Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Informatik
Computer Science

Verification of Human Interaction Security Protocols (HISP) – An Attempt

E-Voting Seminar

24 May 2012

Michael Schläpfer

Overview

1. Introduction

2. Symbolic Protocol Analysis Basics

3. Evolution of the Symbolic Attacker

4. Human Interaction Security Protocols (HISP)

5. Symbolically Modeling HISP

6. Verification of Security Properties in HISP

7. Conclusion

Security protocols

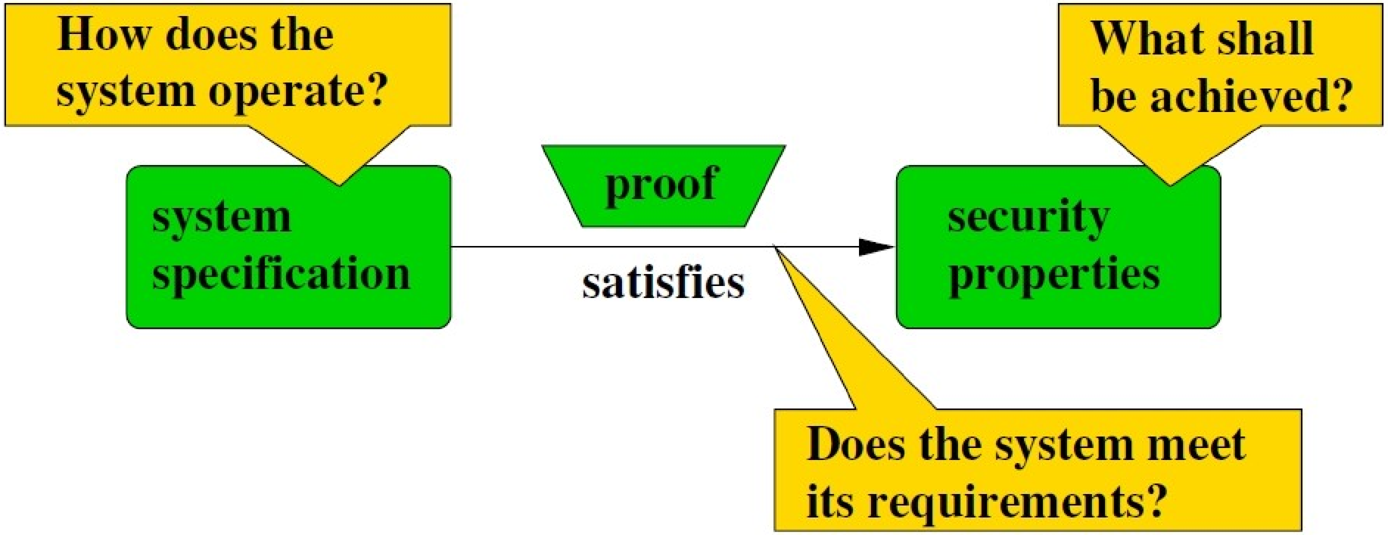
A **protocol** consists of a set of rules (conventions) that determine the exchange of messages between two or more principals. In short, a **distributed algorithm** with emphasis on communication.

Security (or **cryptographic**) protocols use cryptographic mechanisms to achieve security objectives.

Some common security objectives:

- Entity or message authentication
- Key establishment
- Integrity
- Fair exchange
- Non-repudiation
- ...

Formal security models


How does the system operate?

system specification

proof

satisfies

security properties

What shall be achieved?

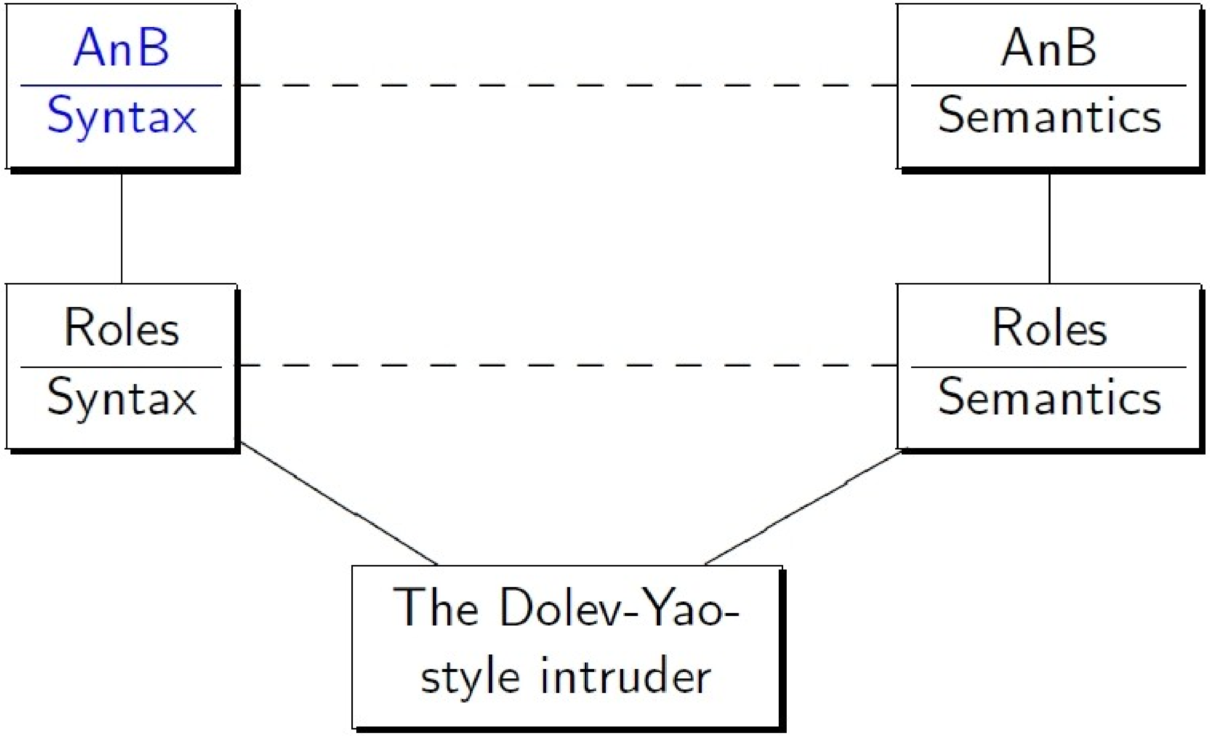
Does the system meet its requirements?

