ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Modeling and Analyzing Security in the Presence of Compromising Adversaries

Cas Cremers
Joint work with
David Basin

Overview

- Background and problem
- Adversaries in existing security notions
- A formal symbolic model
- Tool support
- Results and demo
- Observations
- Future work & conclusions

Security protocols

- Small distributed programs to communicate over untrusted networks
  - One building block: Authenticated Key Exchange

- Multiple sessions (threads) per agent in parallel

- Some agents may be compromised / evil

Symbolic Analysis of security protocols

- Was used to find flaws in many protocols
  - Canonical example: Needham-Schroeder protocol

- Strong abstraction
  - Assumes cryptography is perfect
  - Abstract terms instead of bit strings
  - Possibilistic reasoning

  - Still, properties like secrecy undecidable

- Several logics and automatic tools available
  - AVISPA, ProVerif, *Scyther*, ...

Core of symbolic model

- Labeled transition system
  - Models agents' threads and the adversary
  - Many security problems become reachability problems

  - State (tr,IK,th):
    - Tr : events that have occurred before
    - IK : current adversary knowledge
    - Th : map of thread (session) identifiers to remaining steps

$$\frac{th(tid) = \langle \mathsf{send}(m)\rangle\hat{\ } l}{(tr, IK, th) \longrightarrow (tr\hat{\ }\langle(tid, \mathsf{send}(m))\rangle, IK \cup \{m\}, th[l \leftarrow tid])}\,[\mathsf{send}]$$

$$\frac{th(tid) = \langle \mathsf{recv}(pt)\rangle\hat{\ } l \qquad IK \vdash \sigma(pt) \qquad dom(\sigma) = FV(pt)}{(tr, IK, th) \longrightarrow (tr\hat{\ }\langle(tid, \mathsf{recv}(\sigma(pt)))\rangle, IK, th[\sigma(l) \leftarrow tid])}\,[\mathsf{recv}]$$

Demo

Many possibilities for improving models

- Examples:
  - Scaling up to full protocol suites
  - Computational soundness
  - … etc

- Security guarantees (Adversary model)
  - Adversary controls network
  - In general: only static corruption considered
  - Property:

$$\mathrm{Honest}(A) \wedge \mathrm{Honest}(B) \Rightarrow \neg(\mathcal{A} \text{ knows } SessionKey_{A,B})$$

Adversary models and protocols evolved

These protocols are all „correct“ in symbolic models:

$$A \rightarrow B : A, \{B, na\}_{pk(B)}$$
$$B \rightarrow A : \{B, H(na), nb, K\}_{pk(A)}$$
$$A \rightarrow B : \{H(nb)\}_{pk(B)}$$

$$sessionkey : K$$

BKE

Signed DH

$$A \rightarrow B : \{B, g^{na}\}_{sk(A)}$$
$$B \rightarrow A : \{A, g^{nb}\}_{sk(B)}$$

$$sessionkey\ A : (g^{nb})^{na}$$
$$sessionkey\ B : (g^{na})^{nb}$$

HMQV

$$A \rightarrow B : g^{na}$$
$$B \rightarrow A : g^{nb}$$

$$sessionkey\ A : H(((g^{nb})(pk(b))^{e})^{na+d(sk(a))})$$
$$sessionkey\ B : H(((g^{na})(pk(a))^{d})^{nb+e(sk(b))})$$

