ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Protocol Security in the Presence of Compromising Adversaries

Cas Cremers

Joint work with David Basin

Overview

- **Gap** between the capabilities of **adversaries in formal analysis** and **adversaries in cryptographic security notions (AKE)**

- Overview
  - Problem context & motivation
  - A formal model of compromising adversaries
  - Applications & case studies

Basis: Dolev Yao adversary model





- Models an active intruder with full network control and perfect recall
- Idealized black-box cryptography

$$\frac{t \in M}{M \vdash t} \qquad \frac{M \vdash t_1 \quad M \vdash t_2}{M \vdash (t_1, t_2)} \qquad \frac{M \vdash (t_1, t_2)}{M \vdash t_1} \qquad \frac{M \vdash (t_1, t_2)}{M \vdash t_2}$$

$$\frac{M \vdash t}{M \vdash hash(t)} \qquad \frac{M \vdash t_1 \quad M \vdash t_2}{M \vdash \{t_1\}_{t_2}} \qquad \frac{M \vdash \{t_1\}_{t_2} \quad M \vdash t_2^{-1}}{M \vdash t_1}$$

- **Successful**: interesting theory and powerful tools

Game-based security notions, e.g., key exchange

- Bit strings, probabilistic reasoning
- Reduction to known (or assumed) hard problem
- **Manual proofs**
  - Notable exception: Blanchet's CryptoVerif

- **Adversary model**
  - Active attacker
  - Dynamic compromise of long-term keys,
  - Compromise of session keys
  - Compromise of session-state, randomness,...

- **Successful**: establishing strong guarantees for real-world protocols

Why study corruption?

- **Security is relative to powers of an adversary**
- Real adversaries might...
  - Break into machine and extract disk drive
  - Read out memory
  - Cryptanalyze keys or attack side channels
  - And all of this could happen at any time!

- Flip side: rings of protection in hardware/software
  - TPMs, HSMs, smart cards and tokens vs. main memory, etc.

Formal foundations? Verification methods and tools?

Terms, roles, and protocols

- **Terms**: operators for constructing cryptographic messages

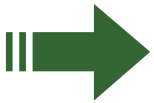
$$Term ::= Agent \mid Fresh \mid Var \mid (Term, Term) \mid \{Term\}_{Term} \mid \ldots$$

- **Roles:** sequences of **agent events**

$$AgentEvent ::= \mathrm{create}(Role, Agent) \mid \mathrm{send}(Agent, Agent, Term) \mid \mathrm{recv}(Agent, Agent, Term)$$
$$\mid \mathrm{generate}(\mathcal{P}(Fresh)) \mid \mathrm{state}(\mathcal{P}(Term)) \mid \mathrm{sessionkeys}(\mathcal{P}(Term))$$

- **Example**

$$I \to R: \quad \{I, K\}_{K_{IR}}$$
$$R \to I: \quad \{R, M\}_{K}$$



$$P(I) = [\ \mathrm{generate}(\{K\});\ \mathrm{send}(I, R, \{I, K\}_{k(I,R)});\ \mathrm{recv}(R, I, \{R, y\}_K)\ ]$$

$$P(R) = [\ \mathrm{recv}(I, R, \{I, x\}_{k(I,R)});\ \mathrm{sessionkeys}(\{x\});\ \mathrm{generate}(\{M\});\ \mathrm{send}(R, I, \{R, M\}_x)\ ]$$

Threads

- A **thread** is a role instance (local session)
  - No limit to number of threads
  - Each thread assigned a unique identifier from the set TID.
  - We **instantiate names** and **syntactically bind** fresh **values** and **variables** to their owning thread, e.g. K#1, y#1

$$P(I) = [\ \ \mathsf{generate}(\{K\});\ \mathsf{send}(I, R, \{I, K\}_{k(I,R)});\ \mathsf{recv}(R, I, \{R, y\}_K)\ \ ]$$

Thread 1
$I \mapsto A$
$R \mapsto B$

$$[\ \ \mathsf{generate}(\{K\sharp 1\});\ \mathsf{send}(A, B, \{A, K\sharp 1\}_{k(A,B)});\ \mathsf{recv}(B, A, \{B, y\sharp 1\}_{K\sharp 1})\ \ ]$$

