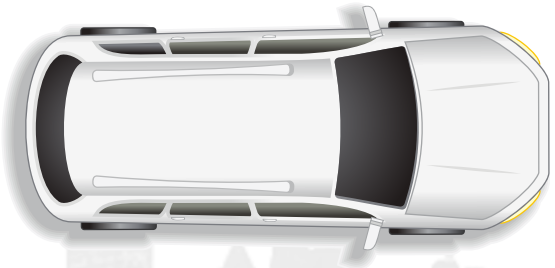
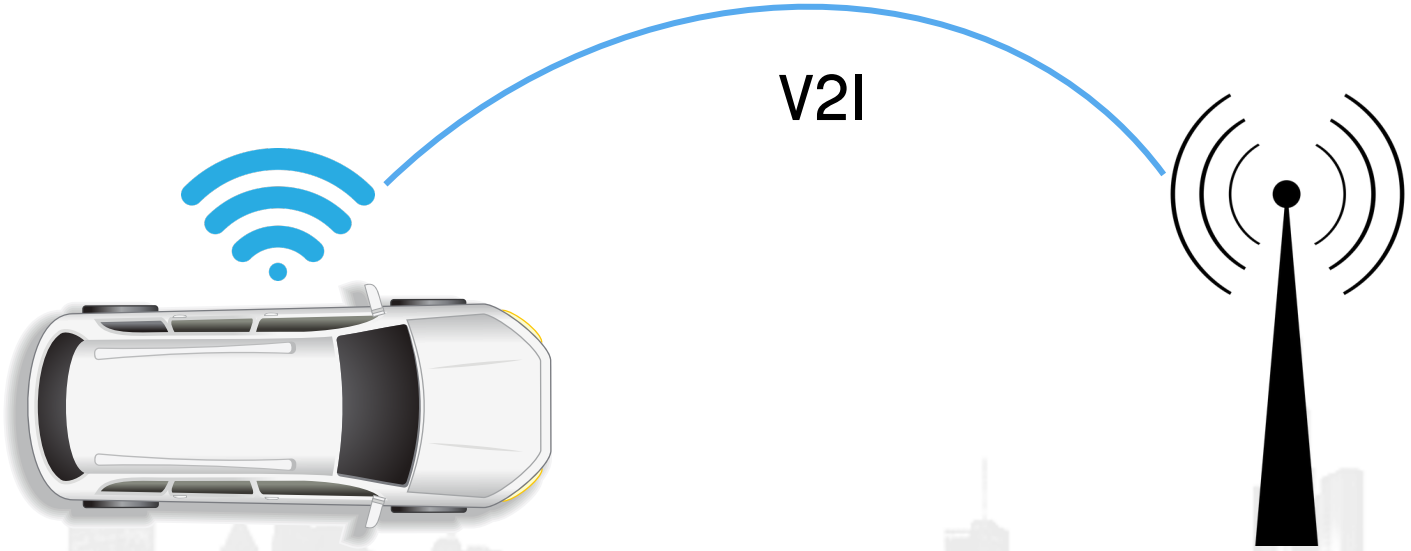
Exploiting Partial Order of Keys to Verify Security of a Vehicular Group Protocol

Felipe Boeira and Mikael Asplund

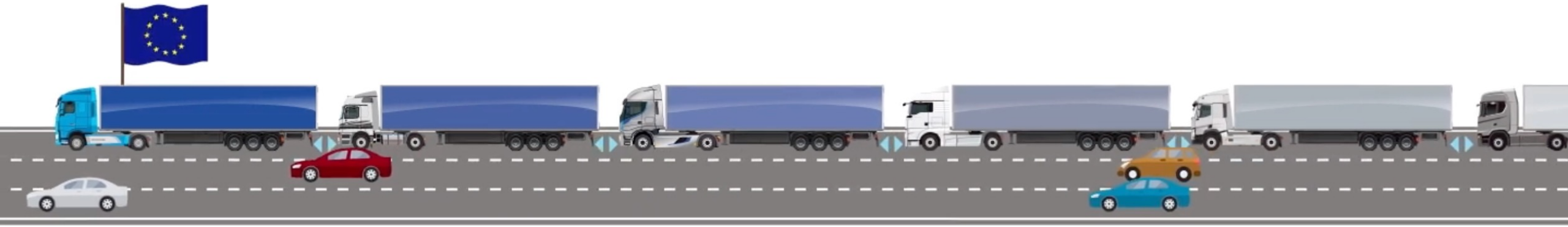
LINKÖPING UNIVERSITY




V2I


V2V
V2I

Ensemble – Multi-Brand Truck Platooning



Source: Ensemble

Problem

- Cyber-physical systems are safety-critical and security vulnerabilities may create safety hazards
- Protocol design is complex and may introduce security weaknesses in the context of cooperative driving
- How to assure that the expected security properties are guaranteed?

