Security Protocols
Model Checking Standards

David Basin
ETH Zurich

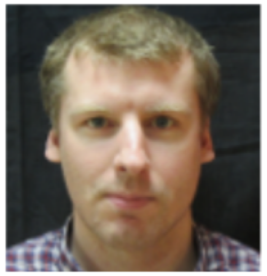
CASTOR Software Days
October 2019

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Thanks

Tamarin Team


Simon Meier


Benedikt Schmidt


Cas Cremers


Ralf Sasse


Jannik Dreier

ISO/9798 (verified using precursor tools)


Simon Meier


Cas Cremers

5G (verified using Tamarin)


Lucca Hirschi


Ralf Sasse


Jannik Dreier


Sasa Radomirovic


Vincent Stettler

A Typical Protocol

IKE, Phase 1, Main Mode, Digital Signatures, Simplified

$$
\begin{array}{lll}
(1) & \mathrm{I} \to \mathrm{R}: & \mathrm{C_I},\ \mathrm{ISA_I} \\
(2) & \mathrm{R} \to \mathrm{I}: & \mathrm{C_I},\ \mathrm{C_R},\ \mathrm{ISA_R} \\
(3) & \mathrm{I} \to \mathrm{R}: & \mathrm{C_I},\ \mathrm{C_R},\ \mathrm{g^x},\ \mathrm{N_I} \\
(4) & \mathrm{R} \to \mathrm{I}: & \mathrm{C_I},\ \mathrm{C_R},\ \mathrm{g^y},\ \mathrm{N_R} \\
(5) & \mathrm{I} \to \mathrm{R}: & \mathrm{C_I},\ \mathrm{C_R},\ \{\mathrm{ID_I},\ \mathrm{SIG_I}\}_{\mathrm{SKEYID_e}} \\
(6) & \mathrm{R} \to \mathrm{I}: & \mathrm{C_I},\ \mathrm{C_R},\ \{\mathrm{ID_R},\ \mathrm{SIG_R}\}_{\mathrm{SKEYID_e}}
\end{array}
$$

Properties?

$$
\begin{array}{lcl}
\mathrm{SKEYID} & = & \mathrm{h(\{N_I, N_R\}, g^{xy})} \\
\mathrm{SKEYID_d} & = & \mathrm{h(SKEYID, \{g^{xy}, C_I, C_R, 0\})} \\
\mathrm{SKEYID_a} & = & \mathrm{h(SKEYID, \{SKEYID_d, g^{xy}, C_I, C_R, 1\})} \\
\mathrm{SKEYID_e} & = & \mathrm{h(SKEYID, \{SKEYID_a, g^{xy}, C_I, C_R, 2\})} \\
\mathrm{HASH_I} & = & \mathrm{h(SKEYID_a, \{g^x, g^y, C_I, C_R, ISA_I, ID_I\})} \\
\mathrm{HASH_R} & = & \mathrm{h(SKEYID_a, \{g^y, g^x, C_R, C_I, ISA_R, ID_R\})} \\
\mathrm{SIG_I} & = & \mathrm{\{HASH_I\}_{K_I^{-1}}} \\
\mathrm{SIG_R} & = & \mathrm{\{HASH_R\}_{K_R^{-1}}}
\end{array}
$$

Does argument order matter?

Why all the nested keyed hashes?

Protocol Design as an Art

Best practices, design by committee, reuse of previous protocols, ...

Whenever I made a roast, I always started off by cutting off the ends, just like my grandmother did. Someone once asked me why I did it, and I realized I had no idea. It had never occurred to me to wonder. It was just the way it was done. Eventually I asked my grandmother. "Why do you always cut off the ends of a roast?" She answered "Because my pan is small and otherwise the roasts would not fit."

— *Anonymous*

Protocol Design as a Science

Science in the root sense

The discovery and knowledge of something that can be demonstrated and verified within a community

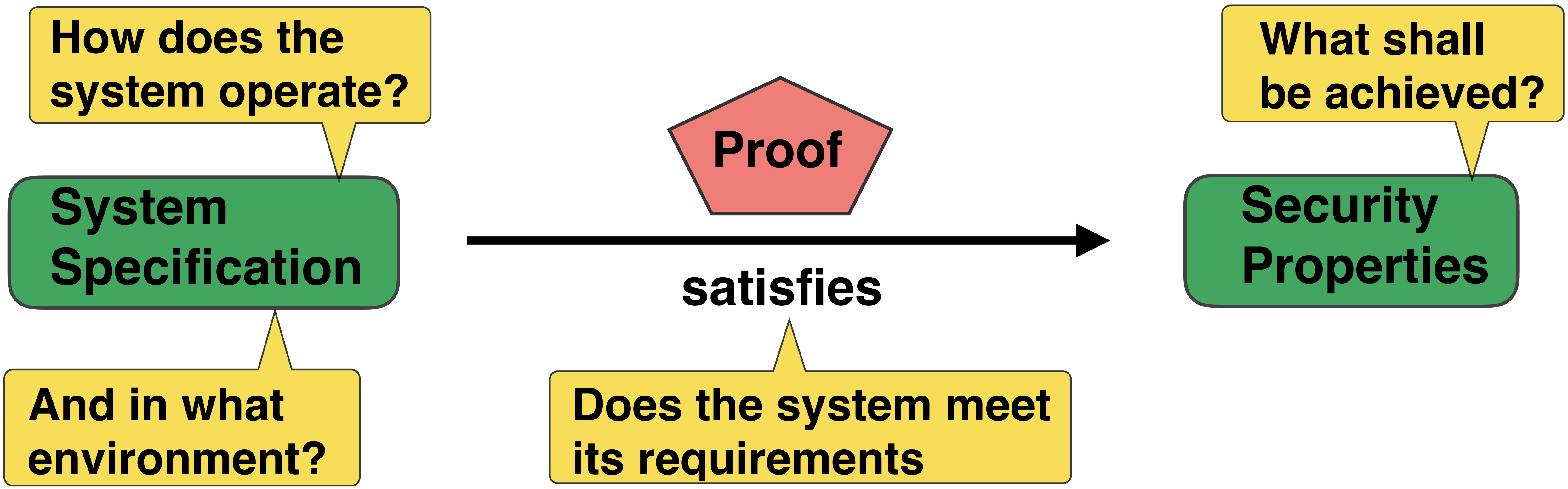
Formal methods as a way to better protocols

- Precise specification of system, environment, properties
- Tool support to debug, verify, and explore alternatives

Progress is being made applying tools to protocols that matter

- **ISO/IEC 9798**, **5G**, TLS 1.3, …
- Companies are (slowly) becoming tool users

Where is the Difficulty?



