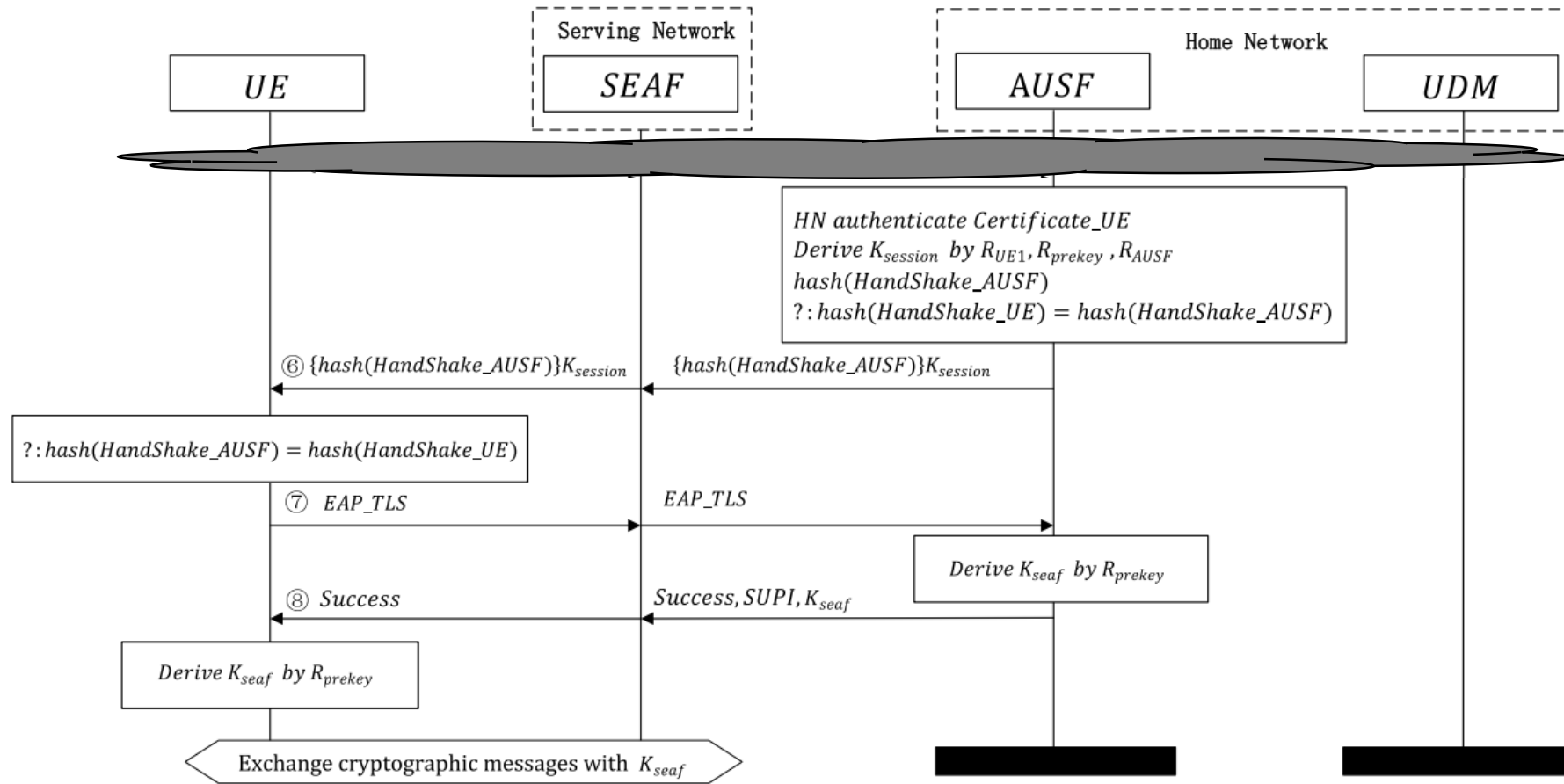
Hash / key Derivation

```
fun h(bitstring, bitstring, bitstring): bitstring.  
fun b2k(bitstring): key.
```