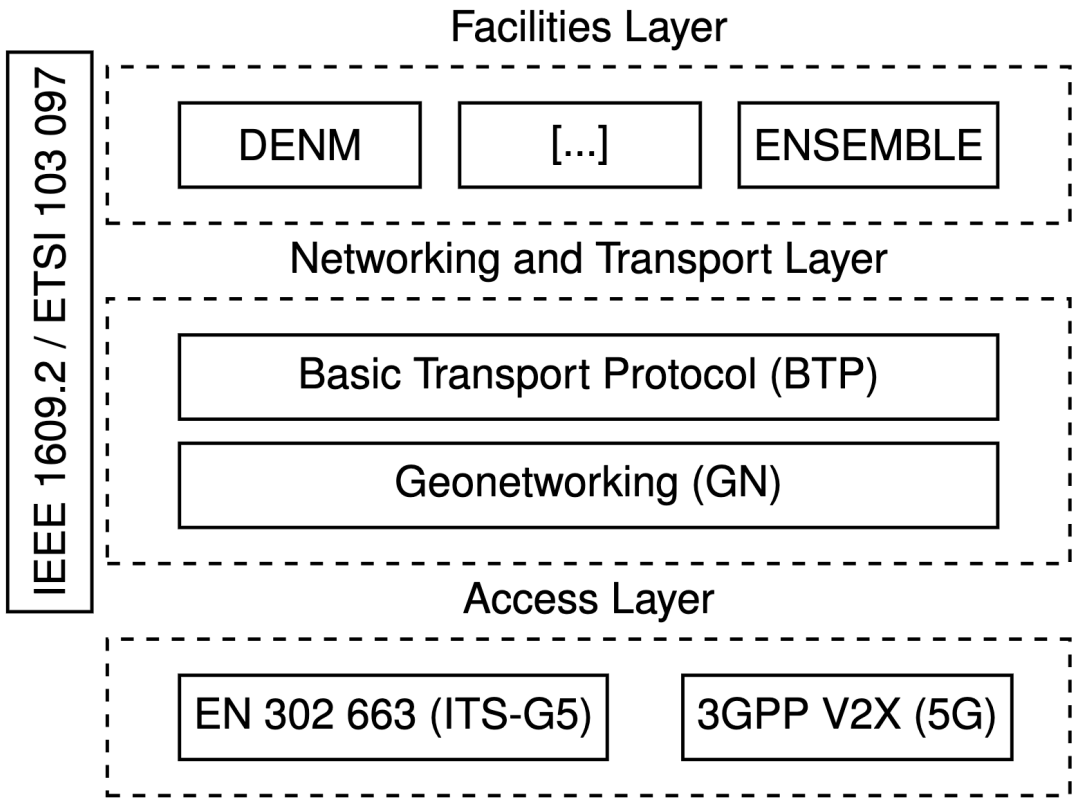
Outline

- Ensemble Protocol
- Ensemble Modeling and Proving Strategy
- Results
- Conclusion

Ensemble Protocol

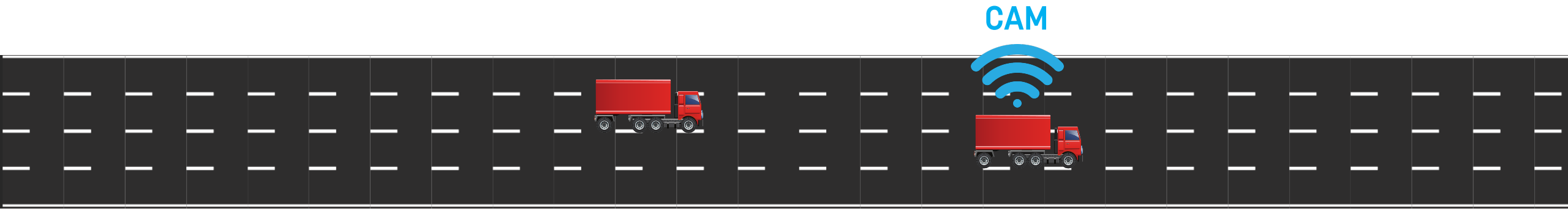
Protocol Stack Overview

- IEEE 1609.2 defines the low-level data structures to achieve security
  - Key encapsulation for recipients
- ETSI 103 097 defines profiles over the data structures for distinct applications
- Ensemble extends the profiles for the platooning application
- In general, signing is applied in Geonetworking and encryption at the Facilities layer