- Formal specification with formal languages
- Semantics of languages allow for verification and validation with mathematical methods

Overview

1. Introduction
2. Symbolic Protocol Analysis Basics
3. Evolution of the Symbolic Attacker
4. Human Interaction Security Protocols (HISP)
5. Symbolically Modeling HISP
6. Verification of Security Properties in HISP
7. Conclusion

Two formal languages



Message notation

Roles: A, B or *Alice*, *Bob*

Agents: a, b, i

Symmetric Keys: K, K_{AB}, . . . ; $\mathsf{sk}(A, S)$

Symmetric Encryption: $\{\!|M|\!\}_K$

Public Keys: K, $\mathsf{pk}(A)$

Private Keys: $\mathsf{inv}(K)$, $\mathsf{inv}(\mathsf{pk}(A))$

Asymmetric Encryption: $\{M\}_K$

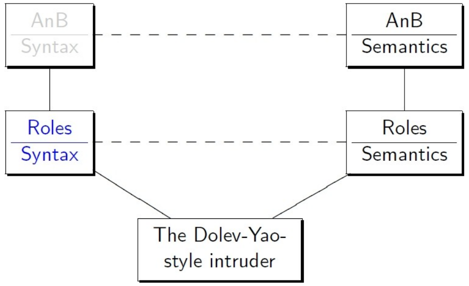
Signing: $\{M\}_{\mathsf{inv}(K)}$

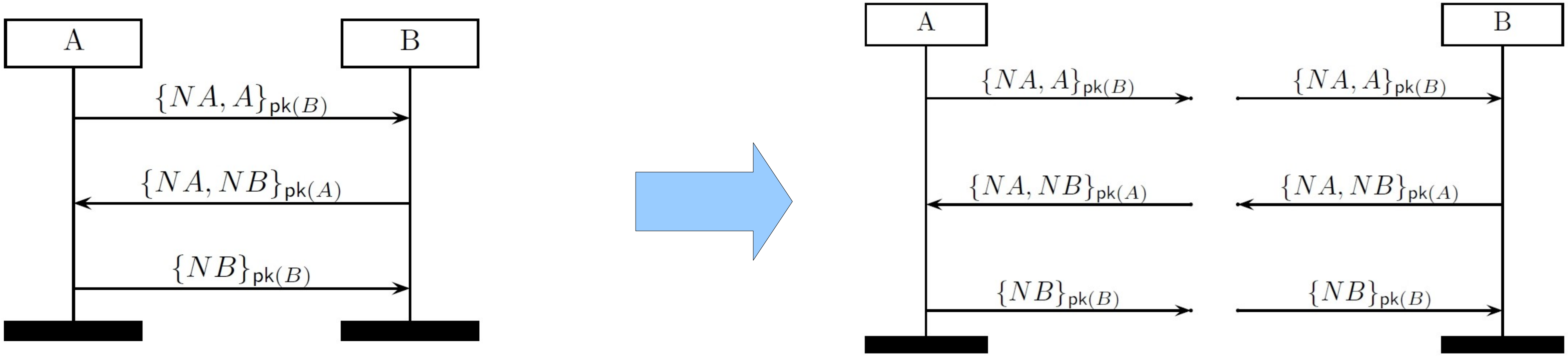
Nonces: NA, $N1$ fresh data items used for challenge/response.
N.B.: sometimes subscripts are used, e.g. N_A, but it does not mean that principals can find out that N_A was generated by A.

Timestamps: T. Denote time, e.g. used for key expiration.

Message concatenation: M_1, M_2, M_3

Role scripts for A and B

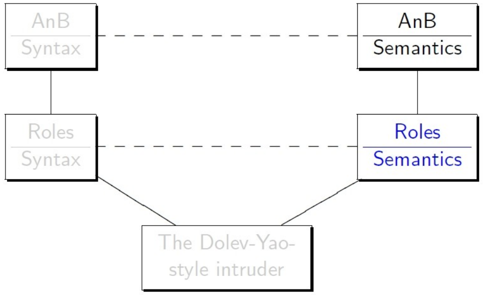




Textual:

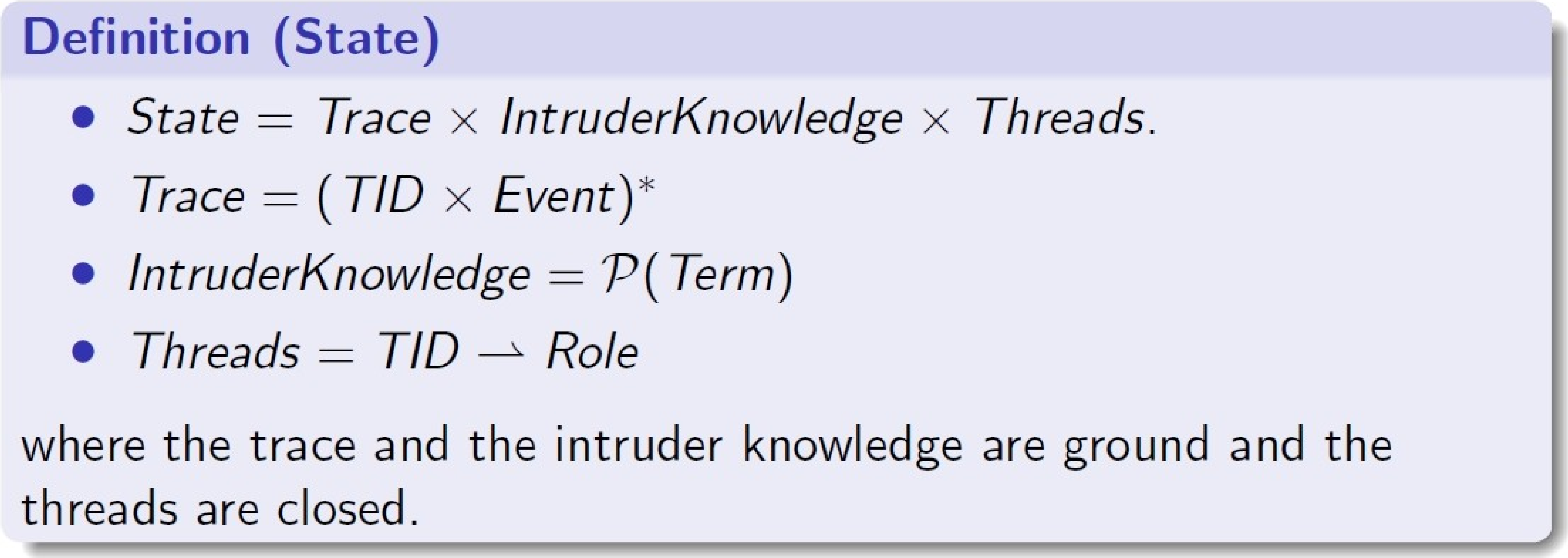
$$\mathrm{NSPK}(A) := \mathrm{snd}(\{NA, A\}_{pk(B)}) \cdot \mathrm{rcv}(\{NA, NB\}_{pk(A)}) \cdot \mathrm{snd}(\{NB\}_{pk(B)})$$

Operational semantics



- Defined by a transition system

$$TS(P, IK_0, th_0) = (State, \rightarrow, ([], IK_0, th_0))$$

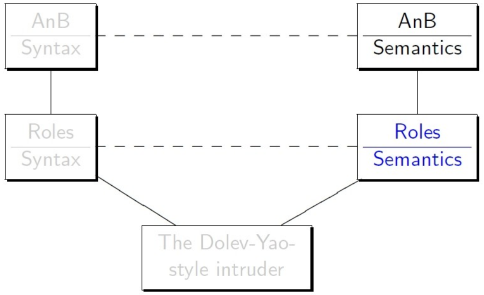


Definition (State)

- $\mathit{State} = \mathit{Trace} \times \mathit{IntruderKnowledge} \times \mathit{Threads}.$
- $\mathit{Trace} = (\mathit{TID} \times \mathit{Event})^*$
- $\mathit{IntruderKnowledge} = \mathcal{P}(\mathit{Term})$
- $\mathit{Threads} = \mathit{TID} \rightharpoonup \mathit{Role}$

where the trace and the intruder knowledge are ground and the threads are closed.

Operational semantics



$$TS(P, IK_0, th_0) = (State, \rightarrow, ([], IK_0, th_0))$$

- Transition relation defined by a set of deduction rules
- Signals sig will be explained later

Rules

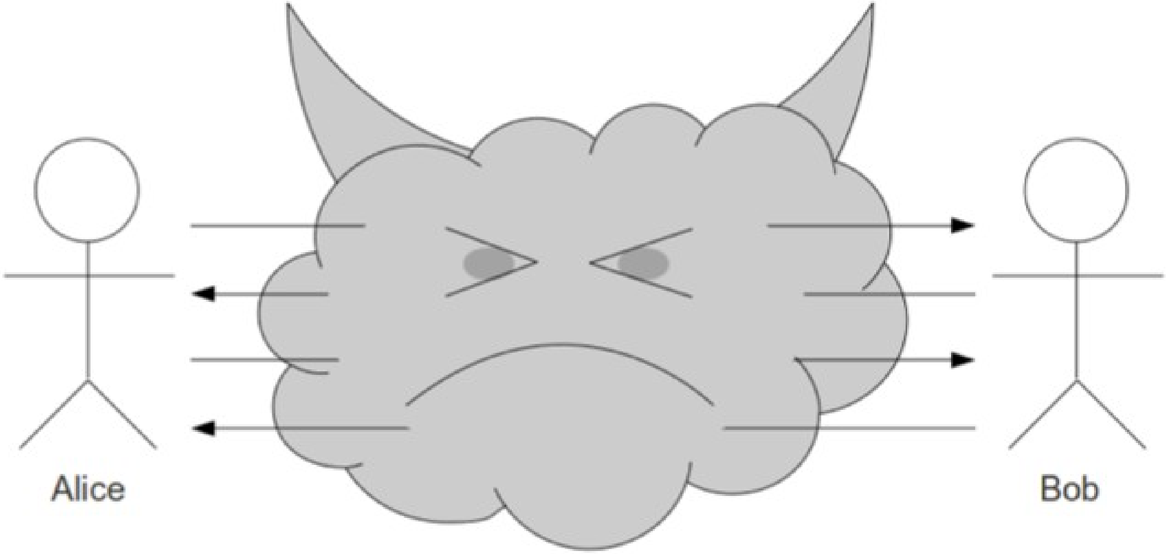
$$\frac{th(tid) = \mathsf{snd}(t) \cdot tl}{(tr, IK, th) \rightarrow (tr \cdot (tid, \mathsf{snd}(t)), IK \cup \{t\}, th[tid \mapsto tl])}\ \mathsf{snd}$$

$$\frac{th(tid) = \mathsf{rcv}(t) \cdot tl \quad dom(\sigma) = var(t) \quad t\sigma \in \mathcal{DY}(IK)}{(tr, IK, th) \rightarrow (tr \cdot (tid, \mathsf{rcv}(t\sigma)), IK, th[tid \mapsto tl\sigma])}\ \mathsf{rcv}$$

$$\frac{th(tid) = \mathsf{sig}(sig, t) \cdot tl}{(tr, IK, th) \rightarrow (tr \cdot (tid, \mathsf{sig}(sig, t)), IK, th[tid \mapsto tl])}\ \mathsf{sig}$$

Modeling the Attacker

Communication in an dangerous world.