5G EAP-TLS Protocol¹⁾

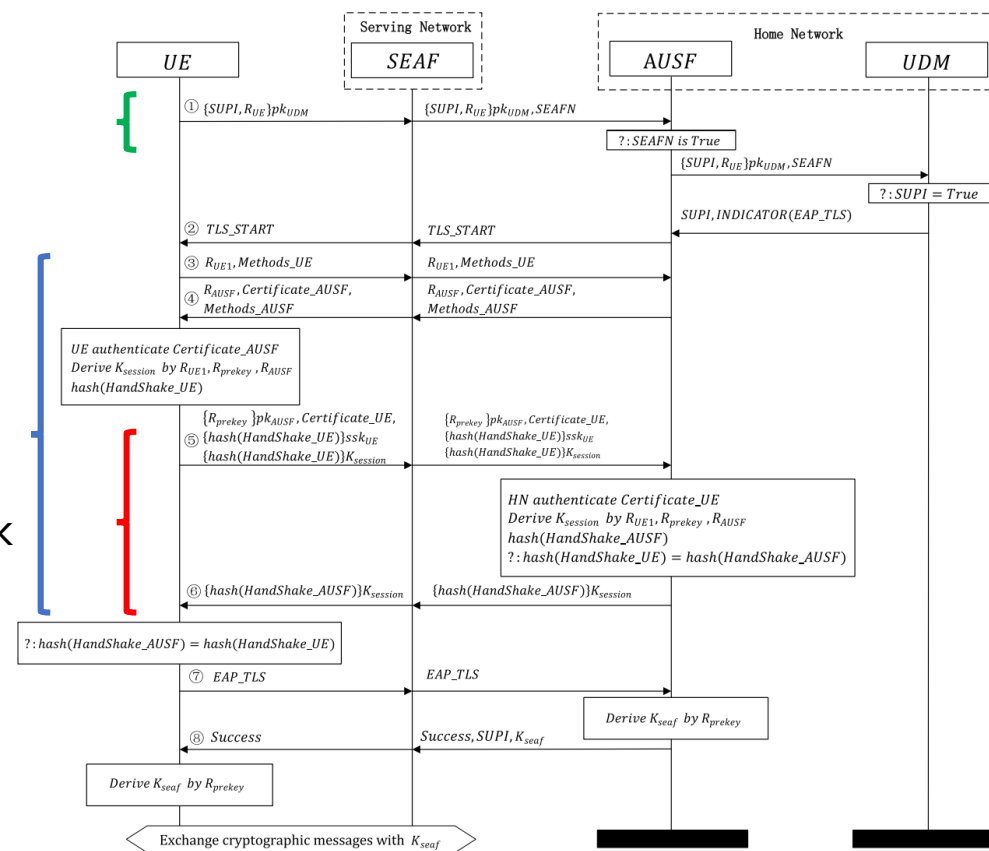


5G EAP-TLS Protocol



5G EAP-TLS Protocol

- Required security properties¹⁾
 - User identity confidentiality
 - User identity confidentiality ← ①
- Authentication and Authorization
 - Subscription authentication ← ⑤,⑥
 - Serving network authentication ← ③,④,⑤,⑥
 - UE authorization
 - Serving network authorization by the home network
 - Access network authorization
- Confidentiality
 - Cipher key agreement ← ③,④,⑤,⑥
 - Confidentiality of user data ← ③,④,⑤,⑥