Platooning Procedures

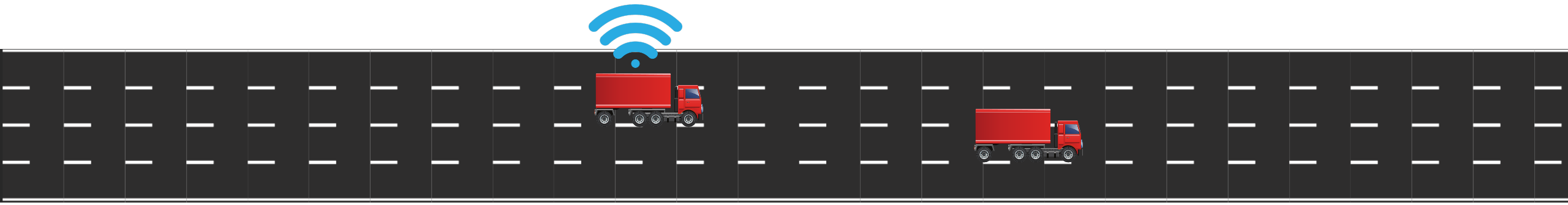
$APP \leftarrow$ 'CAM'
$GN \leftarrow APP, \mathsf{sign}(APP, ltk_n), Cert_n$



Platooning Procedures

generate key pair $jrek$, $\mathsf{pk}(jrek)$
$APP \leftarrow$ `'JoinRequest'`, $\mathsf{pk}(jrek)$, n
$GN \leftarrow APP, \mathsf{sign}(APP, ltk_{n+1}), Cert_{n+1}$

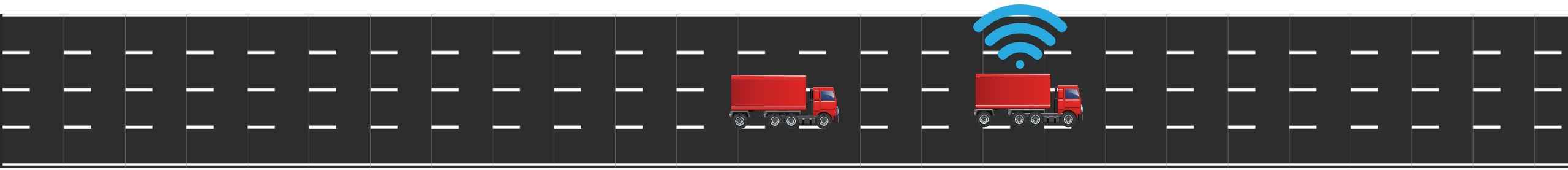
Join Request



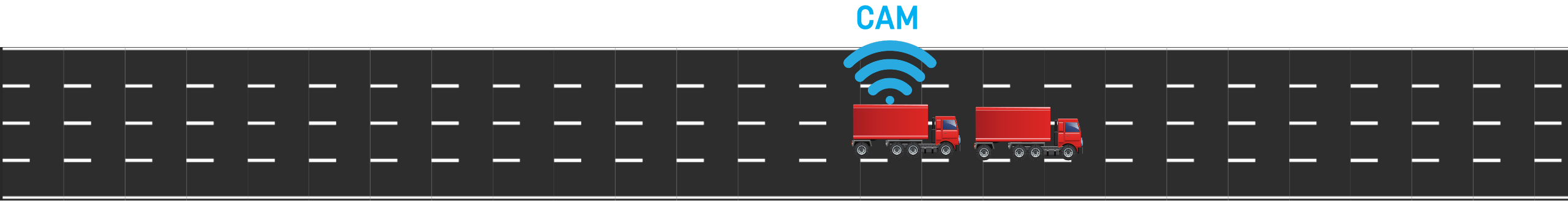
Platooning Procedures

generate key pgk if $n = 1$
generate keys $eJoin, ppk_n$
$rekRecipInfo \leftarrow h(\mathsf{pk}(jrek)), \mathsf{aenc}(eJoin, \mathsf{pk}(jrek))$
$ciphertext \leftarrow \mathsf{senc}(\langle ppk_n, pgk \rangle, eJoin)$
$APP \leftarrow$ `'JoinResponse'`$, rekRecipInfo, ciphertext, P, n + 1$
$GN \leftarrow APP, \mathsf{sign}(APP, ltk_n), Cert_n$