- Design documents are incomplete and imprecise
- Unclear adversary model

- Undecidability
- Even restricted cases intractable

- Properties implicit or imprecise. E.g. “authenticate”

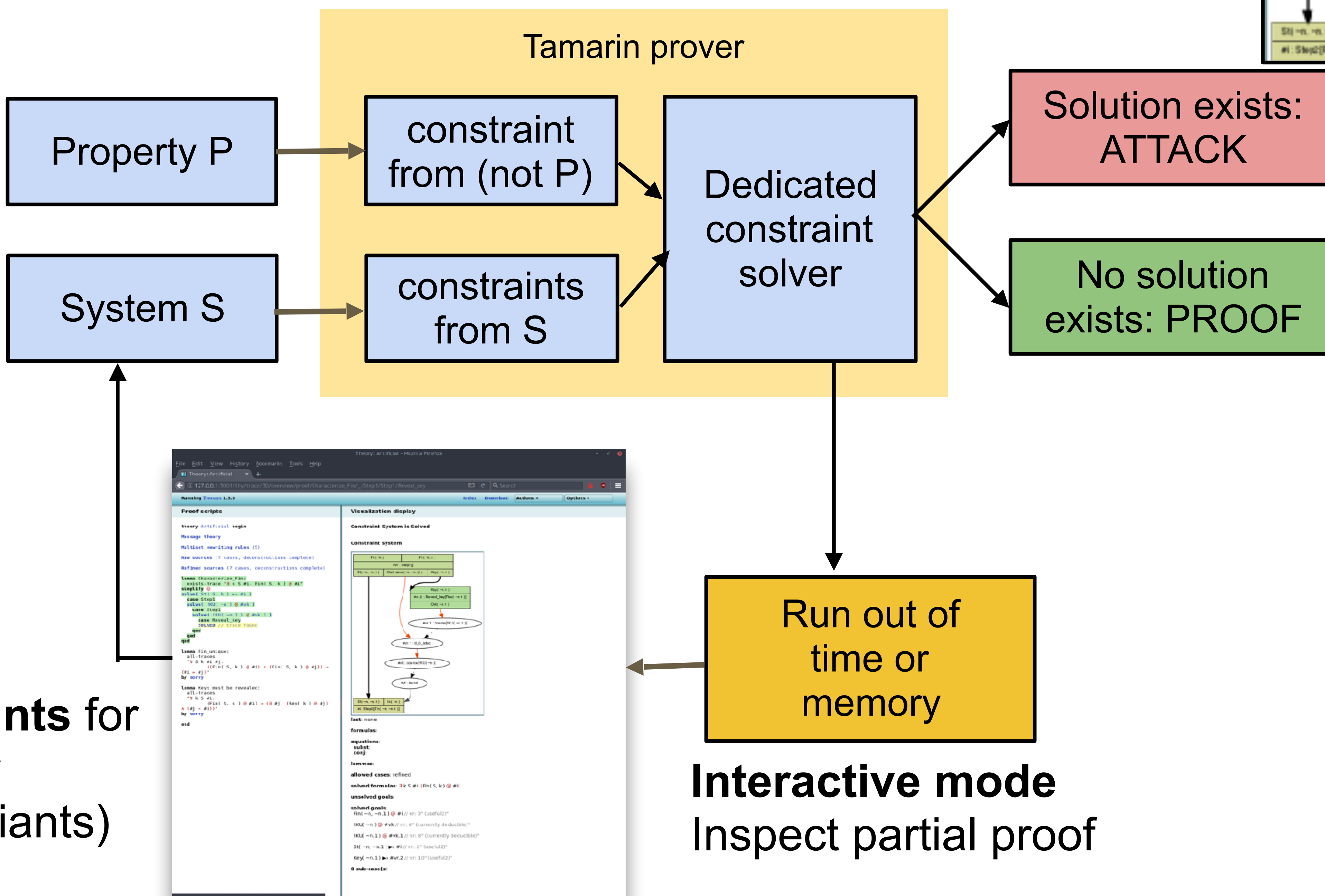
Weapon of Choice

Theorem Prover

Constraint solver

Tamarin prover

Tamarin Prover



Provide **hints** for the prover (e.g. invariants)

Interactive mode
Inspect partial proof

Specifying Protocols with Multiset Rewrite Rules

LHS --[actions]-> RHS

```
[ In( K ),
  State( ThreadID, `step1’ ) ]
```

premises (LHS)

```
   --[ Accepted( ThreadID, K) ]->
```

actions

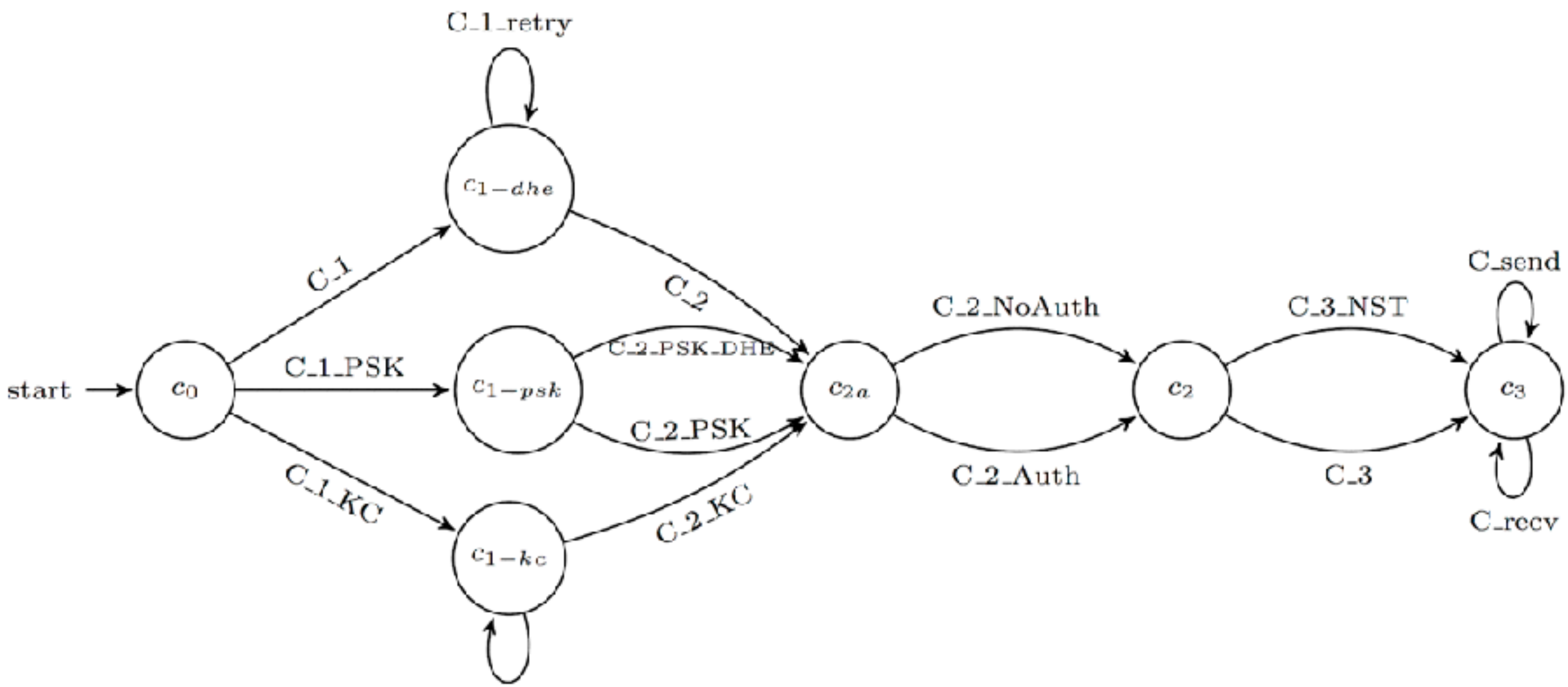
```
[ Out( `ack` ),
  State( ThreadID, `step2’, K ) ]
```

conclusions (RHS)

Gives rise to a transition system with a trace semantics

{In(key), State(tid3,`step1’), …}** $\xrightarrow{\text{Accepted(tid3,key)}}$ **{Out(`ack’), State(tid3,`step2’,key), …} $\xrightarrow{\dots}$ **…**

Specifying Protocols



Example: client state machine

Rules correspond to edges

Specifying Adversary Capabilities

Example of "Session Reveal"

```
[ State( ThreadID, … , Key ) ]
    --[ SessionKeyReveal( ThreadID, Key ) ]->
 [ Out( Key ) ]
```