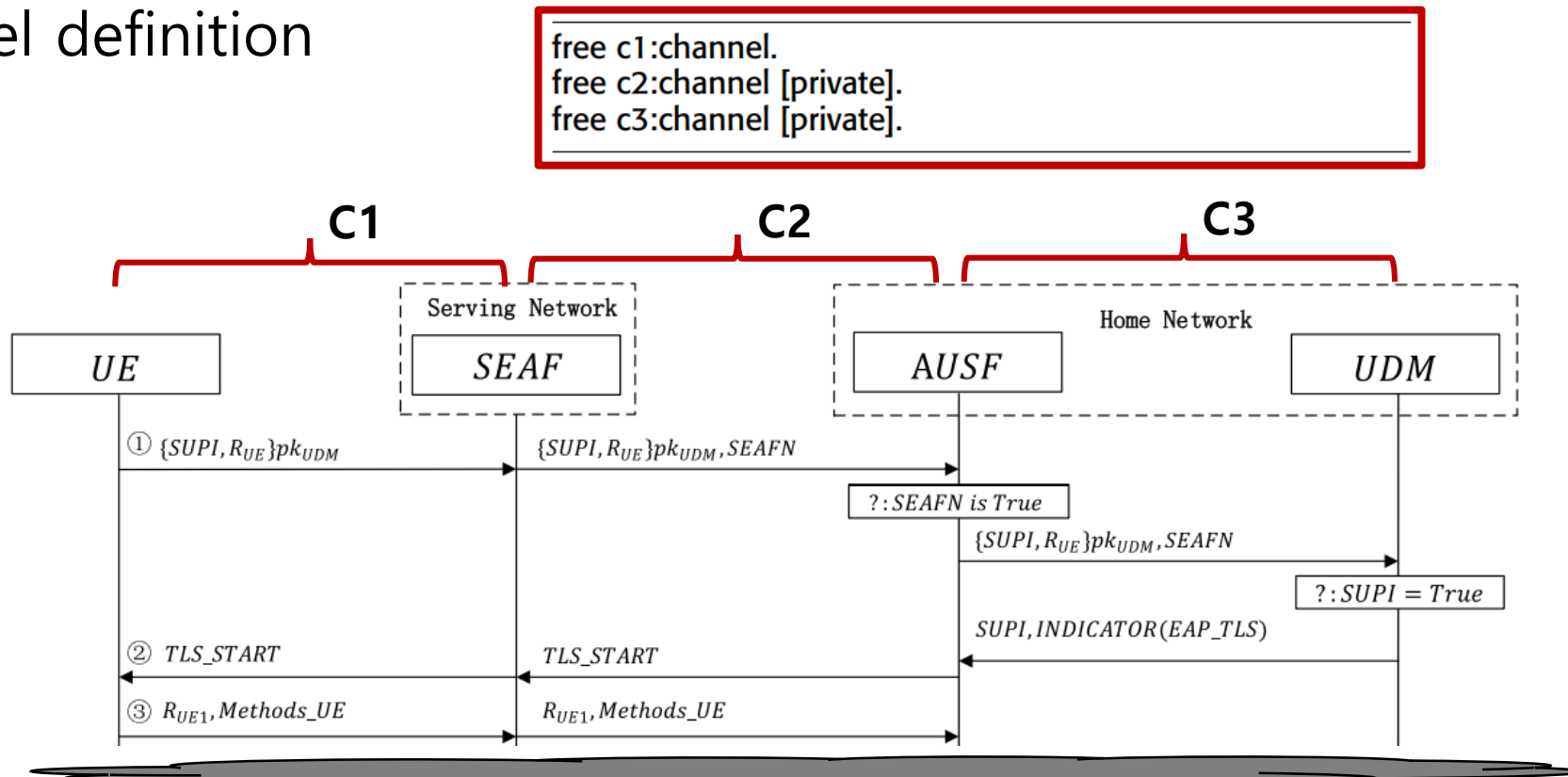
< 5G EAP-TLS Protocol >

Formal model of the 5G EAP-TLS

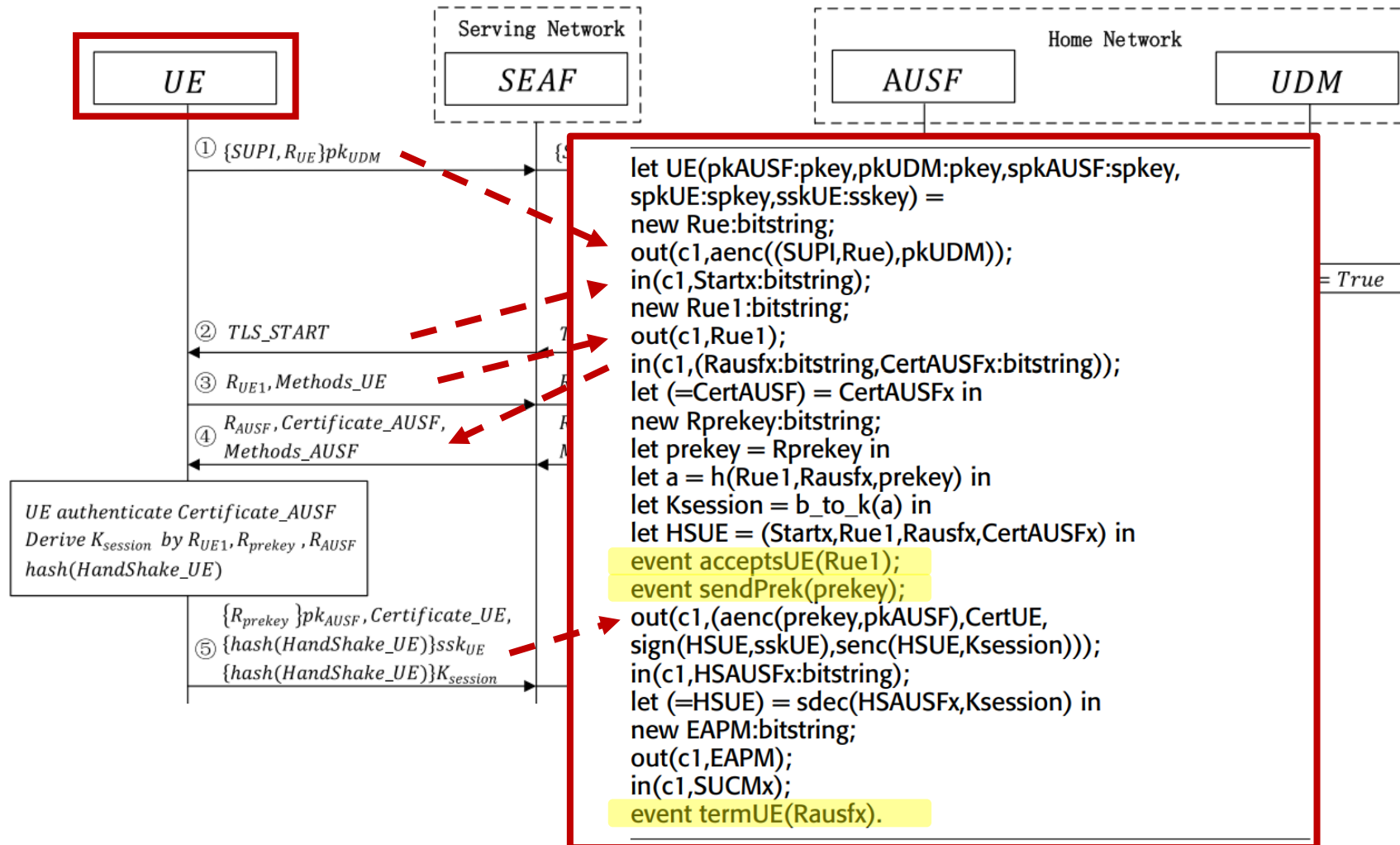
- Formal language
 - Pi calculus
- Threat model
 - Dolev-Yao
 - Attacker has full control over the network
 - Perfect cryptography assumption
- Model checker
 - ProVerif