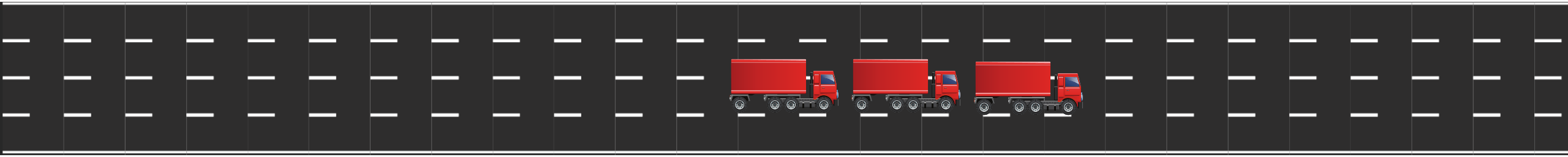
Join Response



Platooning Procedures



Platooning Procedures



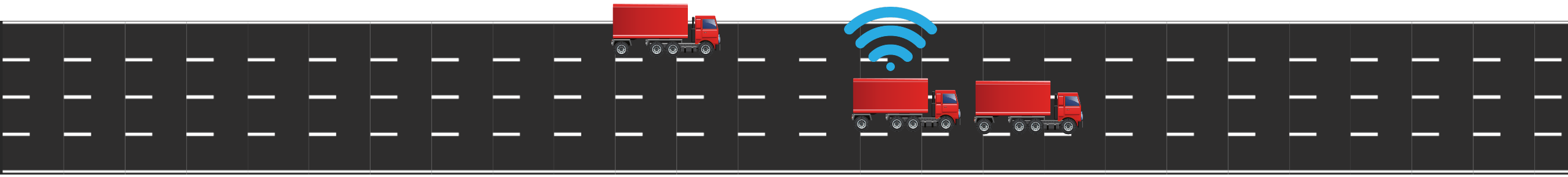
Platooning Procedures

generate key $eLeave$
$symmRecipInfo \leftarrow h(pgk), \mathsf{senc}(eLeave, pgk)$
$leaveMsg \leftarrow 3, position, \texttt{'Reason'}$
$ciphertext \leftarrow \mathsf{senc}(leaveMsg, eLeave)$
$APP \leftarrow \texttt{'Leave'}, symmRecipInfo, ciphertext, P$
$GN \leftarrow APP, \mathsf{sign}(APP, ltk_3), Cert_3$

Platooning Procedures

generate key $eKUR$
$symmRecipInfo \leftarrow h(pgk), \mathsf{senc}(eKUR, pgk)$
$ciphertext \leftarrow \mathsf{senc}(\texttt{'KUR'}, eKUR)$
$APP \leftarrow \texttt{'KUR'}, symmRecipInfo, ciphertext, P$
$GN \leftarrow APP, \mathsf{sign}(APP, ltk_2), Cert_2$

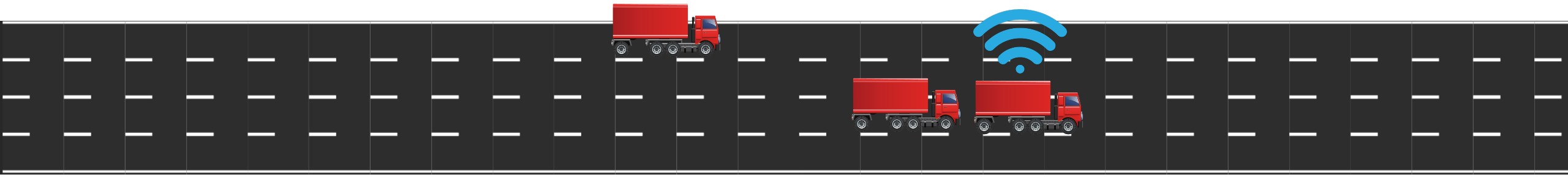
Key Update Request



Platooning Procedures

generate keys $eKupdate, pgkUpdate$
$symmRecipInfo \leftarrow h(ppk_1), \mathsf{senc}(eKupdate, ppk_1)$
$ciphertext \leftarrow \mathsf{senc}(\langle \texttt{'KeyUpdate'}, pgkUpdate \rangle, eKupdate)$
$APP \leftarrow \texttt{'KeyUpdate'}, symmRecipInfo, ciphertext, P$
$GN \leftarrow APP, \mathsf{sign}(APP, ltk_1), Cert_1$