Similar to oracles in computational model

Specifying Properties

Guarded fragment of first order logic with timepoints

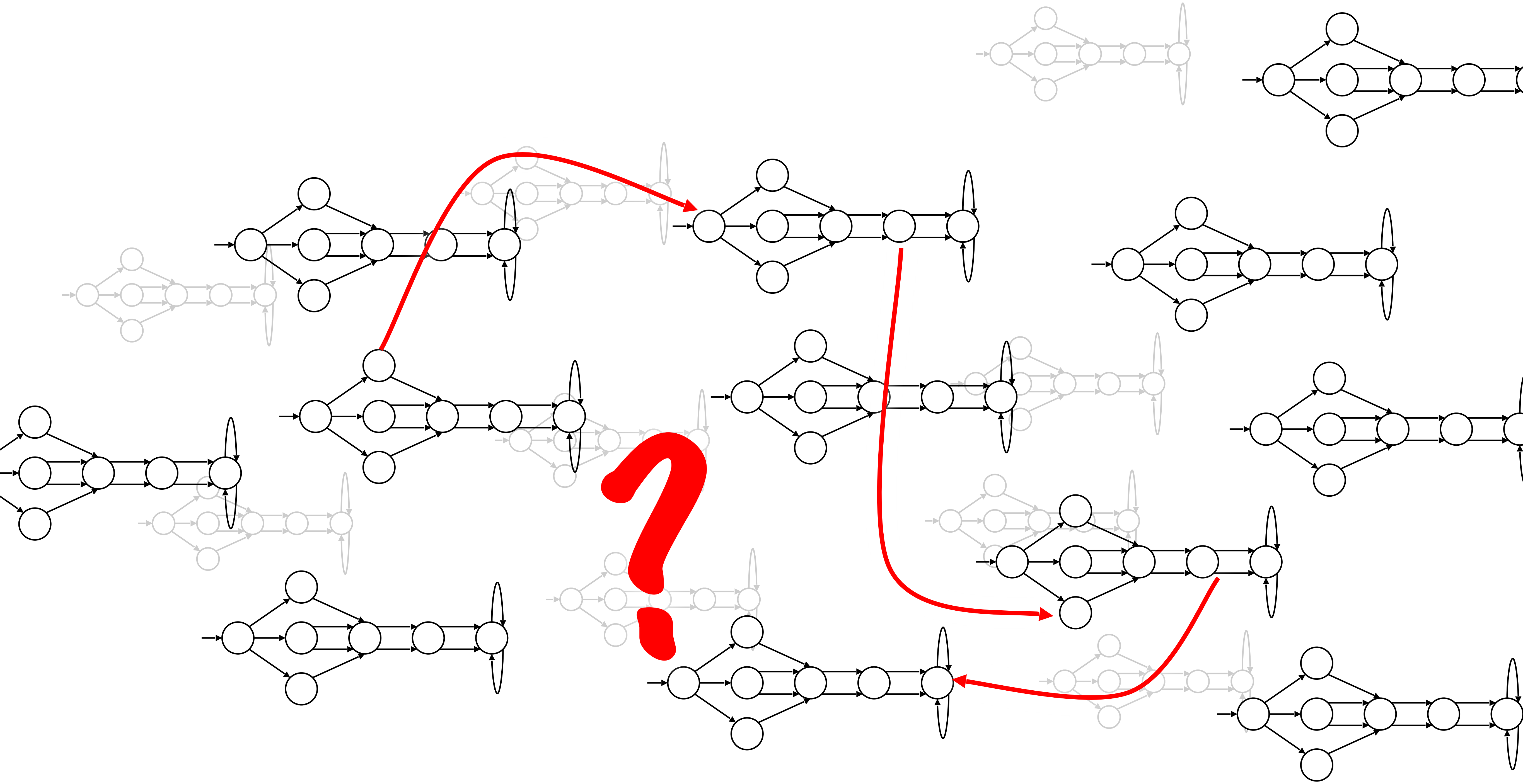
```
lemma my_secret_key:
   "Forall tid key #i.
      Accepted( tid, key )@i => ( not Ex #j. K(key)@j ) "
```

Interpreted over traces

{In(key), State(tid3,`step1'), ...} —Accepted(tid3,key)→ {Out(`ack'), State(tid3,`step2',key), ...} —...→

Does Protocol Satisfy Property?

Or can the adversary attack it?



Example #1: ISO/IEC Standard 9798

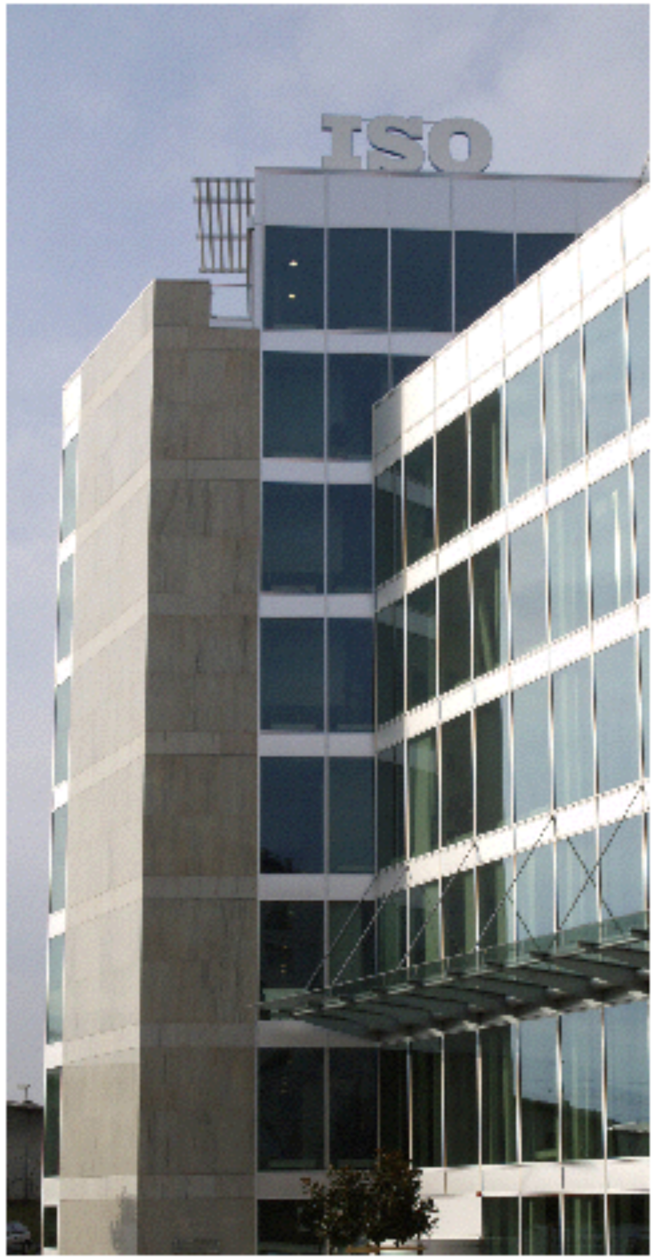
Standard for Entity Authentication Mechanisms

18 base protocols

- Symmetric-key encryption, digital signatures, cryptographic check function
- Unilateral or mutual authentication
- Additional protocols with Trusted Third Party

Many variants from optional fields

D.B., Cremers, Meier, *Provably Repairing the ISO/IEC 9798 Standard for Entity Authentication*, Journal of Computer Security, 2013.