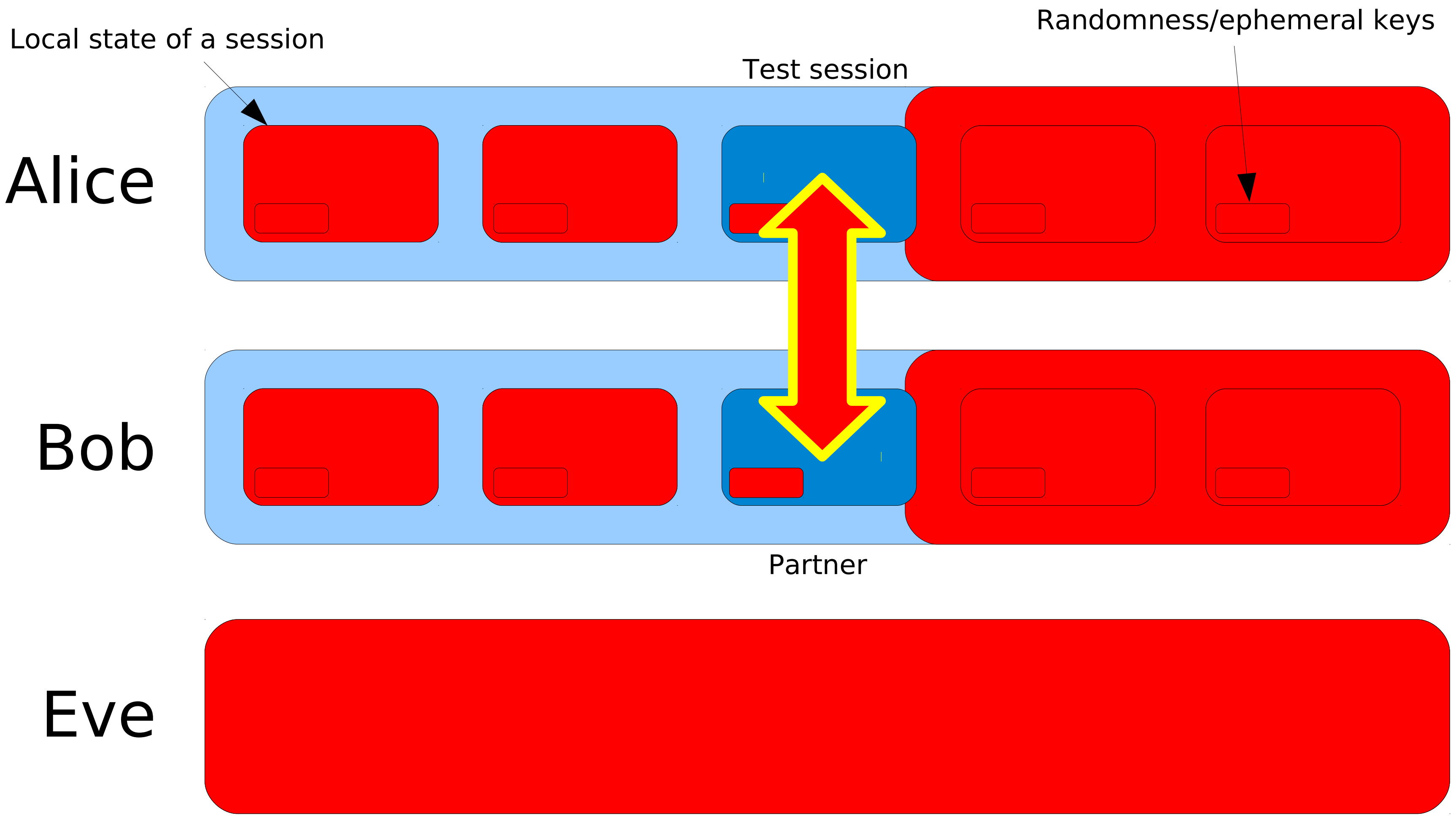
Adversaries in cryptographic models

- Stronger adversary notions in e.g. AKE security
  - Motivated the development of new protocols
  - New protocols in this class are proposed *regularly*

- Compromise of
  - Long-term keys at some point in time (dynamically)
  - Session keys (cryptanalysis?)
  - Session-state (freeze memory?)
  - Randomness/ephemeral keys (leaky RNG?)