Key Update



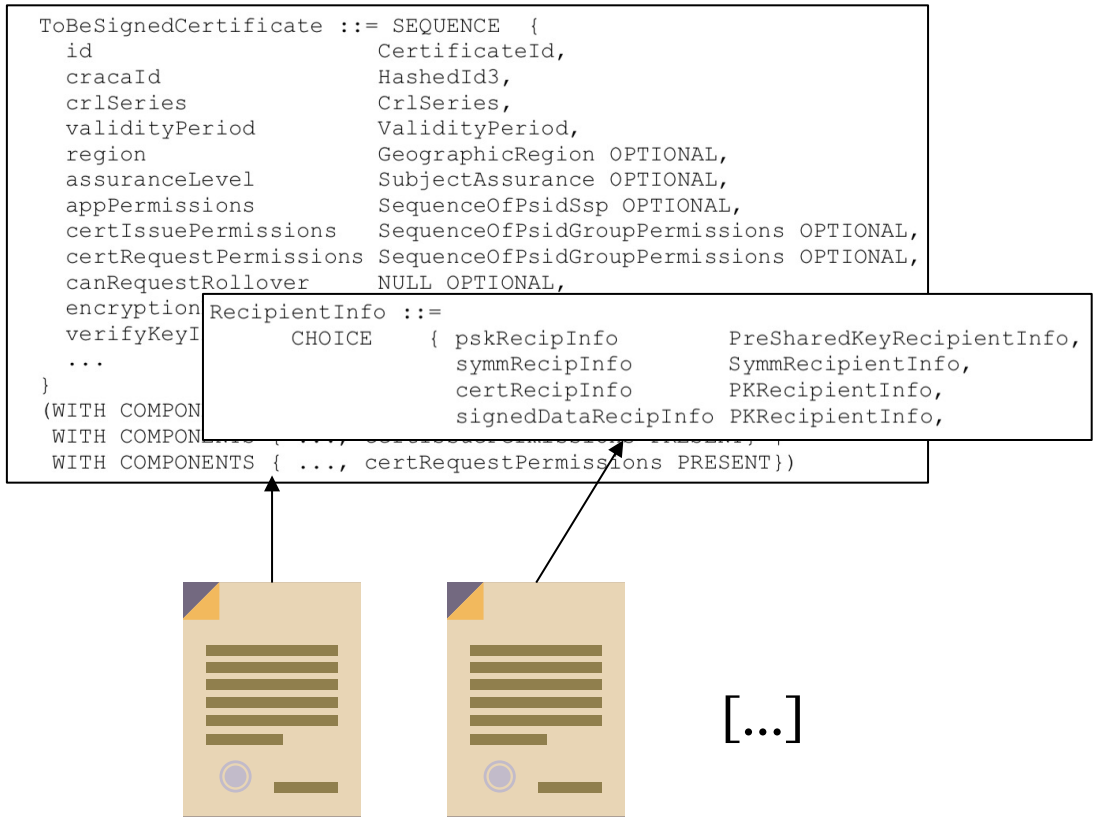
Ensemble Modeling and Proving Strategy

Protocol Security Verification

- Computational vs **symbolic model**
- Tool of choice: Tamarin 
- Model protocol as multiset rewriting rules
- Specify security properties as temporal first-order logic formulas

ASN.1 Sample Packet Generation

- Specifications span across multiple cross-referenced documents and hundreds of pages
- Main documents: IEEE 1609.2 (and its amendments) + ETSI 103 097 + Ensemble D.9 Security
- Leverage ASN.1 specifications to aid in refining the Tamarin model



Main Modeling Aspects

- We consider an honest CA and honest participants in the platoon, but other trucks may be controlled by the attacker

- We specify lemmas to capture liveness, secrecy, and authenticity properties:
  - Liveness: protocol is executable
  - Secrecy of all keys throughout the lifecycle of the protocol
  - Aliveness, weak agreement, and non-injective agreement authenticity properties

- Two model variants:
  - Static: Models three vehicles that form a platoon with an unbounded number of sessions
  - Dynamic: Models an unbounded number of platoons with an unbounded number of vehicles, however, vehicles engage in one session

Non-Termination

- Tamarin heuristics could not terminate the proof computations within the allocated resources
- ”Selecting the next case distinction rule or precomputed case distinction to apply to a constraint system constitutes the key difficulty in the construction of a security proof”*
- Order plays a major role during proving