The ISO/IEC 9798 Standard

History

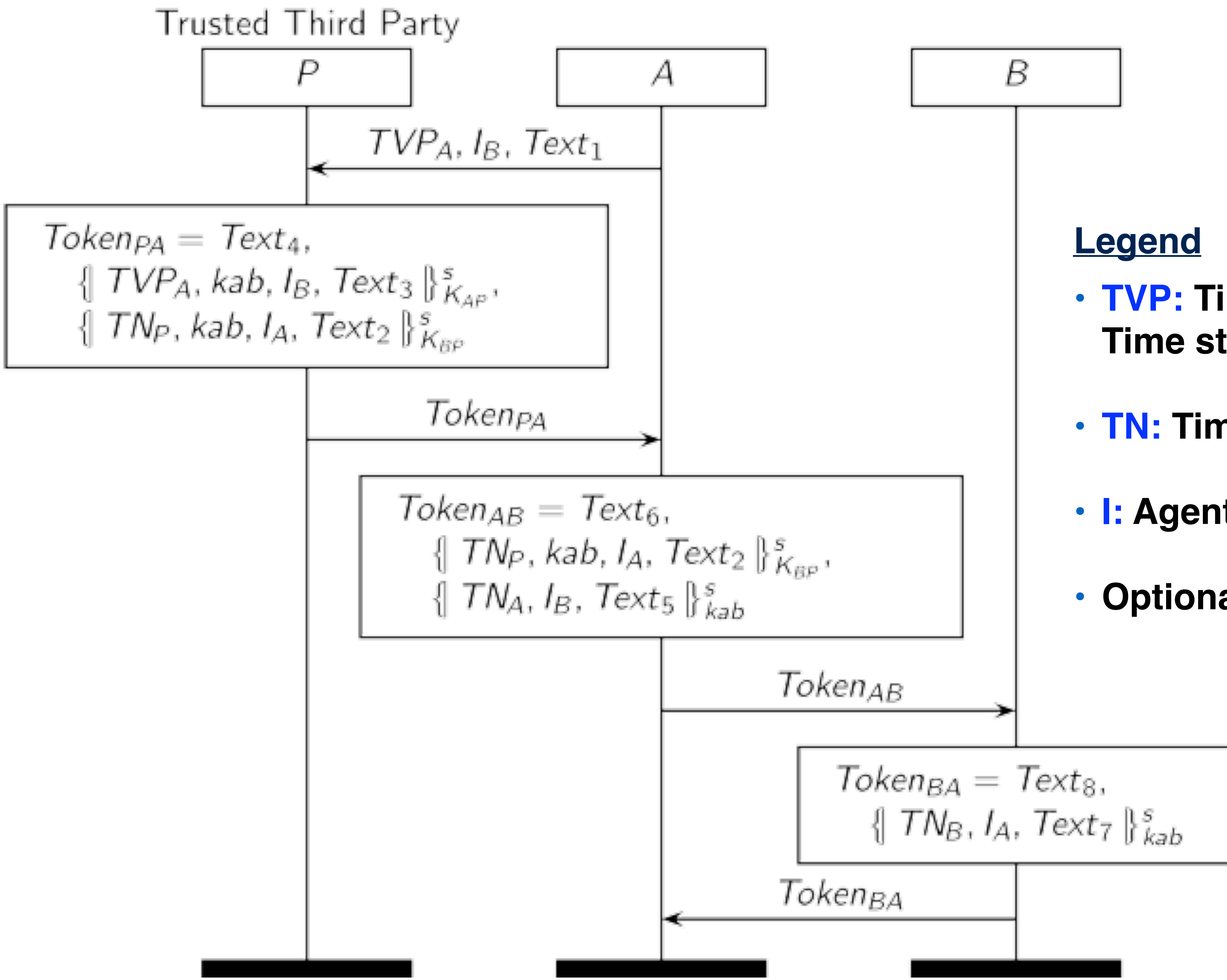
- Active development and updates since 1991
- Basis for ISO 11770 (Key Exchange) and NIST FIPS 196
- Mandated by other standards
  - e.g. European Banking Commission's smart card standards

International Organization for Standardization

Intended properties

- Entity authentication?
- Encrypted/signed payloads?
- Standard makes limited statements:
  “resistance to reflection attacks”

ISO 9798-2-5



Legend

- **TVP: Time Valued Parameter Time stamp, counter, or nonce**
- **TN: Time stamp or counter**
- **I: Agent identifier**
- **Optional Text fields**

Analysis

Request by CryptRec to evaluate standard

CRYPTREC
Cryptography Research and Evaluation Committees

- Cryptography Research and Evaluation Committees
- Funded by the Japanese's government
- Long-running program to evaluate cryptographic mechanisms

Confirmation expected

- Standard under improvement since 1994
- Substantial previous analysis

Tools used (Tamarin Precursors)

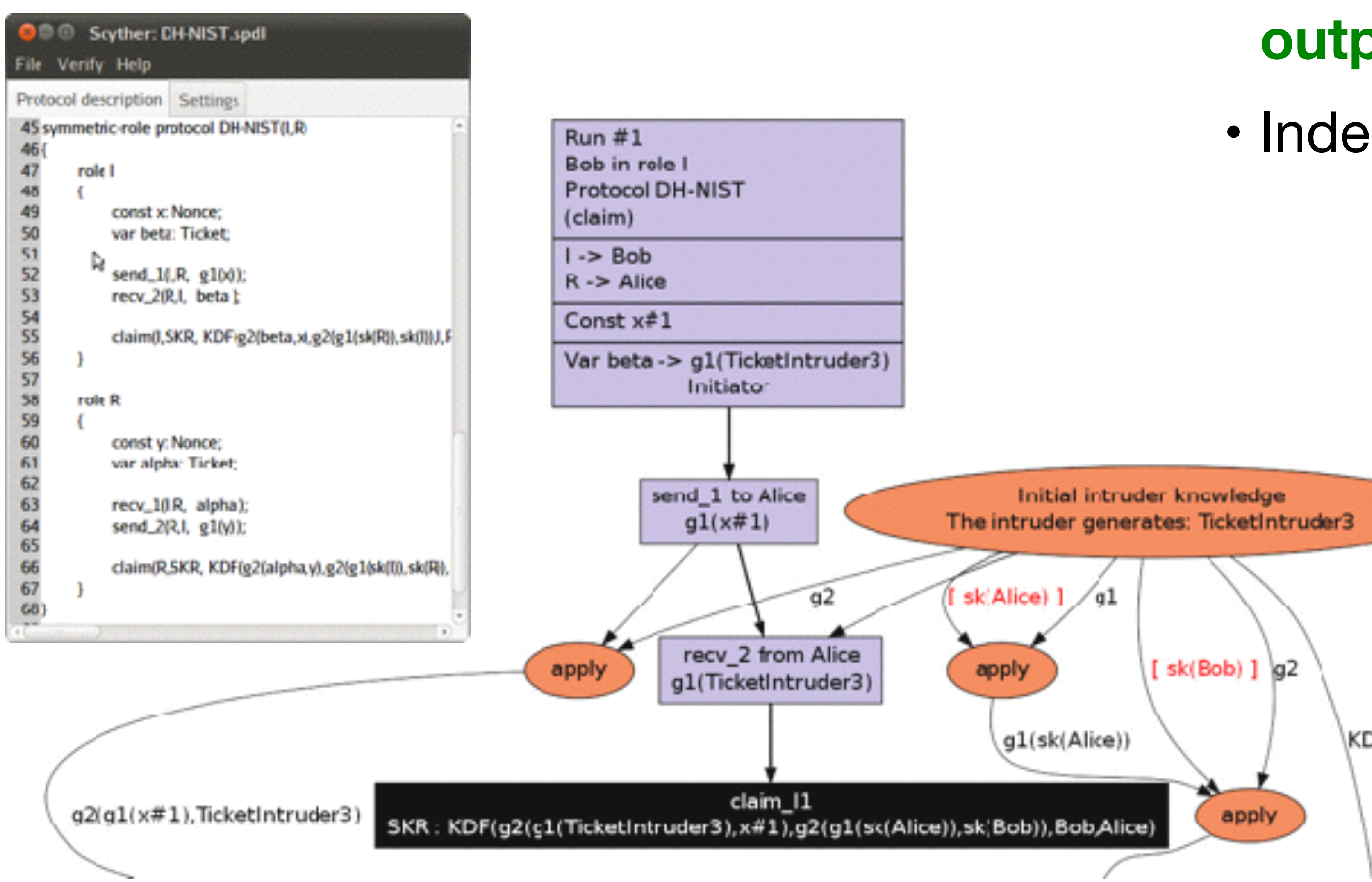
Scyther

Symbolic analysis of security protocols

- Falsification (attack finding)
- Unbounded verification

Scyther-proof

- Embedding of protocol semantics and protocol-independent invariants in the **ISABELLE/HOL** theorem prover
- Algorithm similar to Scyther that **outputs proof script** for Isabelle/HOL
- Independent verifiability