- For currently active threads, we store the remaining sequence of steps in a thread pool $th : TID \nrightarrow AgentEvent^*$

Core symbolic model
(slightly simplified)

$$A \to B \; : \; n$$

- **State** (tr, IK, th)
  - *tr*: trace of events that have occurred
  - *IK*: "intruder knowledge" of adversary, initially IK_0
  - *th*: thread pool, mapping thread identifiers to remaining steps
- **Transition system** modeling agents' threads and **(outside) adversary**

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

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

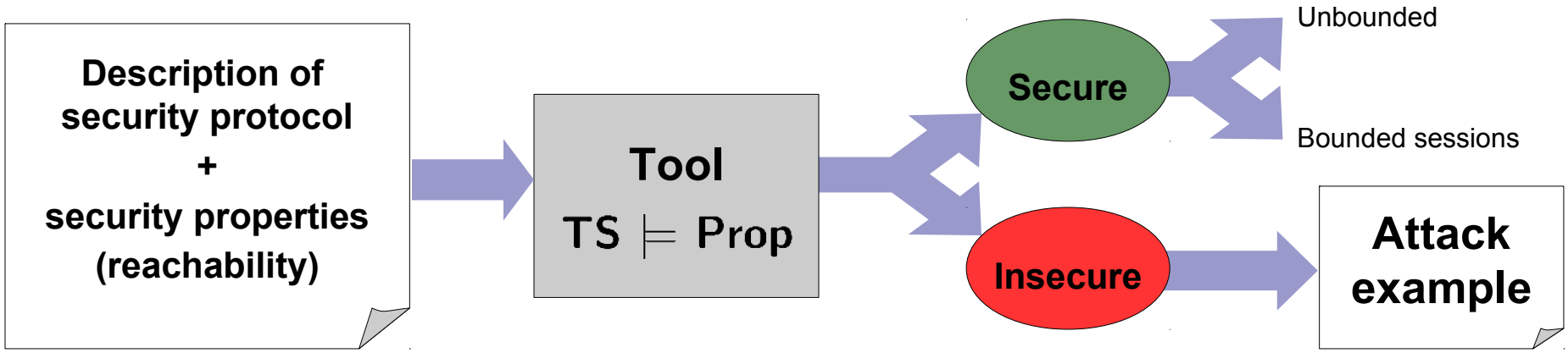
Example of reachable state:

$$\big(\underbrace{\langle (1, \mathsf{send}(A, B, n \sharp 1)) \rangle}_{tr}, \underbrace{IK_0 \cup \{n \sharp 1\}}_{IK}, \underbrace{\{1 \mapsto \langle\rangle, 2 \mapsto \langle \mathsf{recv}(A, B, X \sharp 2) \rangle\}}_{th}\big)$$

Reasoning about protocol semantics (TS)

- General complexity
  - **Reachability** properties are **undecidable**, e.g. secrecy
    (Durgin, Lincoln, Mitchell, Scedrov 1999)
  - **NP-hard,** even when number of sessions is bounded
    (Rusinowitch, Turuani, 1999)

- **Scyther tool** often successful in protocol analysis



Co-evolution of adversary models and protocols

BKE

$$
\begin{aligned}
A \to B: &\quad A, \{B, na\}_{pk(B)} \\
B \to A: &\quad \{B, H(na), nb, K\}_{pk(A)} \\
A \to B: &\quad \{H(nb)\}_K \\
\text{sessionkey:} &\quad K
\end{aligned}
$$