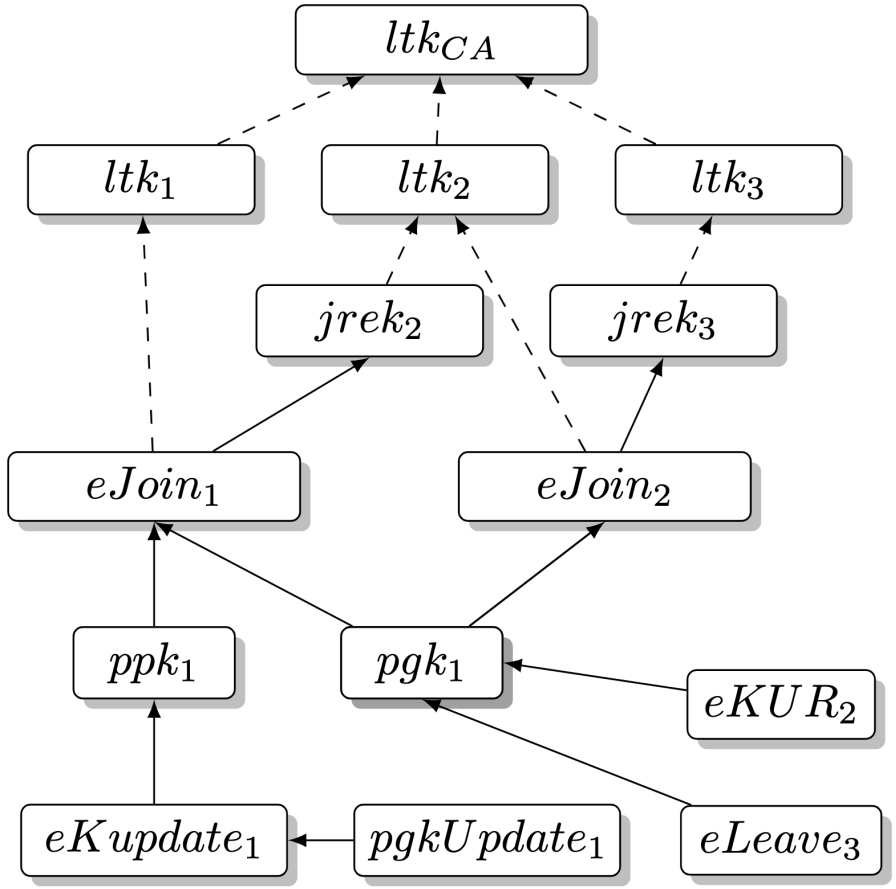
* S. Meier, “Advancing automated security protocol verification,” Ph.D. dissertation, ETH Zurich, 2013.

Relation rules

- Consider a set K of keys and two keys $k_A, k_B \in K$

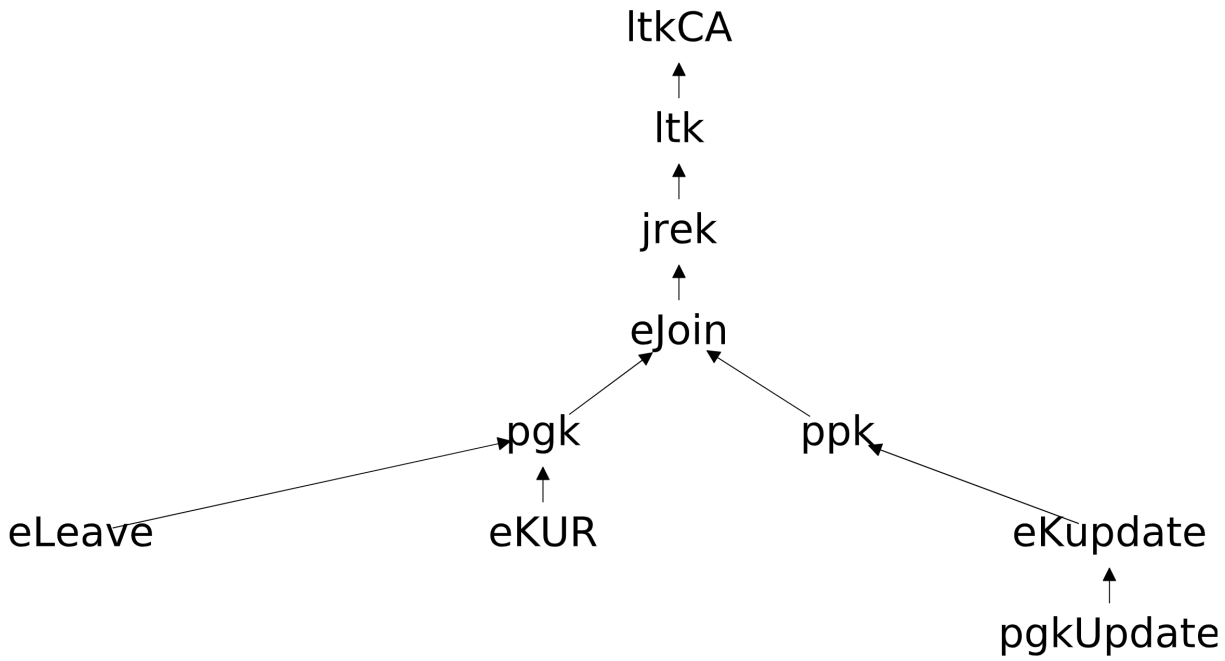
- Secrecy relation $\rightarrow \subseteq K \times K$:
  We consider that $k_A \rightarrow k_B$ whenever $senc(k_A, k_B)$ or $aenc(k_A, pk(k_B))$ occurs in a message sent over the network **(rule 1)**.