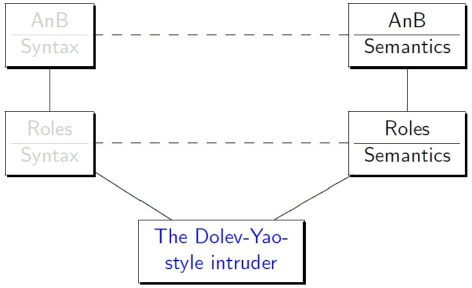


- **On the Security of Public Key Protocols (IEEE Trans. Inf. Th. 1983):**
- Danny Dolev
- Andrew C. Yao

The Dolev-Yao Intruder:

- Controls the network (read, intercept, send)
- Is a legitimate user
- Can apply every publicly available information or function
- Can apply his private information and functions
- Cannot break cryptography

Modeling the Attacker



Definition

Given a set of terms M we define $\mathcal{DY}(M)$ as the least closure of M under the following rules:

$$\frac{}{m \in \mathcal{DY}(M)}\ \text{Axiom } (m \in M) \qquad \frac{s \in \mathcal{DY}(M)}{t \in \mathcal{DY}(M)}\ \text{Algebra } (s \approx t)$$

$$\frac{t_1 \in \mathcal{DY}(M) \quad \ldots \quad t_n \in \mathcal{DY}(M)}{f(t_1, \ldots, t_n) \in \mathcal{DY}(M)}\ \text{Composition } (f \in \Sigma_p)$$

$$\frac{\langle m_1, m_2 \rangle \in \mathcal{DY}(M)}{m_i \in \mathcal{DY}(M)}\ \text{Proj}_i \qquad \frac{\{|m|\}_k \in \mathcal{DY}(M) \quad k \in \mathcal{DY}(M)}{m \in \mathcal{DY}(M)}\ \text{DecSym}$$

$$\frac{\{m\}_k \in \mathcal{DY}(M) \quad \text{inv}(k) \in \mathcal{DY}(M)}{m \in \mathcal{DY}(M)}\ \text{DecAsym} \qquad \frac{\{m\}_{\text{inv}(k)} \in \mathcal{DY}(M)}{m \in \mathcal{DY}(M)}\ \text{OpenSig}$$

A simple example

Example

$$M = \{\ x, \{\!|b, \exp(g,y)|\!\}_k, k, m\ \}$$

$$\{\!|m|\!\}_{\exp(\exp(g,x),y)} \overset{?}{\in} \mathcal{DY}(M)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\!|b, \exp(g,y)|\!\}_k \in \mathcal{DY}(M)} \quad \dfrac{}{k \in \mathcal{DY}(M)}}{\langle b, \exp(g,y)\rangle \in \mathcal{DY}(M)}}{\exp(g,y) \in \mathcal{DY}(M)} \quad \dfrac{}{x \in \mathcal{DY}(M)}}{\exp(\exp(g,y),x) \in \mathcal{DY}(M)}}{\exp(\exp(g,x),y) \in \mathcal{DY}(M)} \quad \dfrac{}{m \in \mathcal{DY}(M)}}{\{\!|m|\!\}_{\exp(\exp(g,x),y)} \in \mathcal{DY}(M)}$$

Protocol properties

Properties:

- Semantics of a security protocol P is a set of traces $||P|| = traces(P)$
- Security goal / property ϕ also denotes a set of traces $||\phi||$

Correctness:

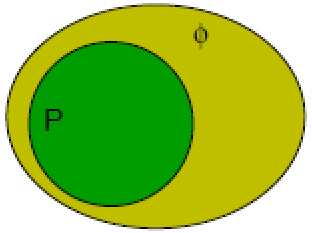
- Protocol P satisfies property ϕ, written $P \models \phi$, iff

$$||P|| \subseteq ||\phi||$$

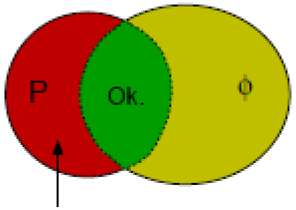
- Attack traces are those in

$$||P|| - ||\phi||$$

- Every correctness statement is either true or false



Ok, no attacks.



Attacks.

Formalizing security properties

Direct formulation:

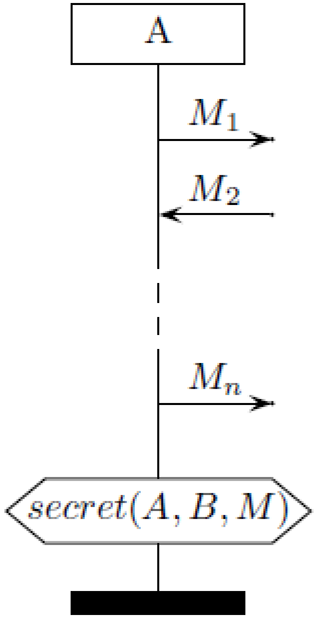
- Formulate property directly in terms of send and receive events occurring in protocol traces, i.e., as a set of (or predicate on) traces
- Drawback: Standard properties like secrecy and authentication become highly protocol-dependent, since they need to refer to the concrete protocol messages

Protocol instrumentation

- Insert special signal events into the protocol roles
- Possible to express properties independently of protocol
- Example:

sig(*secret, A, B, M*)

claims that *M* is a secret shared by roles *A* and *B*



Formalizing secrecy

Example (NSPK Attack)