- Idea: Extend symbolic methods
  - Generic definitions
  - Tool support

Compromising adversaries: intuition



Modeling compromising adversaries

- Many different notions exist in AKE literature
  - Monolithic definitions of 'security notions'
  - Bellare Rogaway 93,95; Bellare Pointcheval Rogaway 2000; Shoup; Canetti Krawczyk 2001; Canetti (UC) 2001-... ; LaMacchia et al 2007; ...
- No agreement in community about the many of the details
  - But details influence protocol judgements!
- Roughly: all models are incomparable

Methodology

- Investigate security notions in cryptographic literature

- Extract common elements
- Abstract from modeling details
  - Execution models, partnering, atomicity of receive-send, ...
- Generalize where possible

- Provide model and, if possible, tool support

Dimensions of compromise

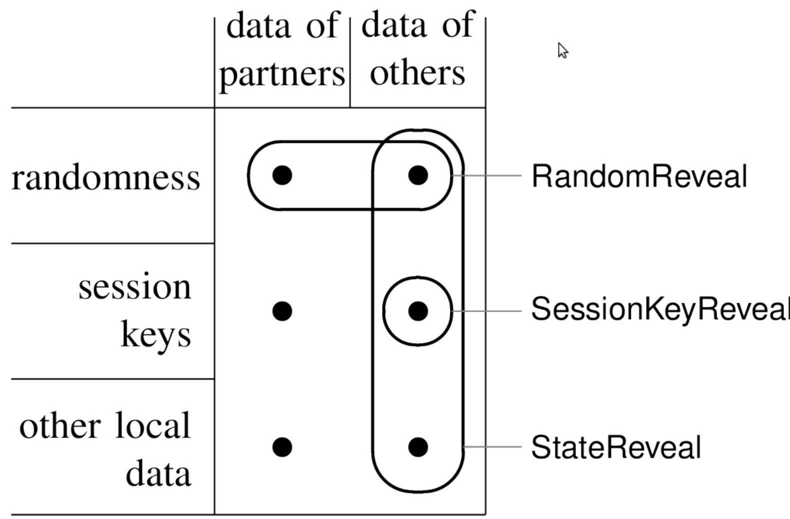
- When
  - Before, during, or after Test
- Whose data
  - Actor, partners, and others
- Which data
  - Long-term keys, session keys, randomness, session-state

- First distinction: *long-term* versus *short-term* data

Reveal long-term data: *whose, when*

	key of actor	keys of peers	keys of others
before Test thread			
during Test thread			
after Test thread			

Reveal short-term data: *whose, which*



$$\frac{tid \neq \text{Test} \qquad tid \notin Partner(tr)}{(tr, IK, th) \longrightarrow (tr\hat{\ }\langle(\text{tid}_{\mathcal{A}}, \mathsf{SessionKeyReveal}(tid))\rangle, IK \cup set((tr{\downarrow}tid){\lfloor}\mathsf{sessionkey}), th)}\,[\mathsf{SessionKeyReveal}]$$

$$\frac{tid \neq \text{Test} \qquad tid \notin Partner(tr) \qquad th(tid) \neq \langle\rangle}{(tr, IK, th) \longrightarrow (tr\hat{\ }\langle(\text{tid}_{\mathcal{A}}, \mathsf{StateReveal}(tid))\rangle, IK \cup last((tr{\downarrow}tid){\lfloor}\mathsf{state}), th)}\,[\mathsf{StateReveal}]$$

$$\frac{}{(tr, IK, th) \longrightarrow (tr\hat{\ }\langle(\text{tid}_{\mathcal{A}}, \mathsf{RandomReveal}(tid))\rangle, IK \cup set((tr{\downarrow}tid){\lfloor}\mathsf{generate}), th)}\,[\mathsf{RandomReveal}]$$

Results in a hierarchy of adversary models

(... 96 adversaries)