- Authenticity relation $\dashrightarrow \subseteq K \times K$:
  To define our authenticity relation, we consider **compromising** a term p as either revealing it or being able to generate a p' that will be accepted by other nodes as p.
  We consider that $k_A \dashrightarrow k_B$ holds if **compromising** k_B allows the attacker to create another key k'_A that will be accepted by the other nodes as the legitimate k_A, which thereby becomes **compromised (rule 2)**.



Key Hierarchy Extraction

- We have created a proof-of-concept extractor of key dependencies from Tamarin models
- Identify classes of keys instead of individual instances, uses maude as backend
- Exploit the linear extension to order the lemma reuses and goal selection in the oracle



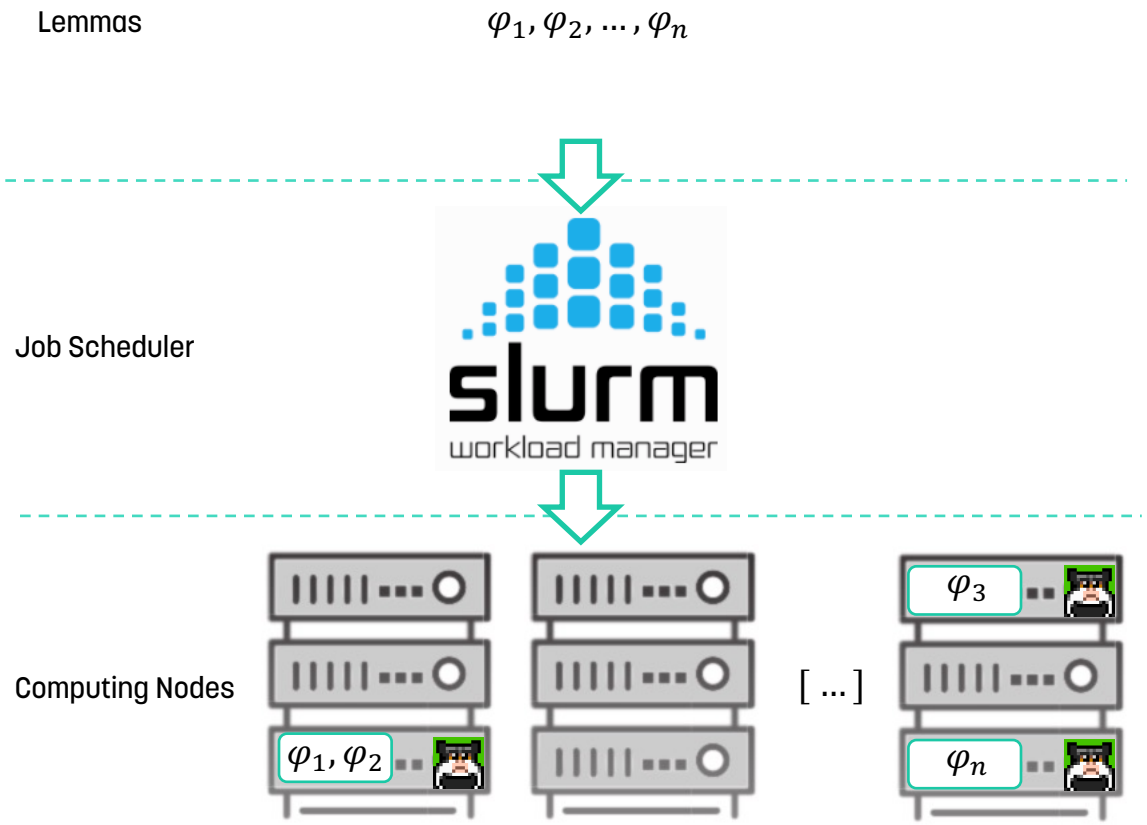
Results

Misbinding Attack

- Occurs when two honest parties establish a common session key without a consistent view of each other's identities
- IEEE 1609.2 contains vulnerable data structure which is used in Ensemble
- Mitigated by including an intended receiver in the application payload
- Must be checked in an implementation in order to avoid vulnerabilities, clarified in the latest Ensemble security specifications