Signed DH

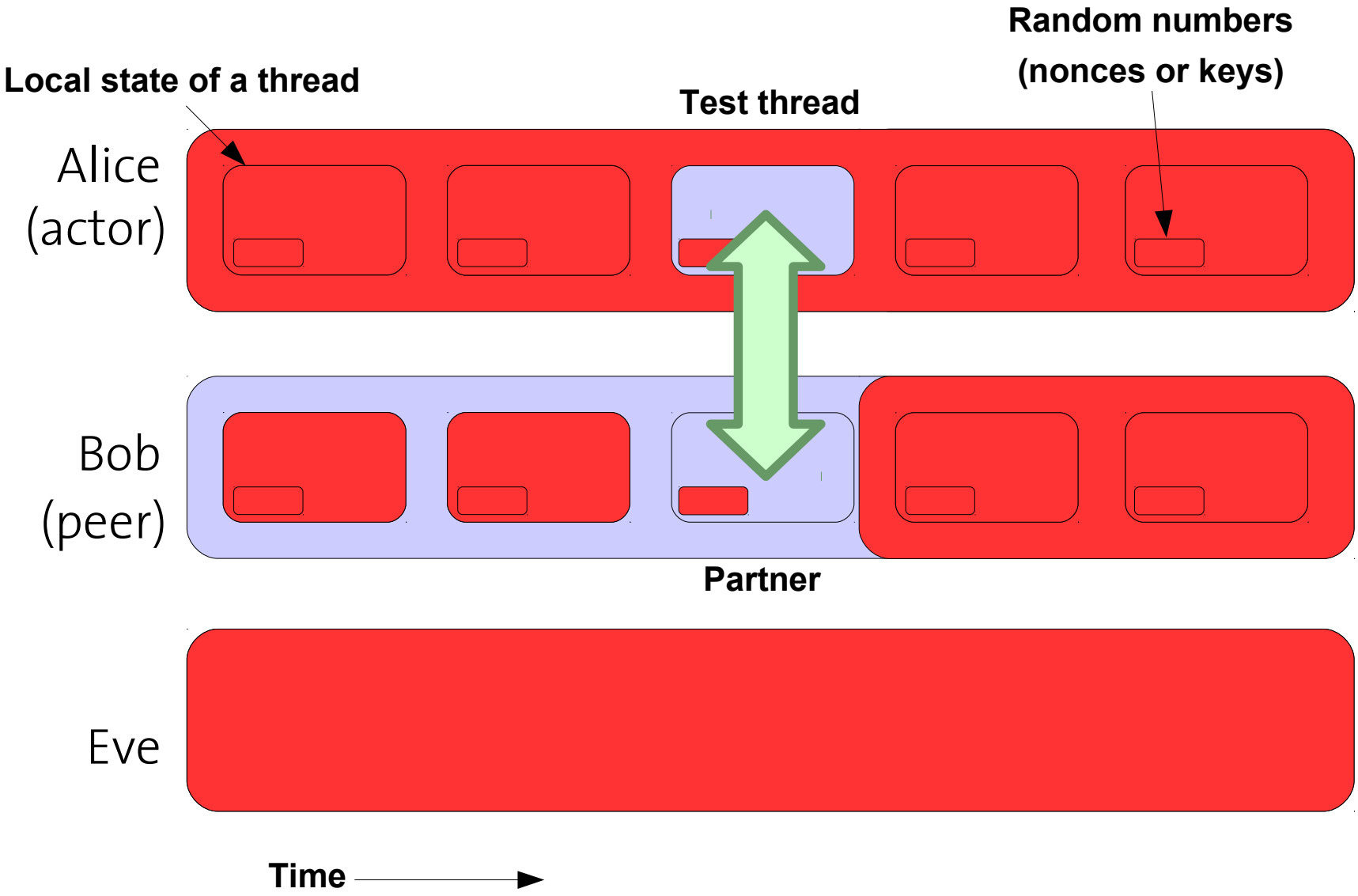
$$
\begin{aligned}
A \to B: &\quad \{B, g^{na}\}_{sk(A)} \\
B \to A: &\quad \{A, g^{nb}\}_{sk(B)} \\
\\
\text{sessionkey } A: &\quad (g^{nb})^{na} \\
\text{sessionkey } B: &\quad (g^{na})^{nb}
\end{aligned}
$$

HMQV (simplified)

$$
\begin{aligned}
A \to B: &\quad g^{na} \\
B \to A: &\quad g^{nb} \\
\\
\text{sessionkey } A: &\quad H(((g^{nb})(pk(B))^{e})^{na+d(sk(A))}) \\
\text{sessionkey } B: &\quad H(((g^{na})(pk(A))^{d})^{nb+e(sk(B))})
\end{aligned}
$$

These key exchange protocols are all "correct" in symbolic models. Finer distinctions possible using cryptographic models.