Formal model of the 5G EAP-TLS

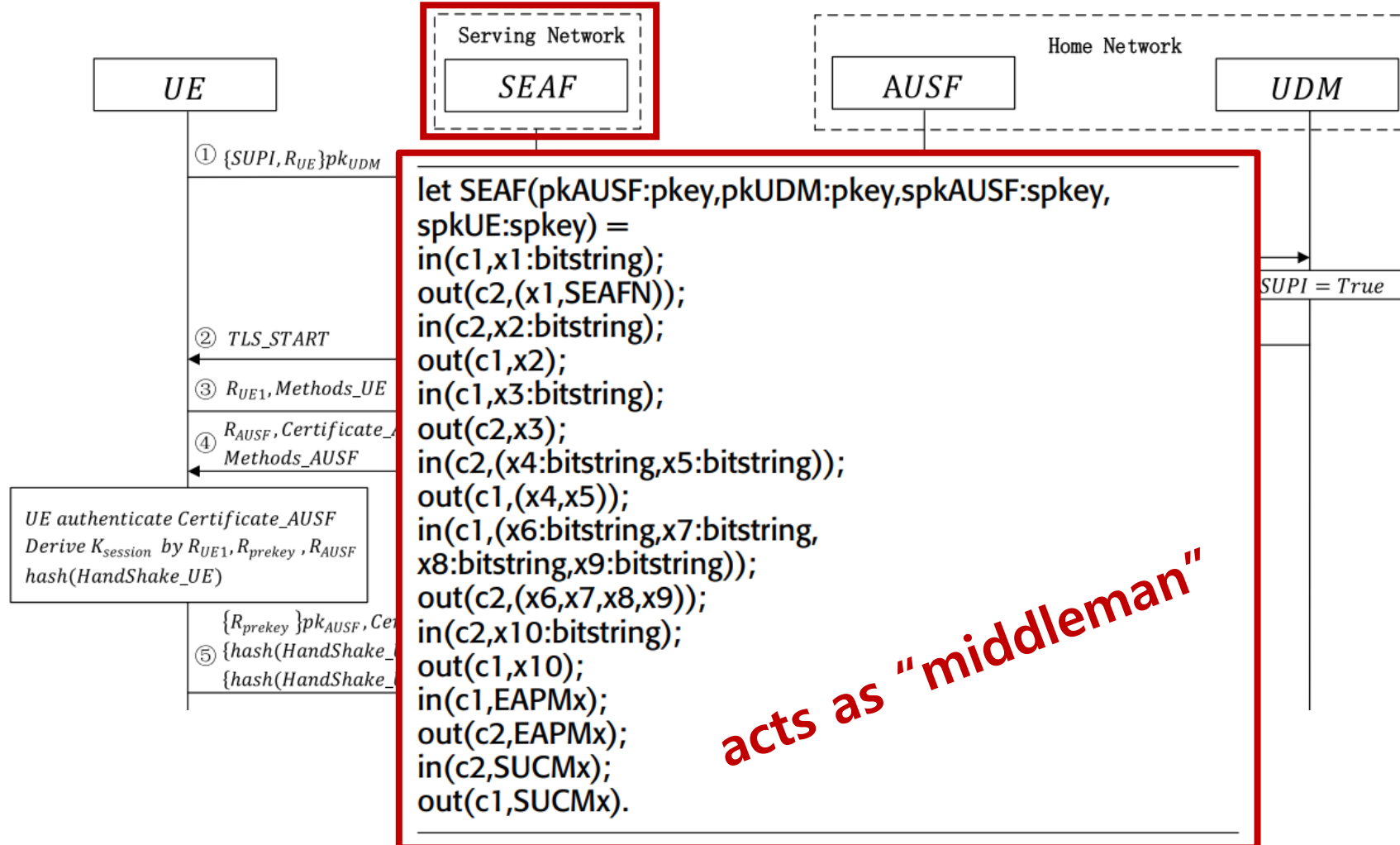
- Channel definition



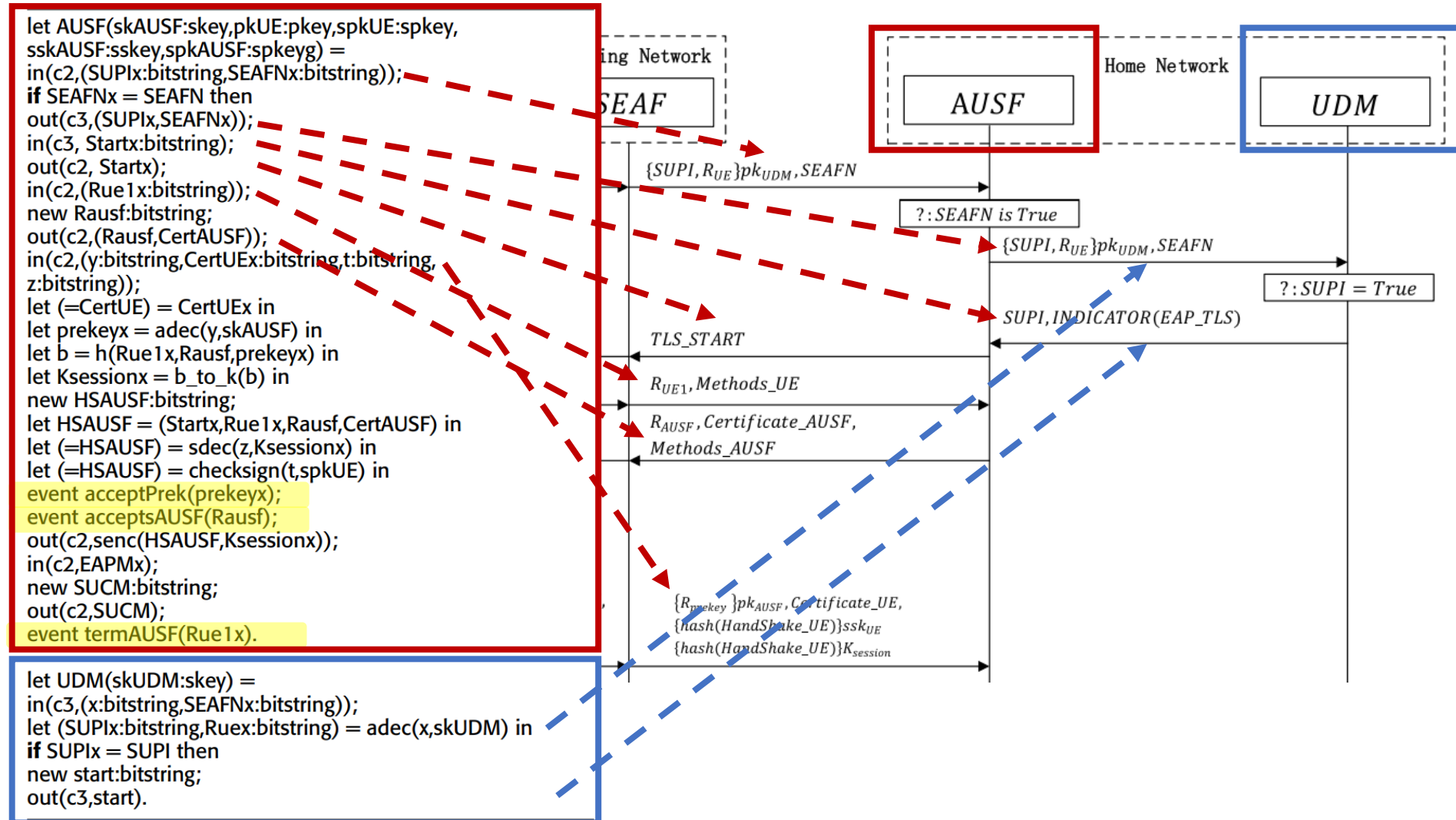
Formal model of the 5G EAP-TLS



Formal model of the 5G EAP-TLS

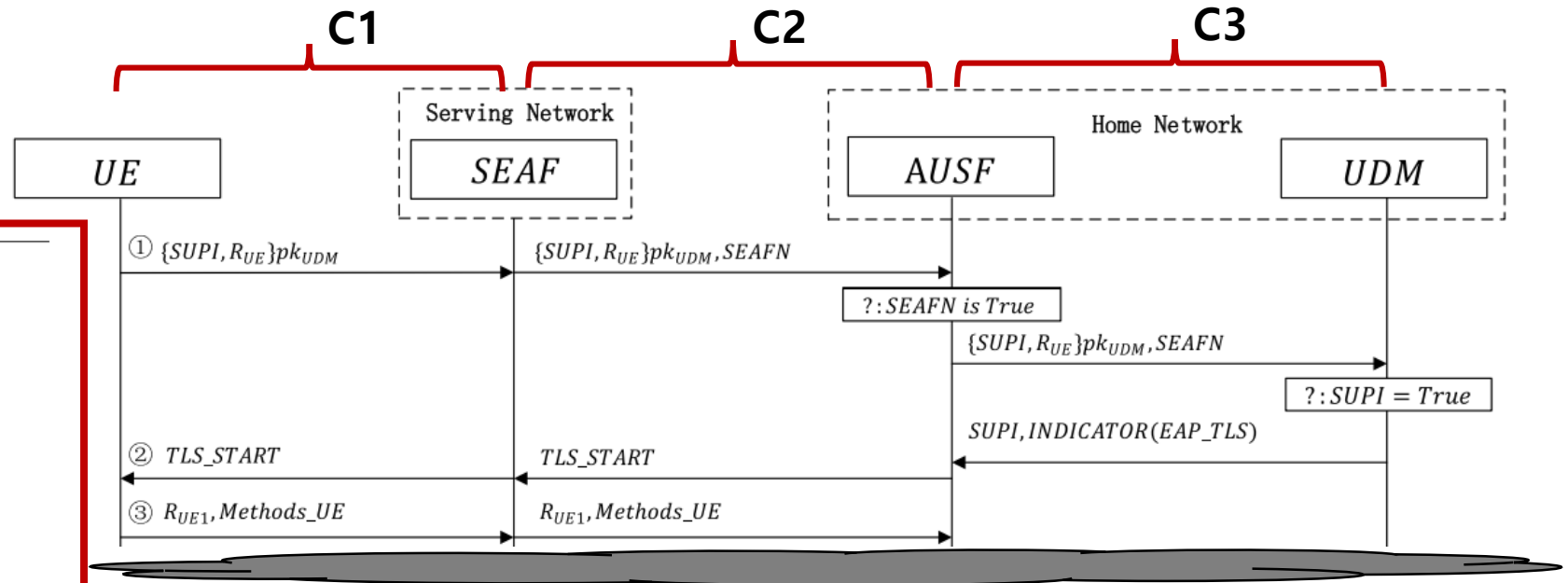


Formal model of the 5G EAP-TLS



Formal model of the 5G EAP-TLS

- Protocol process



```

process
new skUE:skey;
new skAUSF:skey;
new skUDM:skey;
new sskAUSF:sskey;
new sskUE:sskey;