Approximating existing models

	Adversary rules							
	Long-term data				Short-term data			
	Owner		Timing		Type			
Name	others	actor	after	aftercorrect	SessionKey	State	Random	Origin of model
Adv_{EXT}								Dolev-Yao (external)
Adv_{INT}	✓							Dolev-Yao (internal) [32]
Adv_{CA}		✓						Key Compromise Impersonation [24]
Adv_{AFC}				✓				Weak Perfect Forward Secrecy [26]
Adv_{AF}			✓	✓				Perfect Forward Secrecy [19,35]
Adv_{BPR}					✓			BPR2000 [5]
Adv_{BR}	✓				✓			BR93 [6], BR95 [7]
Adv_{CKw}	✓	✓		✓	✓	✓		CK2001-wPFS [26]
Adv_{CK}	✓		✓	✓	✓	✓		CK2001 [13]
$Adv_{\mathrm{eCK\text{-}1}}$	✓				✓		✓	eCK [29]
$Adv_{\mathrm{eCK\text{-}2}}$	✓	✓		✓	✓			

Pure properties versus adversaries

- Side effect:
  - Split security property (or notion) into adversary model and „pure“ security property

Security property	Adversary model	Pure security property
Secrecy	{}	Secrecy
Secrecy (Dolev-Yao)	{ LKRothers }	Secrecy
Perfect Forward Secrecy	{ LKRafter }	Secrecy
KCI resilience	{ LKRactor }	Authentication