Trace	$th(0)$	$th(1)$
$(0, \mathrm{snd}(\{na_0, i\}_{\mathrm{pk}(i)}))$		
$(1, \mathrm{rcv}(\{na_0, a\}_{\mathrm{pk}(b)}))$		
$(1, \mathrm{snd}(\{na_0, nb_1\}_{\mathrm{pk}(a)}))$		
$(0, \mathrm{rcv}(\{na_0, nb_1\}_{\mathrm{pk}(a)}))$		
$(0, \mathrm{snd}(\{nb_1\}_{\mathrm{pk}(i)}))$		
$(1, \mathrm{rcv}(\{nb_1\}_{\mathrm{pk}(b)}))$		

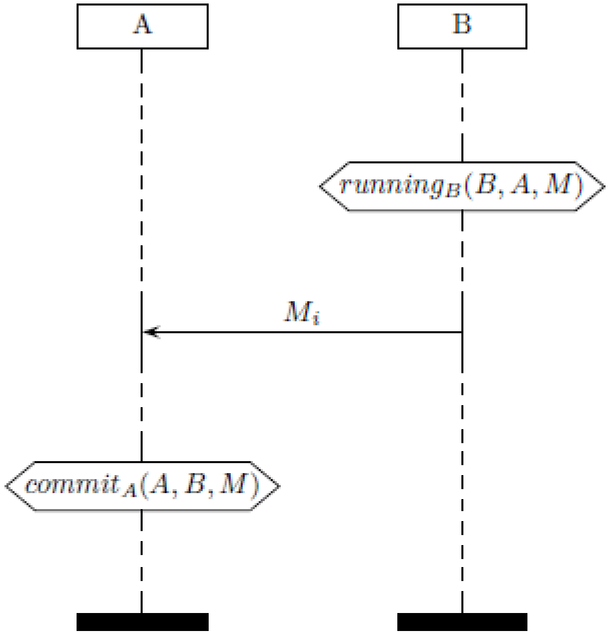
Definition (Secrecy)

The property $Secret(A, B, M)$ consists of all traces tr satisfying

$$\forall tid.\ (tid, \mathrm{sig}(secret, A, B, M)) \in set(tr)\ \wedge\ B \neq i \Rightarrow M \notin \mathcal{DY}(IK(tr))$$

$$IK(tr) = \{m | \exists tid.(tid, snd(m)) \in set(tr)\}$$

Formalizing authentication



Two new signals:

- *running*
- *commit*

Different definitions:

- Aliveness
- Weak agreement
- Non-injective agreement
- Injective agreement
- …

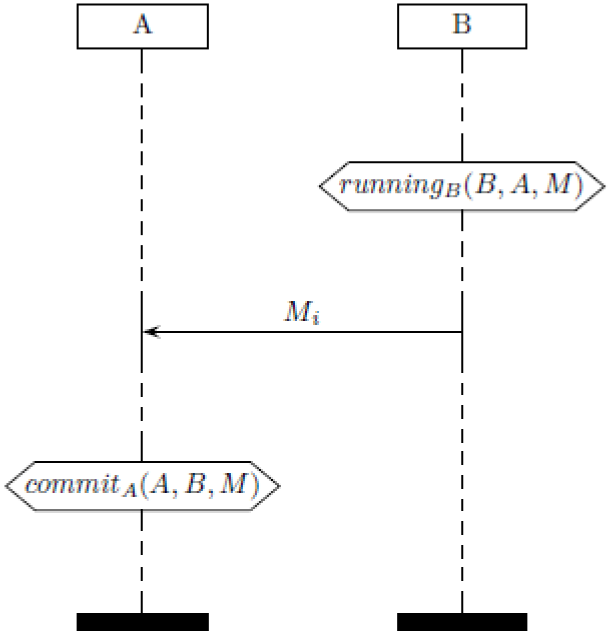
Example:

Definition (Non-injective agreement)

We define $tr \in Agreement_{NI}(A, B, M)$ for a trace tr by

$$\begin{aligned}&\forall tid.\ (tid, \mathrm{sig}(commit_A, A, B, M)) \in set(tr) \wedge B \neq i\\ &\Rightarrow \exists tid'.(tid', \mathrm{sig}(running_B, B, A, M)) \in set(tr)\end{aligned}$$

Formalizing authentication



Two new signals:

- *running*
- *commit*

Different definitions:

- Aliveness
- Weak agreement
- Non-injective agreement
- Injective agreement
- …

Example:

Definition (Injective agreement)

We define $tr \in Agreement(A, B, M)$ for a trace tr iff there is an injective function $g : TID \rightarrow TID$ such that

$$\forall tid.\ (tid, \mathrm{sig}(commit_A, A, B, M)) \in set(tr) \wedge B \neq i$$
$$\Rightarrow (g(tid), \mathrm{sig}(running_B, B, A, M)) \in set(tr)$$