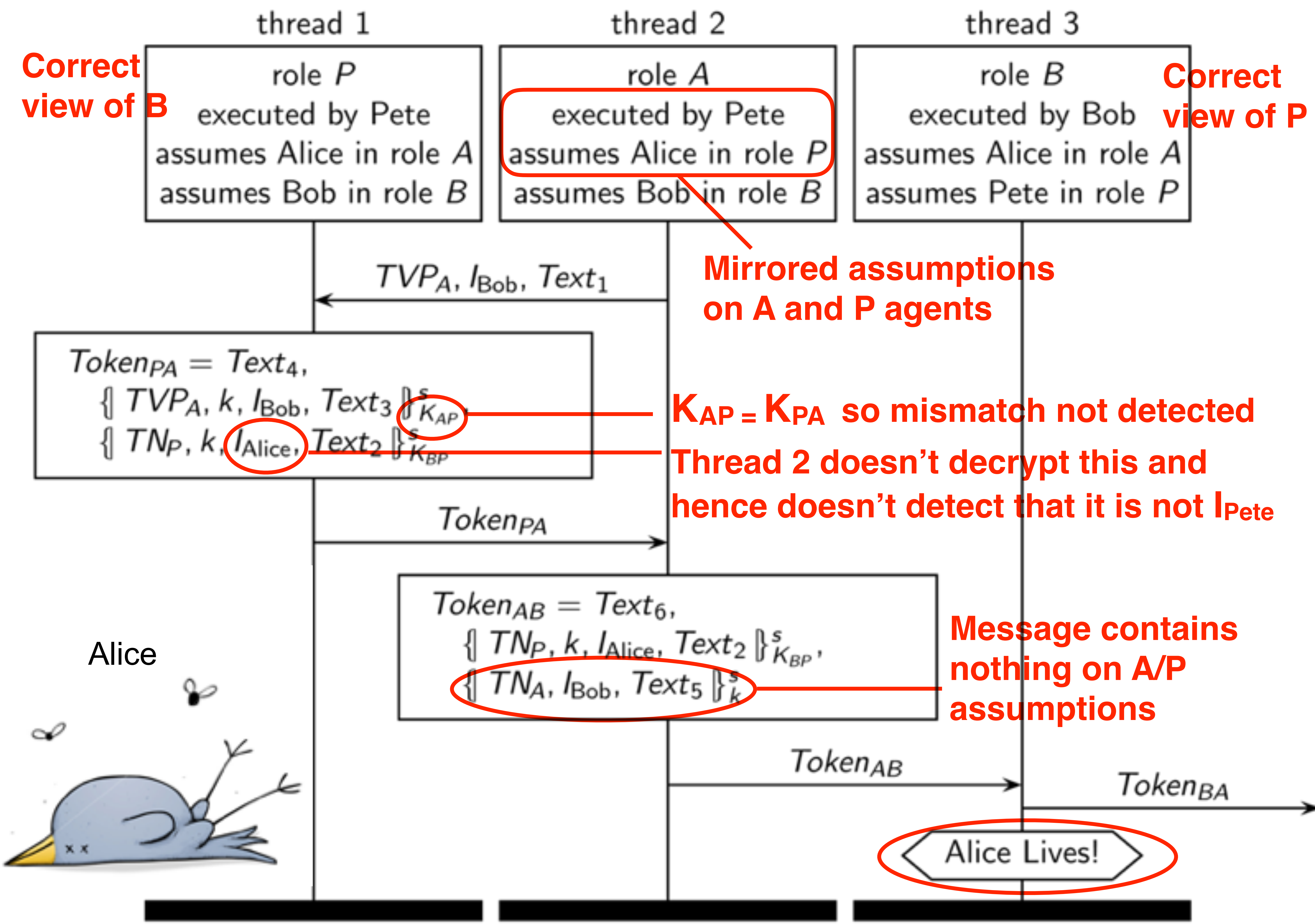
Results

No strong authentication properties

Aliveness < Agreement < Synchronisation

Under some conditions, no authentication

Protocol	Violated property	Assumptions
9798-2-3	A Agreement(B,TNB,Text3)	
9798-2-3	B Agreement(A,TNA,Text1)	
9798-2-3-udkey	A Agreement(B,TNB,Text3)	
9798-2-3-udkey	B Agreement(A,TNA,Text1)	
9798-2-5	A Alive	Alice-talks-to-Alice
9798-2-5	B Alive	
9798-2-6	A Alive	
9798-2-6	B Alive	
9798-3-3	A Agreement(B,TNB,Text3)	
9798-3-3	B Agreement(A,TNA,Text1)	
9798-3-7-1	A Agreement(B,Ra,Rb,Text8)	Type-flaw
9798-4-3	A Agreement(B,TNb,Text3)	
9798-4-3	B Agreement(A,TNa,Text1)	
9798-4-3-udkey	A Agreement(B,TNb,Text3)	
9798-4-3-udkey	B Agreement(A,TNa,Text1)	



thread 1
thread 2
thread 3
Correct view of B
role P
executed by Pete
assumes Alice in role A
assumes Bob in role B
role A
executed by Pete
assumes Alice in role P
assumes Bob in role B
role B
executed by Bob
assumes Alice in role A
assumes Pete in role P
Correct view of P
$TVP_A, I_{Bob}, Text_1$
Mirrored assumptions on A and P agents
$Token_{PA} = Text_4, \{\!| TVP_A, k, I_{Bob}, Text_3 |\!\}^s_{K_{AP}} \{\!| TN_P, k, I_{Alice}, Text_2 |\!\}^s_{K_{BP}}$
$K_{AP} = K_{PA}$ so mismatch not detected
Thread 2 doesn't decrypt this and hence doesn't detect that it is not I_{Pete}
$Token_{PA}$
$Token_{AB} = Text_6, \{\!| TN_P, k, I_{Alice}, Text_2 |\!\}^s_{K_{BP}}, \{\!| TN_A, I_{Bob}, Text_5 |\!\}^s_{k}$
Message contains nothing on A/P assumptions
Alice
$Token_{AB}$
$Token_{BA}$
Alice Lives!

Repairing ISO/IEC 9798

There were numerous design problems!