Tool support

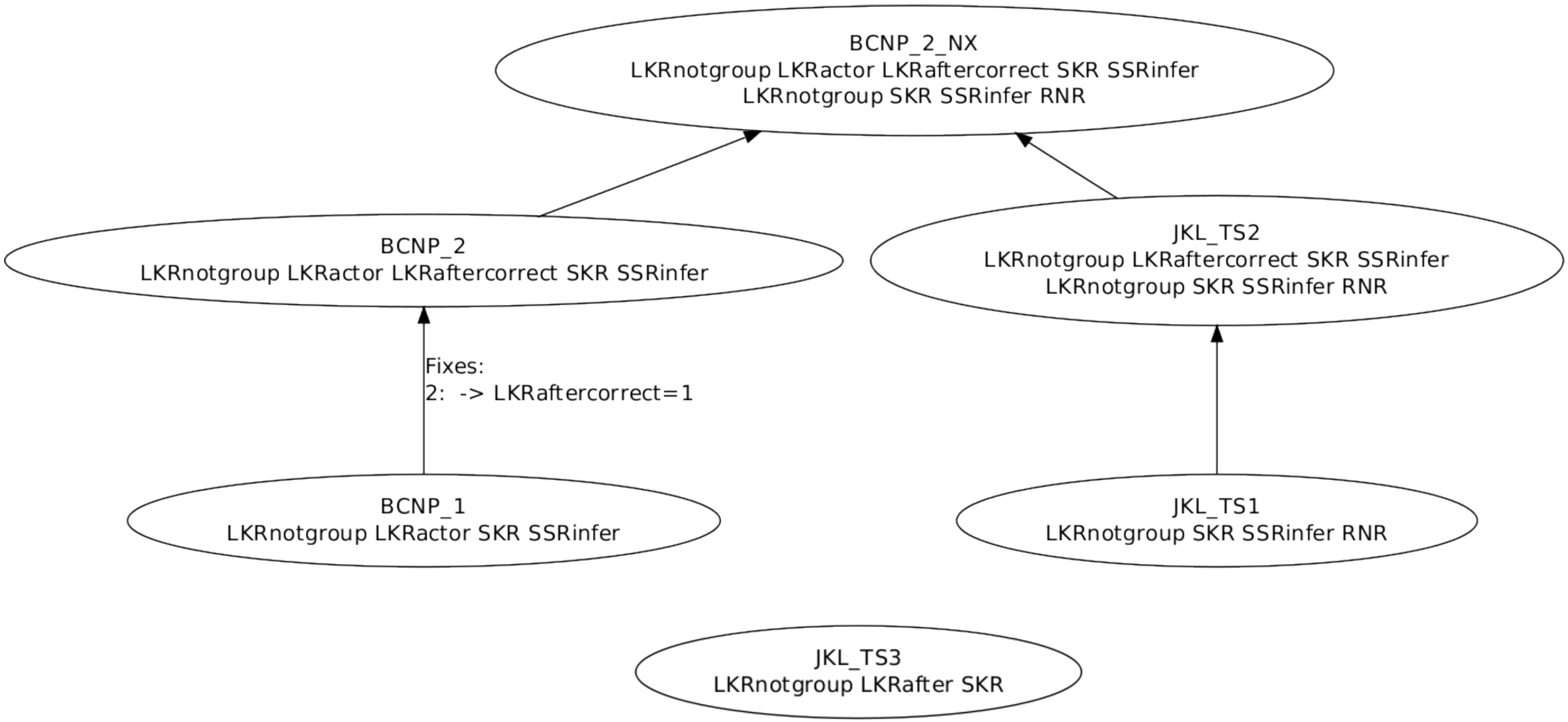
Applications of the tool

- Found novel attack on (H)MQV using state-reveal
- MQV in the NIST standard has less features than the original
  - Adding names can't be wrong, can it?
- Disproved several claims in the literature
  - Extended CK stronger than CK?
  - Extending a protocol with a key confirmation step additionally gives you property X?
  - No 2-message protocol can satisfy Perfect Forward Secrecy?

Using the tool

- Analyse a protocol in all 96 models
  - Precise characterization of the weaknesses of the protocol
- Support protocol developers
  - Explore alternative variants quickly
  - Don't waste time trying to prove a property that doesn't hold
  - Prove the strongest property that holds

Protocol security hierarchy



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Current limitations

- Abstraction in general
  - Attack found (good!)
  - No attack found in formal model

- Some operations difficult to capture in model
  - Commutativity difficult ($g^{ab} = g^{ba}$)
  - No shared variables between threads
  - ...

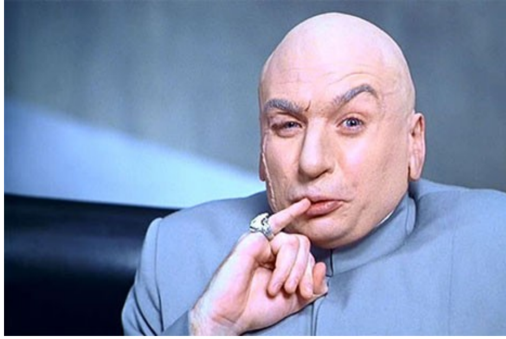
Observations along the way

- Model relation claims
  - Easy way to generate counterexamples
- Partnering (and key types)
  - Many bugs in proofs in the literature
- What is in the local state?
  - Turing machine abstraction versus implementation with TPM
- Atomicity in reactive system models
  - Is it possible to compromise between a receive and a send?

Future work

- Incorporate our adversary models into a concrete (computational) AKE model
  - think about proof modularity with respect to capabilities
- Really establish negative results
  - „Clearly no protocol can be secure for such an adversary“
- Consider other combinations of „pure“ security properties and adversary models
- Consider other adversary rules
  - Time-based compromise notions?
  - Active modification of randomness, state insertion, changing clocks, ...

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Conclusions

- Formal model: modular, generic
  - Applications beyond key exchange
    - Generalizes existing notions
  - Bridges another gap between crypto models and formal models
  - Separates pure security properties from adversary model
    - Paves the way for more detailed analysis of other properties
  - Tool support
    - First tool support for advanced security notions (weak PFS, KCI,...) for analysists as well as protocol developers
  - Older version at http://eprint.iacr.org/2009/079, mail me for the current one cas.cremers@inf.ethz.ch