new SEAFN:bitstring;
let pkUE = pk(skUE) in out(c1,pkUE);
out(c2,pkUE);out(c3,pkUE);
let pkAUSF = pk(skAUSF) in out(c1,pkAUSF);
out(c2,pkAUSF);out(c3,pkAUSF);
let pkUDM = pk(skUDM) in out(c1,pkUDM);
out(c2,pkUDM);out(c3,pkUDM);
let spkAUSF = spk(sskAUSF) in out(c1,spkAUSF);
out(c2,spkAUSF);out(c3,spkAUSF);
let spkUE = spk(sskUE) in out(c1,spkUE);
out(c2,spkUE);out(c3,spkUE);
out(c1,SEAFN);out(c2,SEAFN);out(c3,SEAFN);
( (!UE(pkAUSF,pkUDM,spkAUSF,spkUE,sskUE)) |
  (!SEAF(pkAUSF,pkUDM,spkAUSF,spkUE)) |
  (!AUSF(skAUSF,pkUE,spkUE,sskAUSF,spkAUSF))
  | (!UDM(skUDM)))
    
```

Formal model of the 5G EAP-TLS

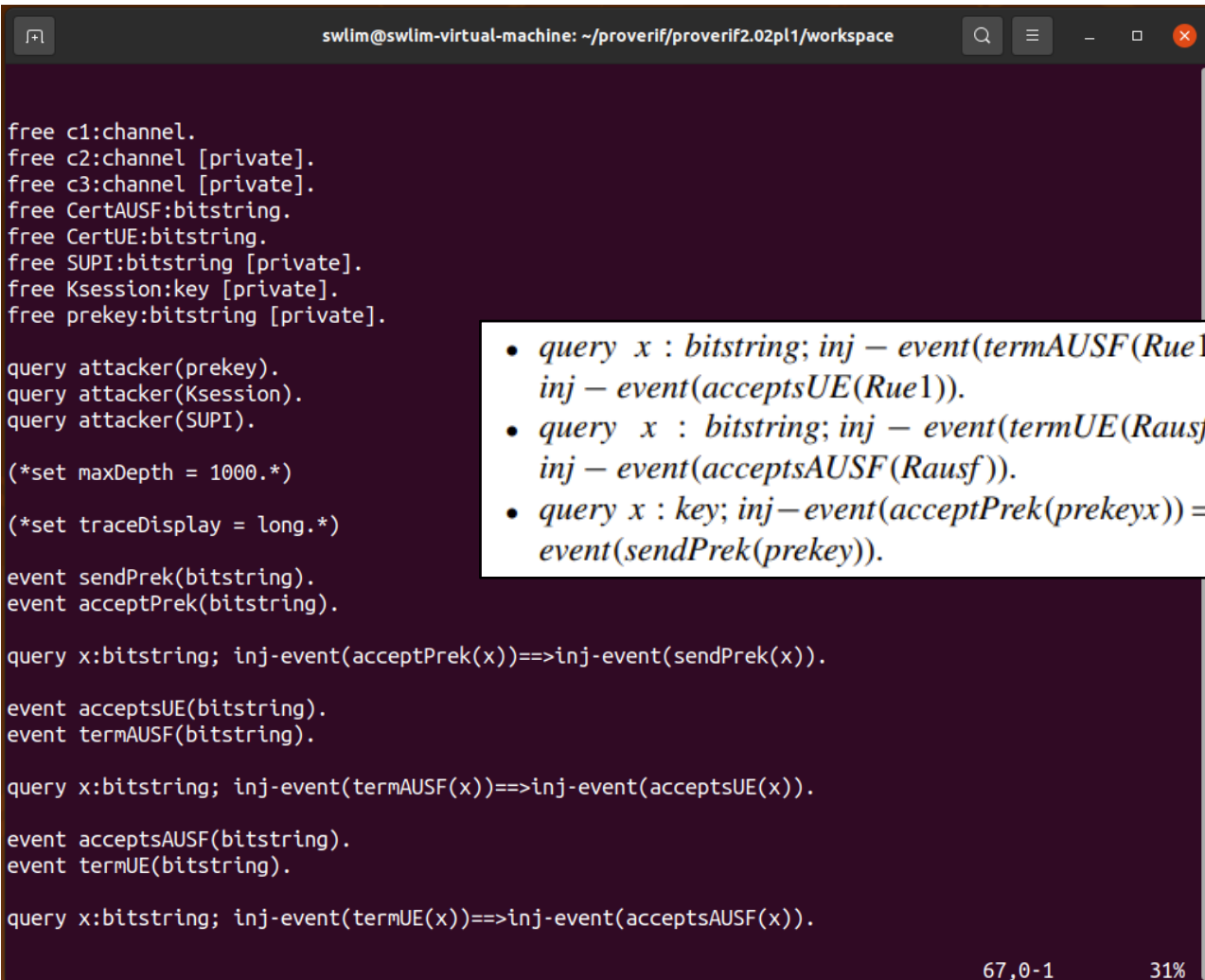
- Security property

*whether the attacker can reach
a state where the term M is available*

*whenever the network terminates a
protocol run, there exists a user who
has accepted to run with the network*

...