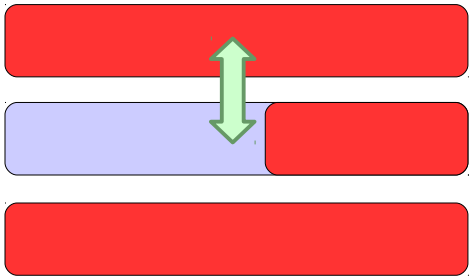
How much information can be compromised?



Dimensions of compromise

- **When:** before, during, or after test session
- **Whose data:** actor, peers, or others
- **Which data:** reveal long-term keys, session keys, state (of thread), or randomness

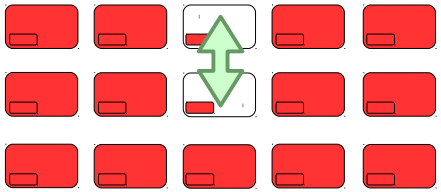
First distinction: *long-term* versus *short-term* data



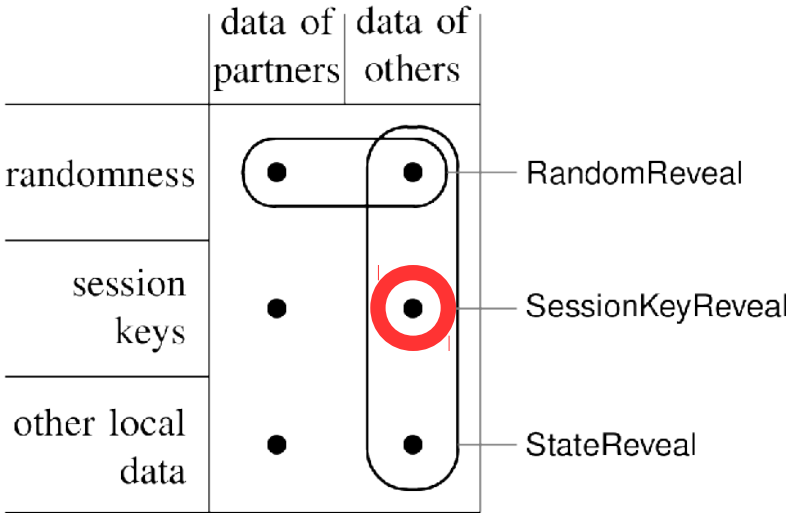
Reveal long-term data: *whose, when*

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

$$\frac{th(Test) = \langle\rangle}{(tr, IK, th) \longrightarrow (tr^\frown\langle(tid_{\mathcal{A}}, \mathsf{LKR}(a))\rangle, IK \cup LongTermKeys(a), th)}[\mathsf{LKR}_{\mathsf{after}}]$$