Overview

1. Introduction

2. Symbolic Protocol Analysis Basics

3. Evolution of the Symbolic Attacker

4. Human Interaction Security Protocols (HISP)

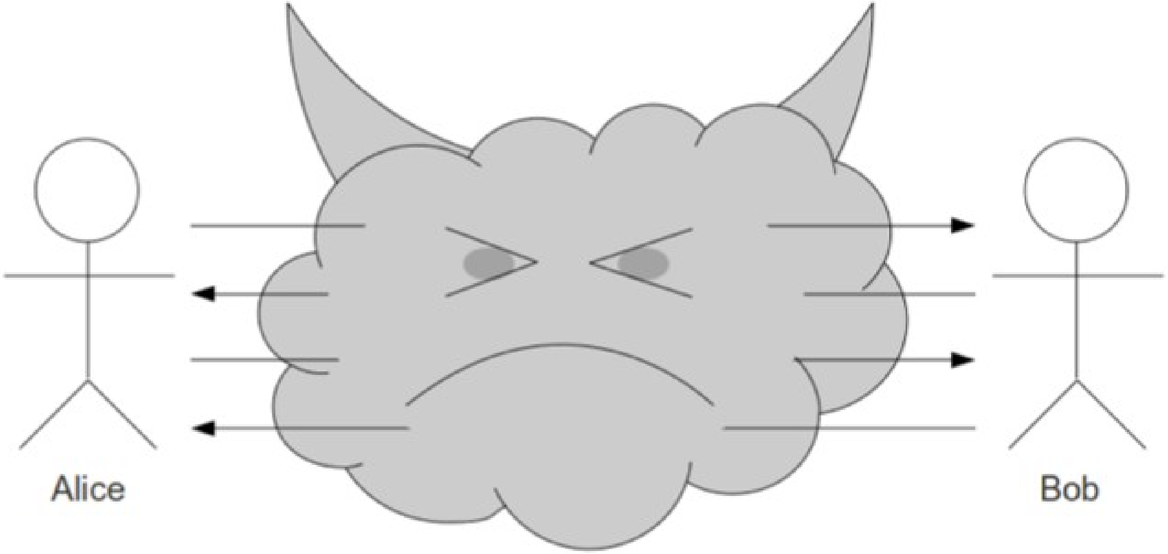
5. Symbolically Modeling HISP

6. Verification of Security Properties in HISP

7. Conclusion

Modeling the Attacker

Communication in an dangerous world.

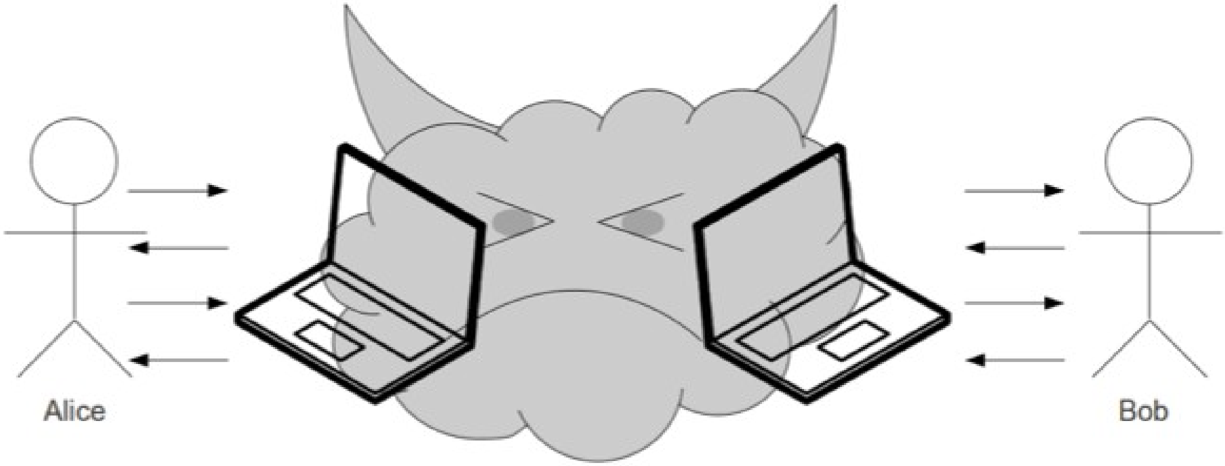


- **On the Security of Public Key Protocols (IEEE Trans. Inf. Th. 1983):**
- Danny Dolev
- Andrew C. Yao

The Dolev-Yao Intruder:

- Controls the network (read, intercept, send)
- Is a legitimate user
- Can apply every publicly available information or function
- Can apply his private information and functions
- Cannot break cryptography

Evolution of the Attacker

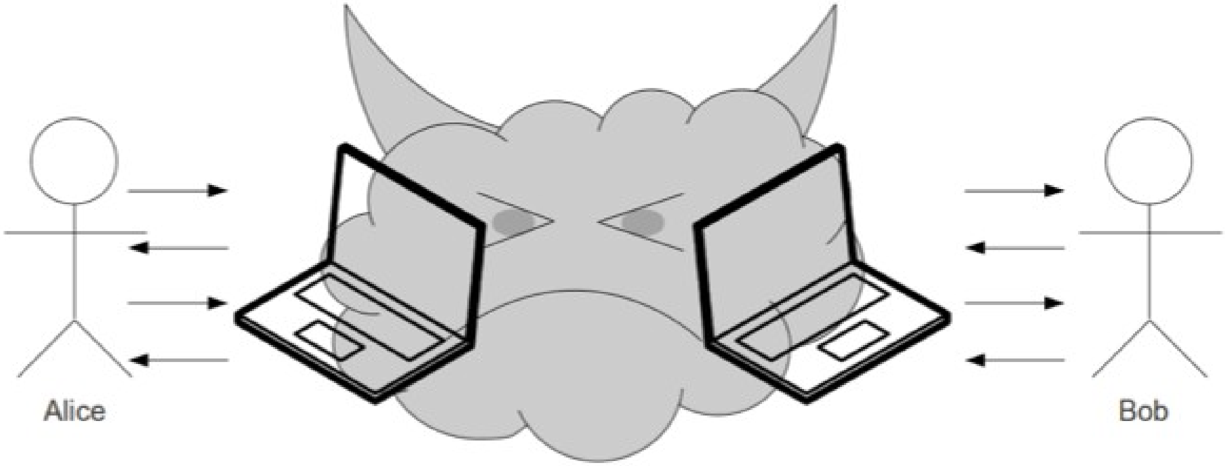


- **Modeling and Analyzing Security in the Presence of Compromising Adversaries (ESORICS 2010):**
- David A. Basin
- Cas Cremers

The extended Dolev-Yao Intruder:

- Additionally gets access to specific long-term secrets
- Allows to verify perfect forward secrecy

Evolution of the Attacker



Depending on the **application** and the resulting **threat sources** we will have to assume a very powerful attacker, capable of **controlling the entire computing platform**.