...



```
swlim@swlim-virtual-machine: ~/proverif/proverif2.02pl1/workspace

free c1:channel.
free c2:channel [private].
free c3:channel [private].
free CertAUSF:bitstring.
free CertUE:bitstring.
free SUPI:bitstring [private].
free Ksession:key [private].
free prekey:bitstring [private].

query attacker(prekey).
query attacker(Ksession).
query attacker(SUPI).

(*set maxDepth = 1000.*)

(*set traceDisplay = long.*)

event sendPrek(bitstring).
event acceptPrek(bitstring).

query x:bitstring; inj-event(acceptPrek(x))=>inj-event(sendPrek(x)).

event acceptsUE(bitstring).
event termAUSF(bitstring).

query x:bitstring; inj-event(termAUSF(x))=>inj-event(acceptsUE(x)).

event acceptsAUSF(bitstring).
event termUE(bitstring).

query x:bitstring; inj-event(termUE(x))=>inj-event(acceptsAUSF(x)).
```

- $query\ x : bitstring; inj - event(termAUSF(Rue1x)) \Rightarrow inj - event(acceptsUE(Rue1)).$
- $query\ x : bitstring; inj - event(termUE(Rausfx)) \Rightarrow inj - event(acceptsAUSF(Rausf)).$
- $query\ x : key; inj - event(acceptPrek(prekeyx)) \Rightarrow inj - event(sendPrek(prekey)).$

67,0-1

31%

18 / 23

Verification results

- The agreement properties (i.e. A1 and A2) are violated

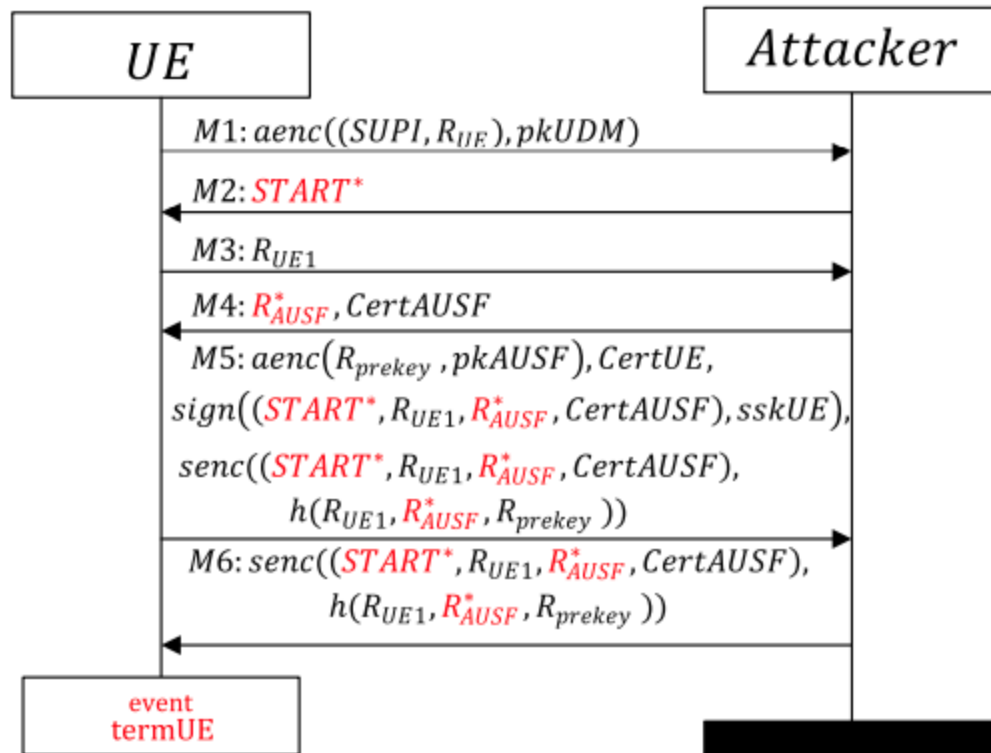
Security property	Result
A1. Both the home network ($AUSF$) and the subscriber (UE) should agree on the identity of each other after successful termination.	False
A2. Both the home network ($AUSF$) and the subscriber (UE) should agree on the pre-master key (R_{prekey}) after successful termination.	False
S1. The adversary must not be able to obtain the $SUPI$ of an honest subscriber.	True
S2. The adversary must not be able to obtain the pre-master key (R_{prekey}) of an honest subscriber.	True
S3. The adversary must not be able to obtain the session key ($K_{session}$) of an honest subscriber.	True

```
.....
Verification summary:
Query not attacker(prekey[]) is true.
Query not attacker(Ksession[]) is true.
Query not attacker(SUPI[]) is true.
Query inj-event(acceptPrek(x_1)) ==> inj-event(sendPrek(x_1)) cannot be proved.
Query inj-event(termAUSF(x_1)) ==> inj-event(acceptsUE(x_1)) is true.
Query inj-event(termUE(x_1)) ==> inj-event(acceptsAUSF(x_1)) cannot be proved.
.....
swlim@swlim-virtual-machine:~/proverif/proverif2.02pl1/workspace$
```