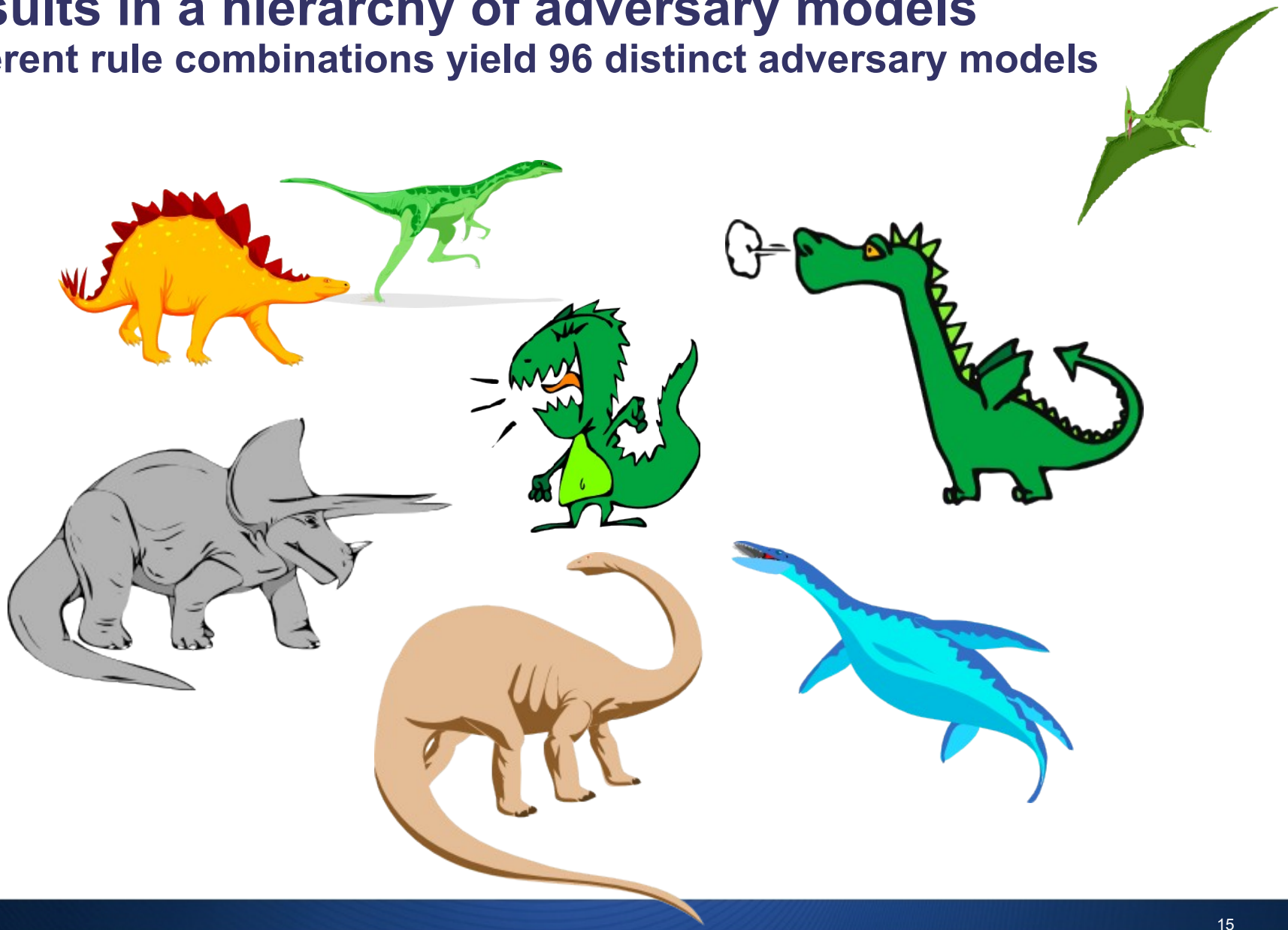
Reveal short-term data: *whose, which*



$$\frac{tid \neq Test \qquad tid \notin Partner(tr, Test)}{(tr, IK, th) \longrightarrow (tr^{\frown}\langle(tid_{\mathcal{A}}, \mathsf{SKR}(tid))\rangle, IK \cup union((tr \downarrow tid) \restriction \mathsf{sessionkeys}), th)}\,[\mathsf{SKR}]$$