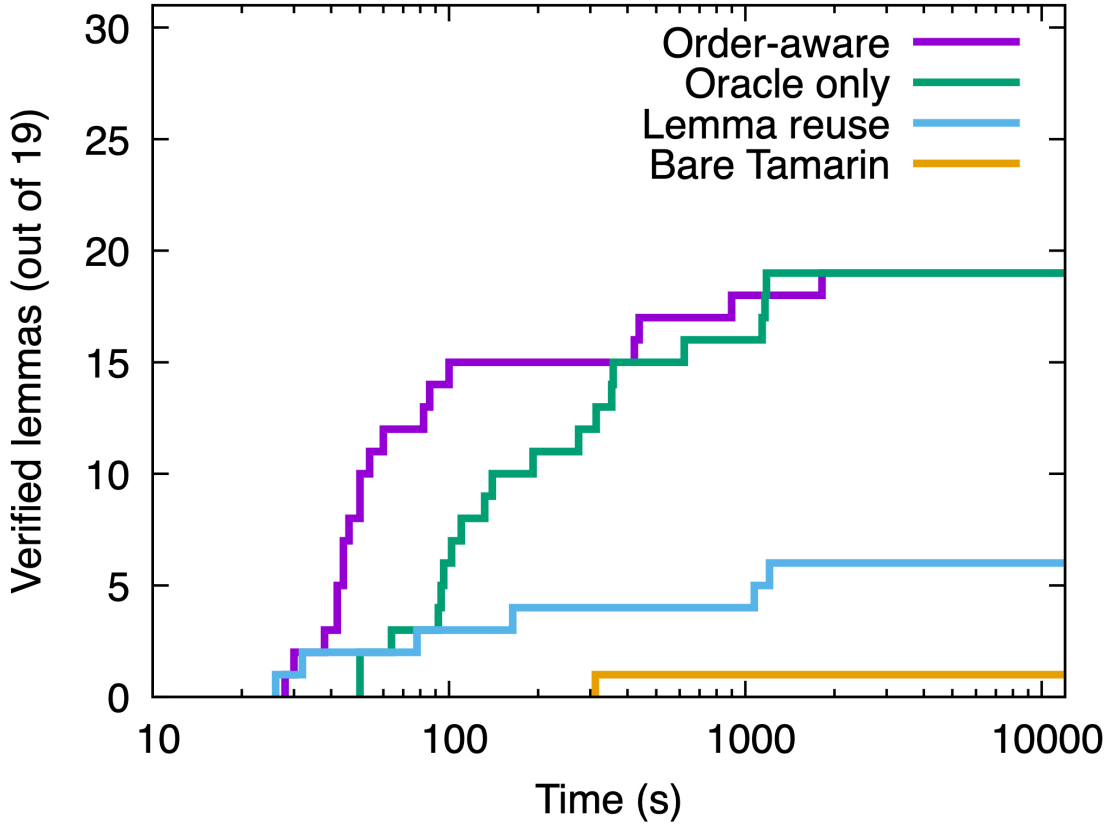
Infrastructure for Proving

- Swedish National Supercomputer Center
  - Sigma Cluster
  - Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz with 32 cores
  - 96 GiB RAM
- Resource allocation for each proof
  - 8 cores for 2 hours
  - 22 GiB of RAM



Proving Results

- Proving was not possible in automatic mode
- *'Oracle only'* leverages key ordering from the hierarchy but not lemma reusability
- *'Order-aware'* uses oracle + reusable lemmas with correct ordering and can prove lemmas faster



Proof method	Oracle	Reuse	Liveness	Secrecy	Authenticity
Bare TAMARIN	N	N	1/1	0/15	0/3
Lemma reuse	N	Y	1/1	4/15	1/3
Oracle only	Y	N	1/1	15/15	3/3
Order-aware	Y	Y	1/1	15/15	3/3

Syntethic Protocol

- We define an experiment that consists of two agents that have a pre-shared key
- In each interaction, they generate a fresh key and encrypt it with the last received key from the partner to transmit it in the network
- Attempt to prove the secrecy of keys under distinct key depths (8 cores, 20 GiB RAM, 30 minutes)

Key depth	Without reuse (ordered by dependency)	With reuse (random order)	With reuse (ordered by dependency)
2	✔	✔	✔
4	✖	3/10	✔
6	✖	2/10	✔
8	✖	✖	✔
10	✖	✖	✖

Conclusion

- We have performed a security analysis of Ensemble, a multi-brand truck platooning protocol in pre-standardisation effort
- To tackle the verification complexity, we have employed a strategy based on the hierarchy and dependencies among criptographic keys
- We believe that developing this strategy further and integrating with a tool such as Tamarin can be useful in several ways, e.g. finding the conditions for key secrecy automatically
- We have reported a potential misbinding attack, which is fixed in the latest specifications, and verified secrecy and authenticity properties

Thank You

www.liu.se

felipeboeira.eu
felipe.boeira@liu.se

LINKÖPING UNIVERSITY