< Results reproduced on my computer >

Verification results

- The counterexample for property A1

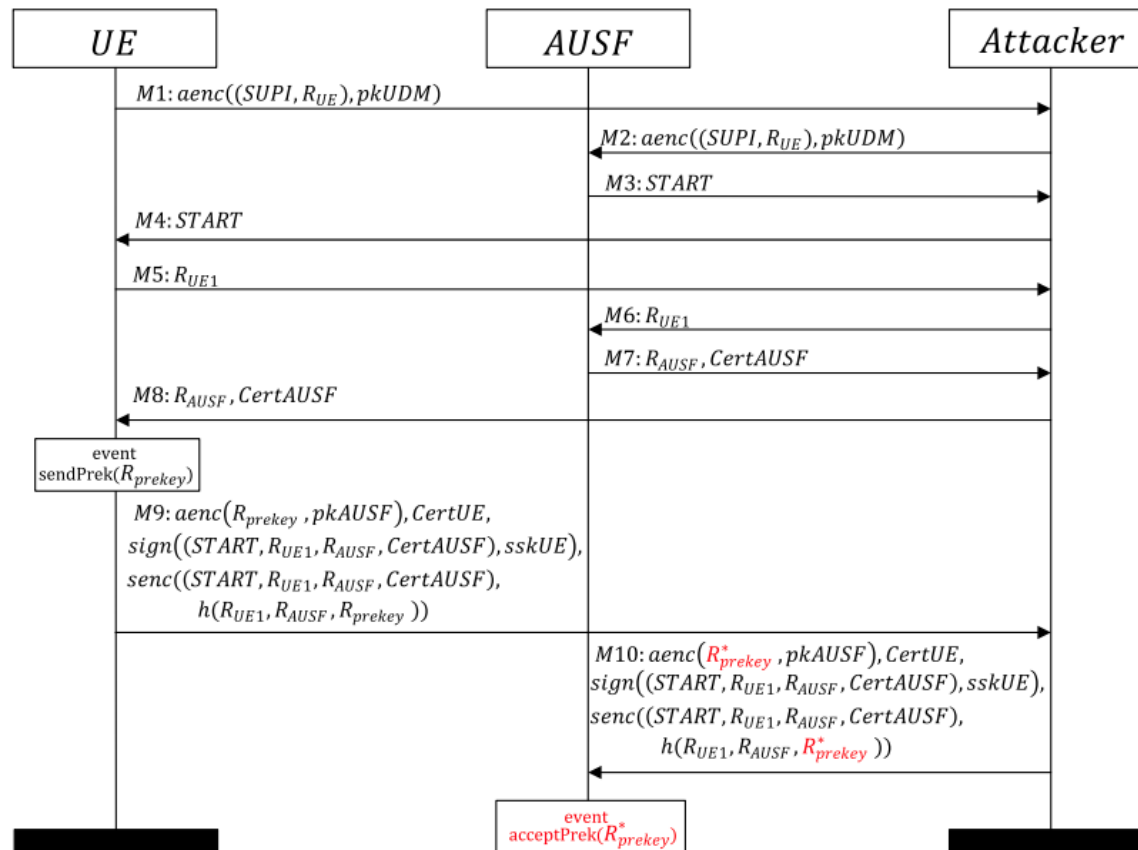


A1. Both the home network (*AUSF*) and the subscriber (*UE*) should agree on the identity of each other after successful termination.

- query* $x : \text{bitstring}; \text{inj} - \text{event}(\text{termAUSF}(\text{Rue1}x)) \Rightarrow \text{inj} - \text{event}(\text{acceptsUE}(\text{Rue1})).$
- query* $x : \text{bitstring}; \text{inj} - \text{event}(\text{termUE}(\text{Rausf}x)) \Rightarrow \text{inj} - \text{event}(\text{acceptsAUSF}(\text{Rausf})).$ **false**

Verification results

- The counterexample for property A2



A2. Both the home network ($AUSF$) and the subscriber (UE) should agree on the pre-master key (R_{prekey}) after successful termination.

- $query\ x : key; inj - event(acceptPrek(prekeyx)) \Rightarrow inj - event(sendPrek(prekey)).$ **false**

Conclusion

- The authors investigate the security properties of 5G EAP-TLS authentication protocol that is being standardized by 3GPP
- They model the protocol and its security properties in the applied pi-calculus and carry out the analysis using model checker ProVerif
- Their analysis reveals several design flaws and counterexamples are reported to show the possibilities of these flaws

Critique

- Through formal verification, flaws in many popular protocols were discovered
- Recently, formal verification is performed together when designing a protocol (e.g., TLS 1.3)
- If we learn how to formally verify, it will be useful when we design a new protocol or improve an existing protocol