- Design followed various best-practice principles
- **Example**: Identity of one agent always included to break symmetry of shared keys
- Great, but doesn't work with three parties

We proposed fixes and machine-checked correctness proofs

- Fixes do not require additional cryptography

Scyther-proof generates proof scripts for Isabelle-HOL

- Allows independent verification of results (no need to trust our tool)

Effort

Modeling effort

- ca. 2 weeks
- Abstraction level of standard close to formal models

Generating proof scripts using Scyther-proof

- 20 seconds

Checking correctness of scripts in Isabelle/HOL

- 3 hours (correctness for all protocols used in parallel)

Experience similar with other standards of comparable complexity

- and also with proprietary designs

ISO/IEC Conclusions

International Organization for Standardization

Improving the ISO/IEC 9798 standard

- Old version: **only weak authentication**, sometimes none
- Successful interaction between researchers and standardization committee
- **New version of the standard** released guaranteeing **strong authentication**
- Machine-checked symbolic proofs of standard

More generally

- Automated formal analysis is feasible and useful
- However, tools used were limited
  - No support for Diffie-Hellman & intricate security properties
  - No rekeying, databases, complex control flow

What about protocols orders of magnitude more complex?

Example #2: 5G

New standard for mobile communication, standardized by 3GPP

- Release 15 (5G Phase 1) adopted June 14, 2018

Worldwide commercial service in 2020

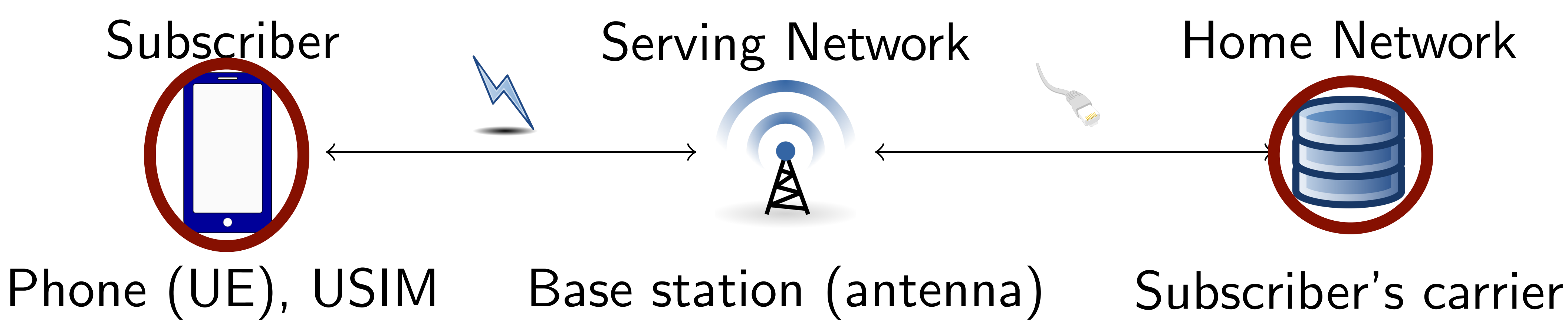
- 5 billion mobile subscribers in 2016
- 60% of world population has 4G access

Numerous protocols including Authentication and Key Agreement (AKA)

D.B., Dreier, Hirschi, Radomirovic, Sasse, Stettler,
A Formal Analysis of 5G Authentication, CCS 2018.

Authentication and Key Agreement

Protocol to authenticate a user's equipment and a serving network and establish shared session keys between them.



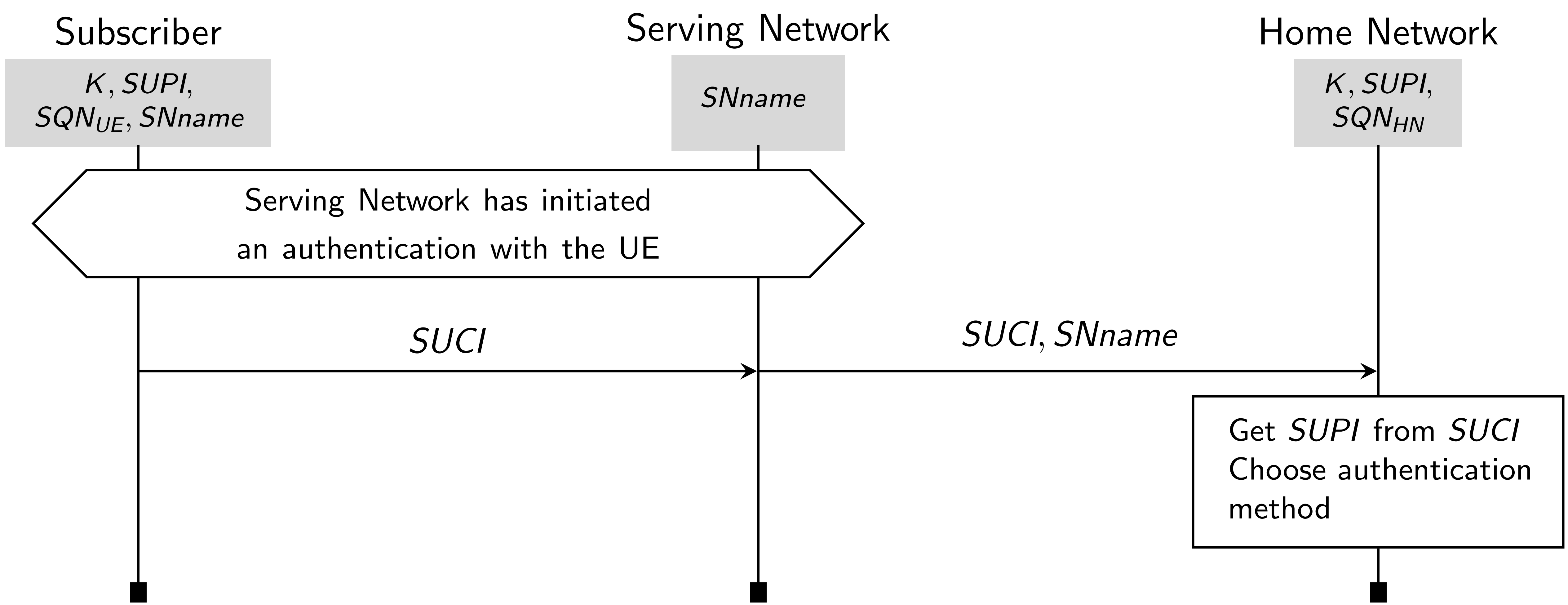
USIM and Home Network share:

- **Symmetric key *K***
- **Permanent identifier *SUPI* (Subscriber Permanent Identifier) used later to derive a *SUCI* (Subscriber Concealed Identifier)**
- **Sequence number *SQN***
- **Home Network's public key pkH_N**