Results in a hierarchy of adversary models

Different rule combinations yield 96 distinct adversary models

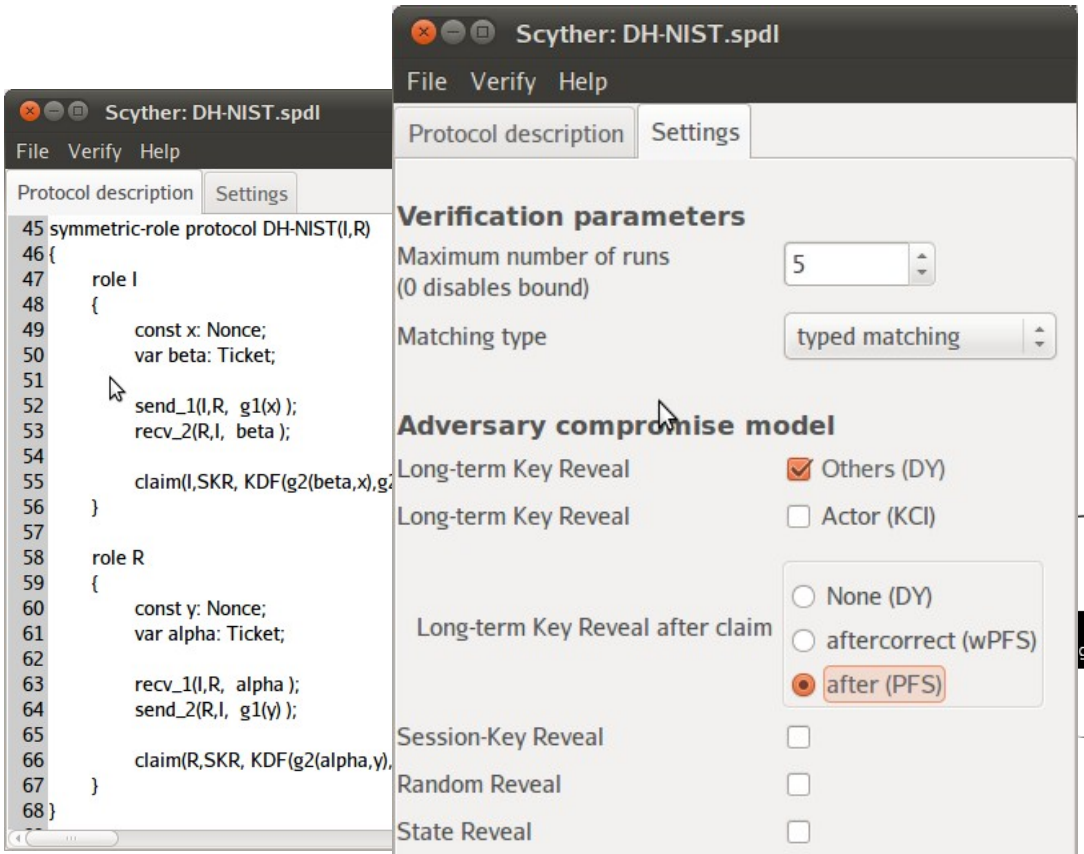


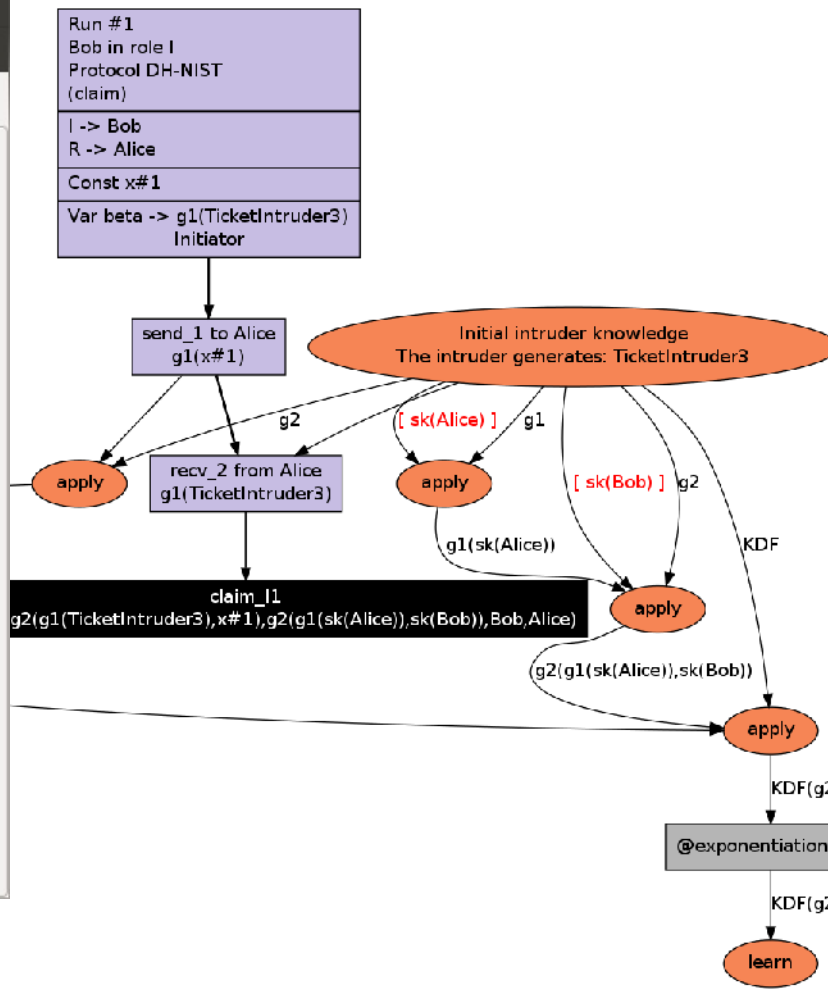
Recasting existing models

	Long-term data				Short-term data			
	Owner		Timing		Type			
Name	others	actor	after	aftercorrect	SessionKey	State	Random	Origin of model
Adv_{EXT}								external Dolev-Yao
Adv_{INT}	✓							Dolev-Yao
Adv_{CA}		✓						Key Compromise Impersonation
Adv_{AFC}				✓				Weak Perfect Forward Secrecy
Adv_{AF}			✓	✓				Perfect Forward Secrecy
Adv_{BR}	✓				✓			BR93 , BR95
Adv_{CKw}	✓	✓		✓	✓	✓		CK2001-wPFS
Adv_{CK}	✓		✓	✓	✓	✓		CK2001
$Adv_{eCK\text{-}1}$	✓				✓		✓	eCK
$Adv_{eCK\text{-}2}$	✓	✓		✓	✓			

... plus dozens of new models

Tool support: extension of the *Scyther tool*





[Id 2] Protocol DH-NIST, role I, claim type SKR, cost 84

Applications

Many new (∗) and rediscovered (√) attacks

	EXT	INT	CA	AFC	AF	BR	CKw	CK	eCK-1	eCK-2
DH-ISO									✓	
DH-ISO-C								✓	✓	
DHKE-1							⋆		✓	✓
HMQV-C							✓	✓		
HMQV					✓		⋆	✓		
NAXOS					✓		✓	✓		
KEA+				⋆	✓		✓	✓	✓	✓
NSL			⋆	✓	✓		✓	✓	✓	✓
BKE			⋆	✓	✓		✓	✓	✓	✓
Yahalom-Paulson			⋆	✓	✓	⋆	✓	✓	✓	✓
NS		✓	⋆	✓	✓	✓	✓	✓	✓	✓

- Nontrivial analysis
  - Previously by hand: 1 attack = 1 publication
  - Now tool-based: automatic, within seconds
- Can determine strength of a protocol (WRT 96 different models), establishing/disproving relationships between protocols

Unexpected side effects in applied cryptography

- Formalizing models reveals **complex relations between AKE security notions**
  - "Examining Indistinguishability-Based Security Models for Key Exchange Protocols...", ASIACCS 2011
- Automatically generate **counterexamples to folklore**
  - "Session-state Reveal is stronger than Ephemeral Key Reveal...", ACNS'09
- Suggests **new directions**
  - "One-round Strongly Secure Key Exchange with Perfect Forward Secrecy and Deniability", with M. Feltz, manuscript.

Conclusions
Compromising Adversaries

- **Bridges significant gap** between crypto and formal models
- For users of formal methods
  - **Stronger adversary model** than standard DY
  - **Tool-supported** formal methodology with **new applications**
  - First **formal definitions of KCI, wPFS, etc.**
- For AKE protocol designers and provers
  - Enable **fast evaluation/comparison of protocols**
  - Provides hints for the **maximal provable computational security**
- Tool freely available (search for “Scyther tool”)

cas.cremers@inf.ethz.ch