Overview

1. Introduction
2. Symbolic Protocol Analysis Basics
3. Evolution of the Symbolic Attacker
4. Human Interaction Security Protocols (HISP)
5. Symbolically Modeling HISP
6. Verification of Security Properties in HISP
7. Conclusion

Human Interaction Security Protocols

A **human interaction protocol (HIP)** consists of a set of rules (conventions) that determine the exchange of messages between two or more principals where at least one principle is human. In short, a **distributed algorithm** with emphasis on communication between humans and machines.

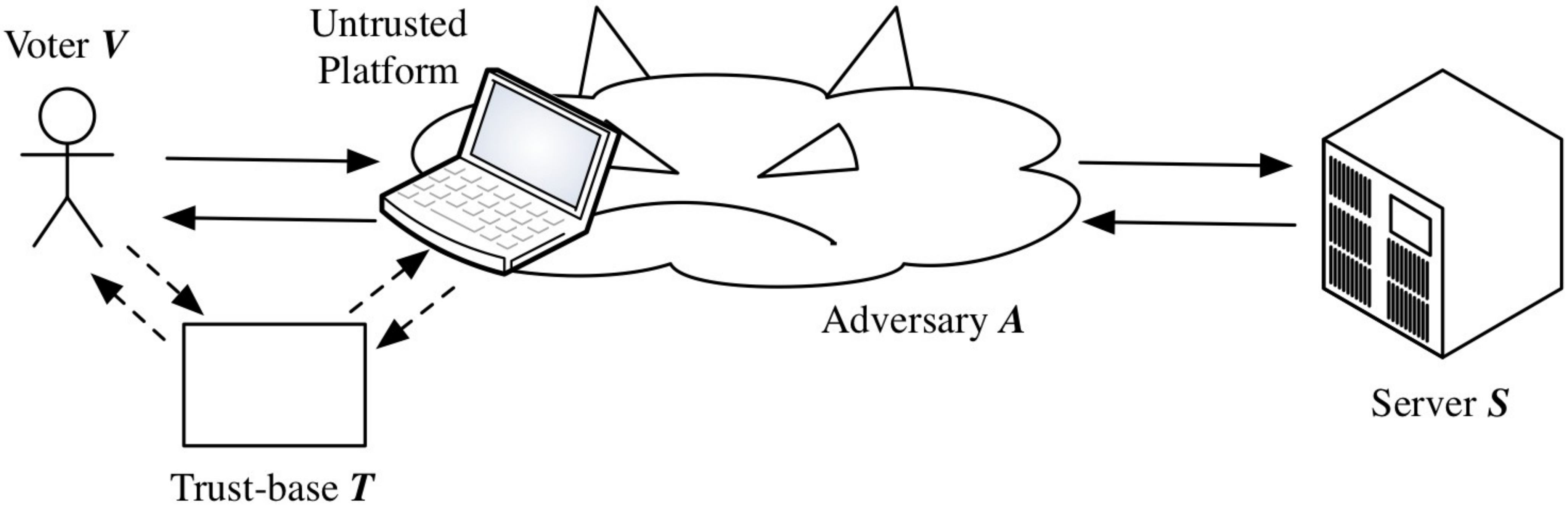
Human Interaction Security protocols (HISP) use cryptographic mechanisms to achieve security objectives between humans and machines.

Security objectives include:

- Entity or message authentication
- Integrity
- Non-repudiation
- Secrecy
- …

Humans are limited in terms of computing capabilities and therefore they need help for the computations required by cryptographic protocols!

The Simple HISP Problem



- We abstract from the user's platform
- The attacker offers the network services to the user
- Abstracting from the construction of the messages as it is done in Dolev-Yao-like models cannot cover the Secure Platform Problem in general
- Trusted functionalities modeled by a trust-base

Overview

1. Introduction
2. Symbolic Protocol Analysis Basics
3. Evolution of the Symbolic Attacker
4. Human Interaction Security Protocols (HISP)
5. Symbolically Modeling HISP
6. Verification of Security Properties in HISP
7. Conclusion

Overview

Operational Semantics States:

$State := Trace \times K_V \times K_S \times K_T \times K_I \times Threads$

$Trace := (TID \times Event)^*$

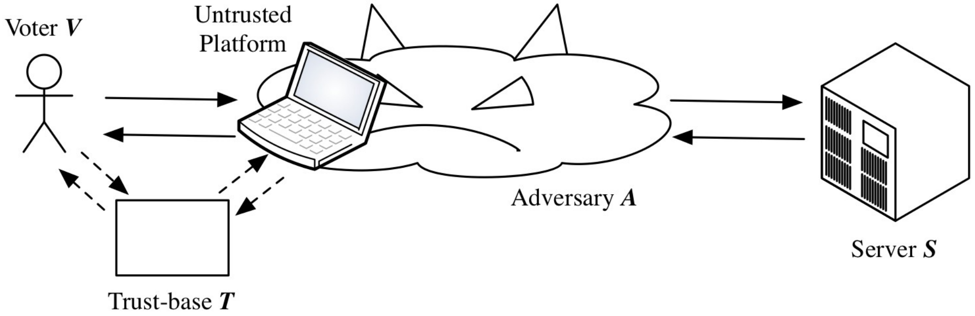
$K_V := \wp(Term)$

$K_S := \wp(Term)$

$K_T := \wp(Term)$

$K_I := \wp(Term)$

$Threads := TID \rightarrow Role$



Operational Semantics Transition Rules:

$$\frac{th(tid)=snd(A,B,M)\cdot tl \qquad A=v\wedge B=t\vee A=t\wedge B=v}{(tr,K_V,K_S,K_T,K_I,th)\rightarrow(tr\cdot(tid,snd(A,B,M))\,K_V,K_S,K_T,K_I,th[tid\rightarrow tl])}\; snd_{sec}$$

$$\frac{th(tid)=snd(A,B,M)\cdot tl \qquad \neg(A=v\wedge B=t\vee A=t\wedge B=v)}{(tr,K_V,K_S,K_T,K_I,th)\rightarrow(tr\cdot(tid,snd(A,B,M))\,K_V,K_S,K_T,K_I\cup\{M\},th[tid\rightarrow tl])}\; snd_{insec}$$

$$\frac{th(tid)=rcv(A,B,M)\cdot tl \qquad A=v\wedge B=t\vee A=t\wedge B=v \qquad (tid',snd(A,B,M')) \qquad M\sigma=M'}{(tr,K_V,K_S,K_T,K_I,th)\rightarrow(tr\cdot(tid,rcv(A,B,M\sigma))\,K_V\cup\{M\sigma\},K_S,K_T\cup\{M\sigma\},K_I,th[tid\rightarrow tl\sigma])}\; rcv_{sec}$$

$$\frac{th(tid)=rcv(A,B,M)\cdot tl \qquad \neg(A=v\wedge B=t\vee A=t\wedge B=v) \qquad dom(\sigma)=vars(M) \qquad M\sigma\in DY(K_I)}{(tr,K_V,K_S,K_T,K_I,th)\rightarrow(tr\cdot(tid,rcv(A,B,M\sigma))\,K_V\cup\{M\sigma|B=v\},K_S\cup\{M\sigma|B=s\},K_T\cup\{M\sigma|B=t\},K_I,th[tid\rightarrow tl\sigma])}\; rcv_{insec}$$

$$\frac{th(tid)=sig(sig,M)\cdot tl}{(tr,K_V,K_S,K_T,K_I,th)\rightarrow(tr\cdot(tid,sig(sig,M)),K_V,K_S,K_T,K_I,th[tid\rightarrow tl])}\; sig$$

Overview

1. Introduction
2. Symbolic Protocol Analysis Basics
3. Evolution of the Symbolic Attacker
4. Human Interaction Security Protocols (HISP)
5. Symbolically Modeling HISP
6. Verification of Security Properties in HISP
7. Conclusion

Formalizing secrecy

$$\frac{th(tid)=snd(A,B,M)\cdot tl \qquad A=v \land B=t \lor A=t \land B=v}{(tr, K_V, K_S, K_T, K_I, th) \rightarrow (tr \cdot (tid, snd(A,B,M))\, K_V, K_S, K_T, K_I, th[tid \rightarrow tl])} \; snd_{sec}$$

$$\frac{th(tid)=snd(A,B,M)\cdot tl \qquad \neg(A=v \land B=t \lor A=t \land B=v)}{(tr, K_V, K_S, K_T, K_I, th) \rightarrow (tr \cdot (tid, snd(A,B,M))\, K_V, K_S, K_T, K_I \cup \{M\}, th[tid \rightarrow tl])} \; snd_{insec}$$

$$\frac{th(tid)=rcv(A,B,M)\cdot tl \qquad A=v \land B=t \lor A=t \land B=v \qquad (tid', snd(A,B,M')) \qquad M\sigma=M'}{(tr, K_V, K_S, K_T, K_I, th) \rightarrow (tr \cdot (tid, rcv(A,B,M\sigma))\, K_V \cup \{M\sigma\}, K_S, K_T \cup \{M\sigma\}, K_I, th[tid \rightarrow tl\sigma])} \; rcv_{sec}$$

$$\frac{th(tid)=rcv(A,B,M)\cdot tl \qquad \neg(A=v \land B=t \lor A=t \land B=v) \qquad dom(\sigma)=vars(M) \qquad M\sigma \in DY(K_I)}{(tr, K_V, K_S, K_T, K_I, th) \rightarrow (tr \cdot (tid, rcv(A,B,M\sigma))\, K_V \cup \{M\sigma | B=v\}, K_S \cup \{M\sigma | B=s\}, K_T \cup \{M\sigma | B=t\}, K_I, th[tid \rightarrow tl\sigma])} \; rcv_{sec}$$

$$\frac{th(tid)=sig(sig,M)\cdot tl}{(tr, K_V, K_S, K_T, K_I, th) \rightarrow (tr \cdot (tid, sig(sig,M)), K_V, K_S, K_T, K_I, th[tid \rightarrow tl])} \; sig$$

Definition (Secrecy)

The property *Secret*(A, B, M) consists of all traces *tr* satisfying

$$\forall tid.\ (tid, sig(secret, A, B, M)) \in set(tr) \ \land \ B \neq i \Rightarrow M \notin \mathcal{DY}(IK(tr))$$

$$IK(tr) := \{m | \exists tid.(tid, snd(A,B,m)) \in set(tr) \land \neg(A=v \land B=t \lor A=t \land B=v)\}$$

And the same applies for Authenticity!

Overview

1. Introduction
2. Symbolic Protocol Analysis Basics
3. Evolution of the Symbolic Attacker
4. Human Interaction Security Protocols (HISP)
5. Symbolically Modeling HISP
6. Verification of Security Properties in HISP
7. Conclusion

Conclusion

Summary:

- Human Interaction Security Protocols are widespread
- No formal symbolic support for security verification so far
- Extension of existing approaches that are used by existing verification tools
- Foundation also for modeling the Secure Platform Problem in e-voting

Open issues and future work:

- Formalize orthogonal problem of computability for V and T (deduction rules or equational theory)
- Formalize channel restrictions and limitations between V and T
- Extend security goal definitions (e.g., e-voting related properties)
- Include in existing model checking tools
- Implement proof of concept with example protocols / attacks

Questions

?