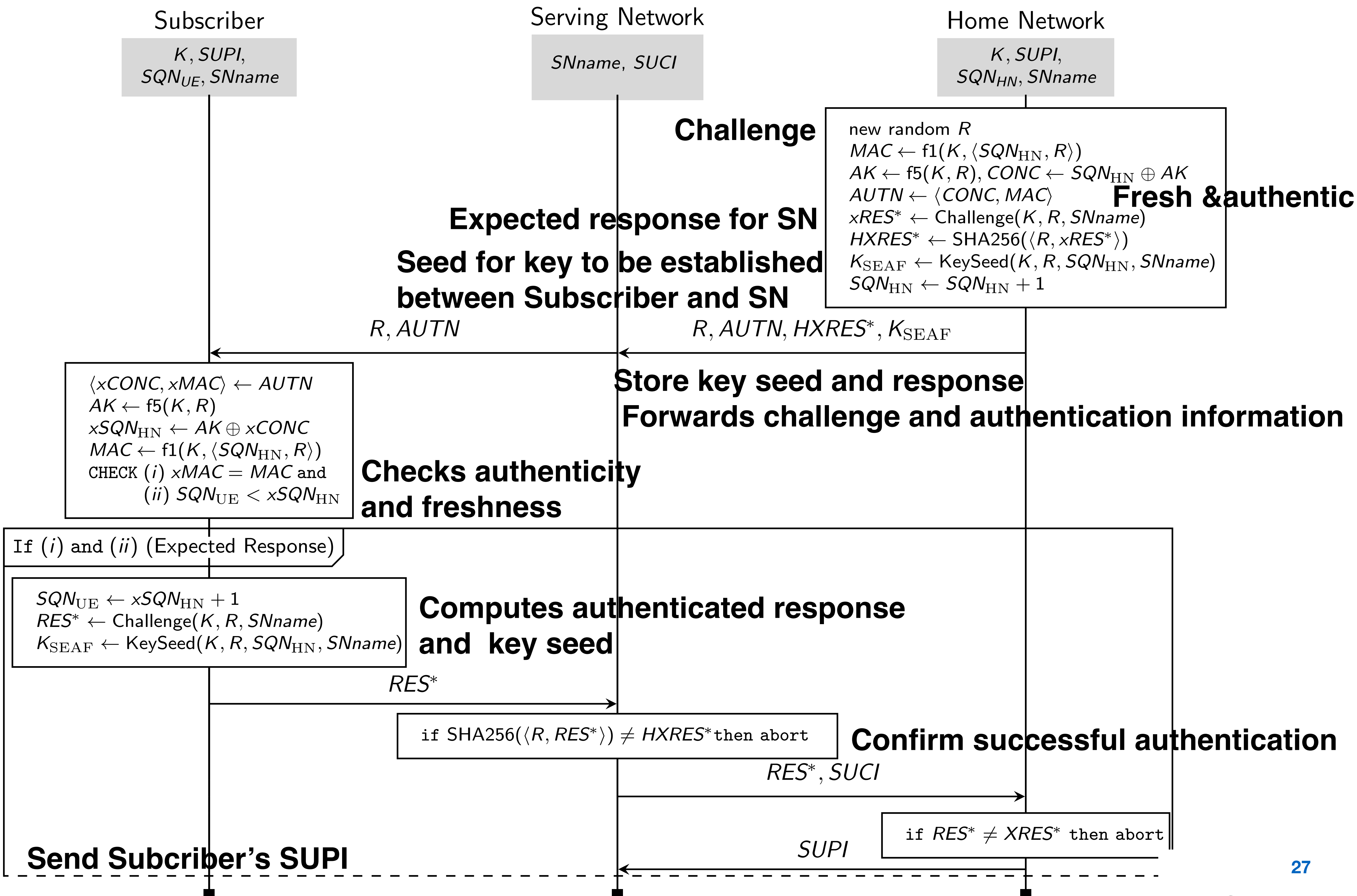
5G Initialization

Subscriber sends his permanent identifier *SUPI* encrypted with Home Network's public key:

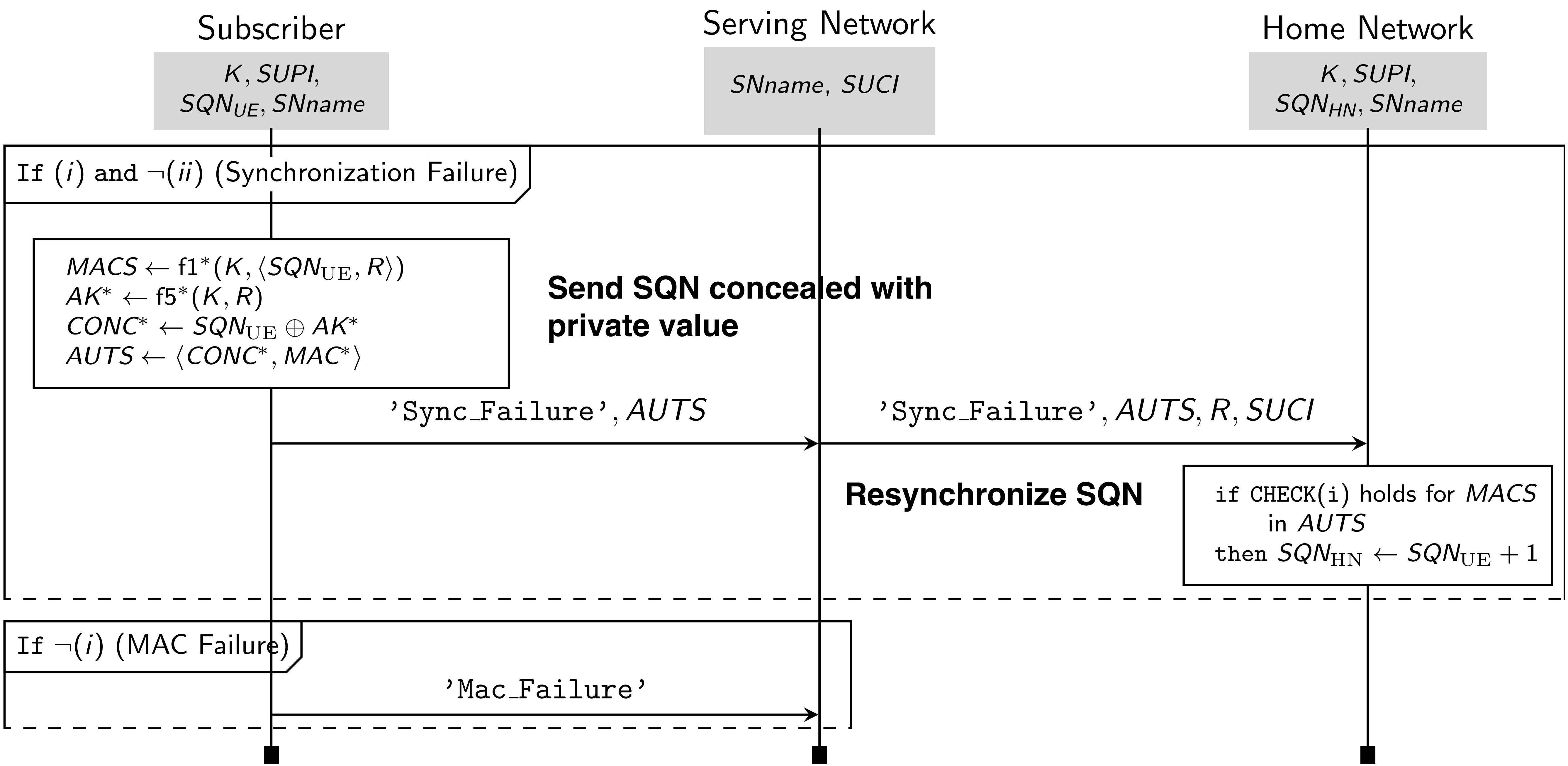
$$SUCI = \langle \text{aenc}(\langle SUPI, R_s \rangle, pk_{\mathrm{HN}}), idHN \rangle$$



AKA Protocol (Successful Authentication Case)



AKA Protocol (Failure Cases)



So is Protocol Secure?

Is home network talking to subscriber or an imposter?

Privacy? Is subscriber traceable and by whom?

Verification extremely challenging

- **Stateful protocol**: sequence numbers and 14 possible protocol states
- Use of **XOR** (a non-convergent theory)
- Privacy requirements are **equivalence properties**
- **Unbounded** number of **sessions**

⇒ Uses recent Tamarin extensions

- Support for **observational equivalence** (for privacy) and **XOR**

Formal Analysis of AKA in Tamarin

Formalized draft v1.0.0 of Release 15 from March 2018

- Followed standardization for ca. 1 year (part time)

Extracted the protocol specification and security goals from 3GPP Technical Specification

- 722 pages over 4 documents

Tamarin model: ~500 lines

Specification of desired goals + lemmas for termination: ~1000 lines, 124 lemmas

Identified minimal set of trust assumptions for each property

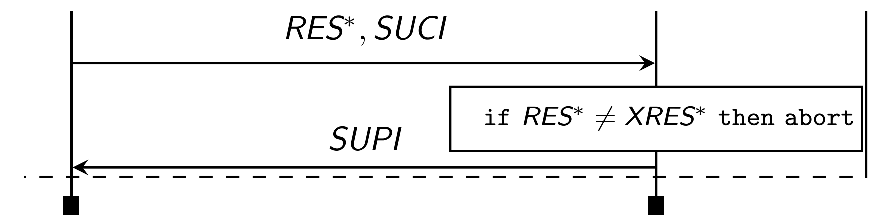
- I.e., strongest possible adversary model

Computation time: 5+ hours (also using “oracle” support)

Results: Authentication

Standard specifies surprisingly few and weak authentication goals

Agreement of Subscribers/SNs on session key K_{SEAF} is not required and fails



- Last message of Home Network to Serving Network not bound to specific session
- Can result in session keys being associated to wrong SUPI
  Concrete attack: use to bill wrong subscriber for services!
- Earlier draft of standard (0.7.1) did not have this flaw

Standard only aims at implicit authentication, whereas many security goals require key confirmation

- Potential for errors in subsequent protocols
- Complicates security analysis
- We proposed and verified two improvements

Results: Security and Privacy

Session key K_{SEAF} remains secret assuming no corrupted long-term keys and secure channel between SN and HN

No perfect forward secrecy for session key K_{SEAF}

Long-term key K remains secret

Subscriber identity *SUPI* remains secret, assuming no corrupted SN or HN

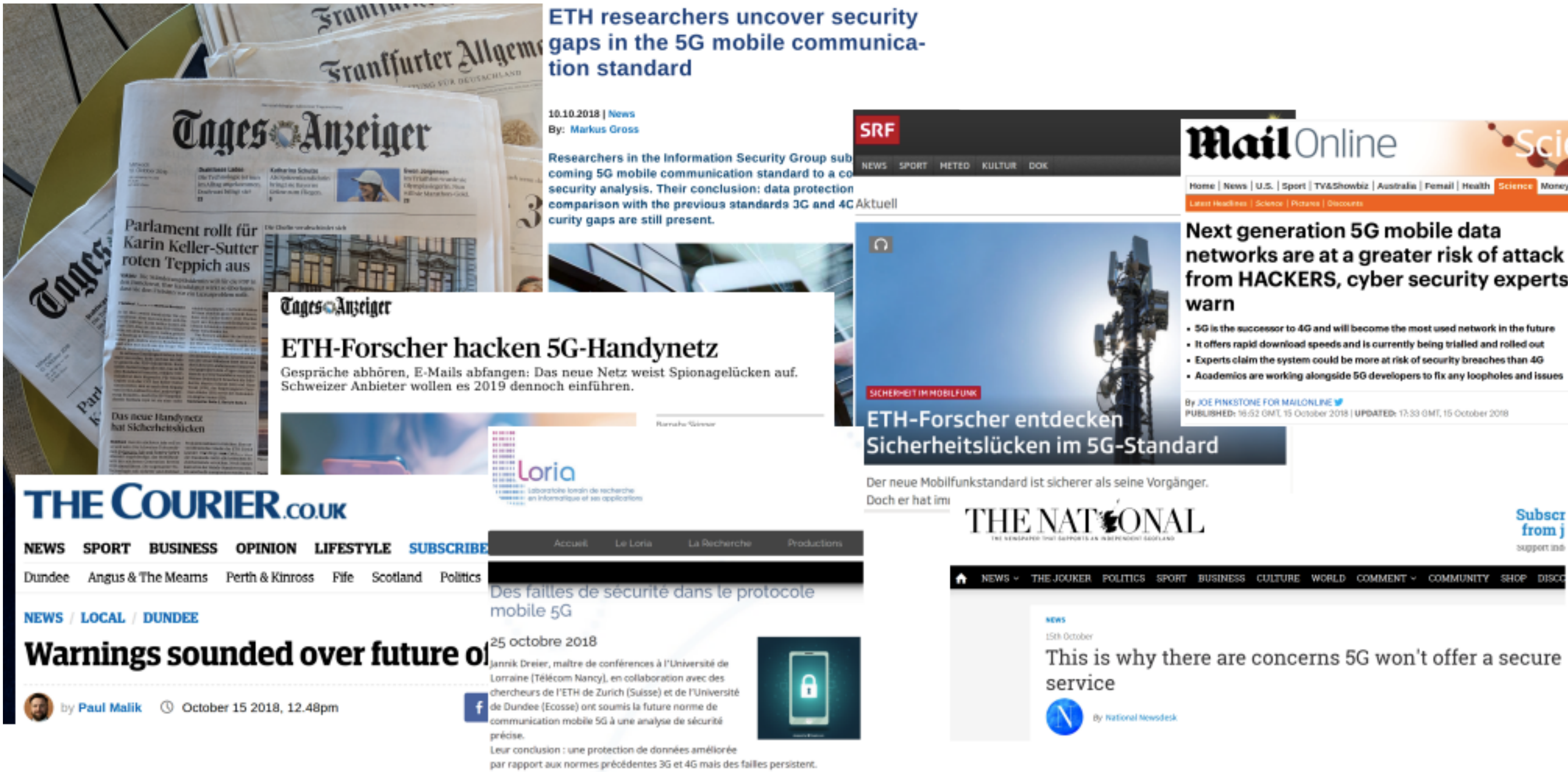
- Defeats IMSI-catchers
- But **insufficient to ensure untraceability**!
  By replaying old messages, an active attacker can use **error messages** to **trace subscribers**
- Fixing this requires major redesign

$MACS \leftarrow \mathrm{f1}^*(K, \langle SQN_{\mathrm{UE}}, R\rangle)$
$AK^* \leftarrow \mathrm{f5}^*(K, R)$
$CONC^* \leftarrow SQN_{\mathrm{UE}} \oplus AK^*$
$AUTS \leftarrow \langle CONC^*, MAC^*\rangle$

$\xrightarrow{\texttt{'Sync\_Failure'}, AUTS}$

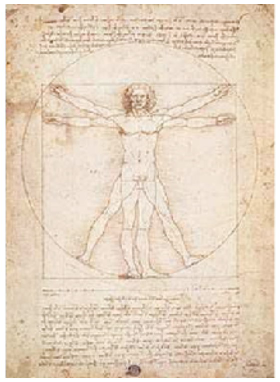
Ongoing discussion with 3GPP on possible fixes

Results: media



Conclusions

Art versus Science



Tools sufficiently advanced that standardization efforts should now be accompanied by formal models and analysis

- Good hygiene: be explicit about protocol, adversary, and properties
- Find errors or produce proofs
- Follow standardization efforts: check modifications for upcoming releases

Research challenges

- **COMPLEXITY, Complexity**, **complexity**
- Improving scope and accuracy
- Education: getting the message out and training engineers

References

- D.B., Cas Cremers, Simon Meier, *Provably Repairing the ISO/IEC 9798 Standard for Entity Authentication*, Journal of Computer Security, 2013.
- Simon Meier, D.B., Cas Cremers. *Efficient Construction of Machine-Checked Symbolic Protocol Security Proofs*, Journal of Computer Security 2013.
- D.B., Cas Cremers, Kunihiko Miyazaki, Sasa Radomirovic, Dai Watanabe. *Improving the Security of Cryptographic Protocol Standards, IEEE Security and Privacy*, 2015.
- D.B., Cas Cremers, Cathy Meadows, *Model Checking Security Protocols*, Handbook of Model Checking, 2018.
- D.B. Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, Ralf Sasse, Vincent Steiler, *A Formal Analysis of 5G Authentication*, CCS 2018.
- Benedikt Schmidt, Simon Meier, Cas Cremers, D.B., *Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties*, CSF 2012.