Policy-Based Design and Verification for Mission Assurance

Shiu-Kai Chin[1], Sarah Muccio[2], Susan Older[1], and Thomas Vestal[1,2]
[1] *Syracuse University, Syracuse, New York*
[2] *Air Force Research Laboratory, Rome, New York*

Why Trust and Assurance Matter

"That is, the pilot can trust information that a target is the foe, not innocent inhabitants of a school building or hospital or embassy. . . . **This new way of war is data dependent. So we need to think in terms of trust and securing trust.**"

Michael Wynne, Former SECAF

"No operator should ever have to ask . . . 'Will my weapon work?' . . . Cyberspace warfare creates just this possibility."

General John A. Shaud, USAF

Why Trust and Assurance Matter

"That is, the pilot can trust information that a target is the foe, not innocent inhabitants of a school building or hospital or embassy. . . . **This new way of war is data dependent. So we need to think in terms of trust and securing trust.**"

Michael Wynne, Former SECAF

"No operator should ever have to ask . . . 'Will my weapon work?' . . . Cyberspace warfare creates just this possibility."

General John A. Shaud, USAF

Why Trust and Assurance Matter

"That is, the pilot can trust information that a target is the foe, not innocent inhabitants of a school building or hospital or embassy. . . . **This new way of war is data dependent. So we need to think in terms of trust and securing trust.**"

Michael Wynne, Former SECAF

"No operator should ever have to ask . . . 'Will my weapon work?' . . . Cyberspace warfare creates just this possibility."

General John A. Shaud, USAF

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Introduction

Intended audience

- Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

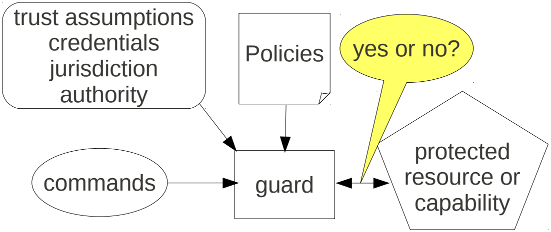
- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Our Viewpoint

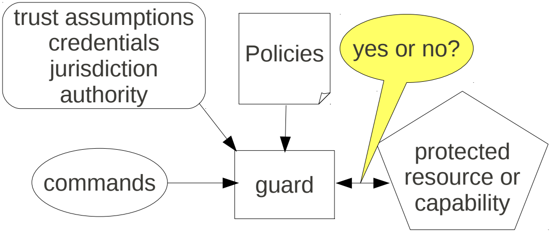


When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers

Our Viewpoint

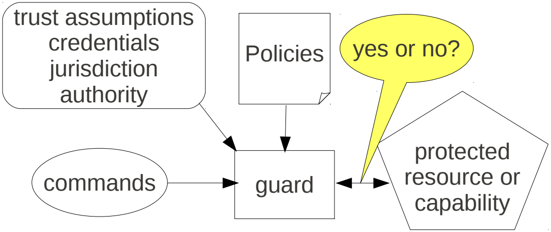


When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers

Our Viewpoint

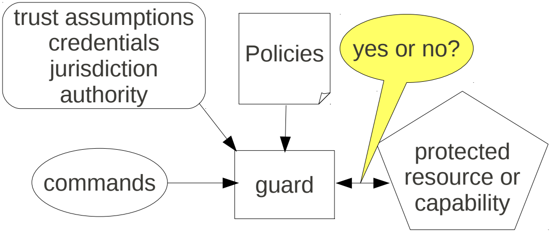


When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers

Our Viewpoint



When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers

A Logical Approach to Access Control

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

A Logical Approach to Access Control

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

A Logical Approach to Access Control

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

A Logical Approach to Access Control

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

A Logical Approach to Access Control

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

A Logical Approach to Access Control

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

Our focus: Concept of Operations (CONOPS)

CONOPS definition

> *"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."*

JP 5-0, Joint Operation Planning

Why focus on CONOPS?

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

Our focus: Concept of Operations (CONOPS)

CONOPS definition

> *"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of . . . components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission . . ."*
>
> JP 5-0, Joint Operation Planning

Why focus on CONOPS?

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

Our focus: Concept of Operations (CONOPS)

CONOPS definition

> *"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of . . . components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission . . ."*

JP 5-0, Joint Operation Planning

Why focus on CONOPS?

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

Our focus: Concept of Operations (CONOPS)

CONOPS definition

> *"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."*

JP 5-0, Joint Operation Planning

Why focus on CONOPS?

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

Our focus: Concept of Operations (CONOPS)

CONOPS definition

> *"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of . . . components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission . . ."*

JP 5-0, Joint Operation Planning

Why focus on CONOPS?

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{aligned}
P &::= A \;/\; P \,\&\, Q \;/\; P \,|\, Q \\
\varphi &::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
&\qquad P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{aligned}
$$

Kripke structures

$$
\begin{aligned}
W &= \text{non-empty } \{\text{worlds}\} \\
I &= \textbf{PropVar} \to \mathcal{P}(W) \\
J &= \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} &= \langle W, I, J \rangle
\end{aligned}
$$

Semantics

$$
\begin{aligned}
\mathcal{E}_\mathcal{M}[\![p]\!] &= I(p) \\
\mathcal{E}_\mathcal{M}[\![\neg\varphi]\!] &= W - \mathcal{E}_\mathcal{M}[\![\varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_\mathcal{M}[\![\varphi_1]\!]) \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_\mathcal{M}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_\mathcal{M}[\![P \text{ says } \varphi]\!] &= \{w \,|\, J(P)(w) \subseteq \mathcal{E}_\mathcal{M}[\![\varphi]\!]\} \\
\mathcal{E}_\mathcal{M}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{aligned}
P &::= A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi &::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
&\quad\;\; P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{aligned}
$$

Kripke structures

$$
\begin{aligned}
W &= \text{non-empty } \{\text{worlds}\} \\
I &= \textbf{PropVar} \to \mathcal{P}(W) \\
J &= \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} &= \langle W, I, J \rangle
\end{aligned}
$$

Semantics

$$
\begin{aligned}
\mathcal{E}_{\mathcal{M}}[\![p]\!] &= I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] &= W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] &= \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lll}
P & ::= & A \,/\, P \,\&\, Q \,/\, P \mid Q \\
\varphi & ::= & p \,/\, \neg\varphi \,/\, \varphi_1 \wedge \varphi_2 \,/\, \varphi_1 \vee \varphi_2 \,/\, \varphi_1 \supset \varphi_2 \,/\, \varphi_1 \equiv \varphi_2 \,/ \\
 & & P \Rightarrow Q \,/\, P \text{ says } \varphi \,/\, P \text{ controls } \varphi \,/\, P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lll}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \ / \ P \,\&\, Q \ / \ P \,|\, Q \\
\varphi & ::= & p \ / \ \neg\varphi \ / \ \varphi_1 \wedge \varphi_2 \ / \ \varphi_1 \vee \varphi_2 \ / \ \varphi_1 \supset \varphi_2 \ / \ \varphi_1 \equiv \varphi_2 \ / \\
 & & P \Rightarrow Q \ / \ P \text{ says } \varphi \ / \ P \text{ controls } \varphi \ / \ P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \,|\, J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$P ::= A \;/\; P \,\&\, Q \;/\; P \,|\, Q$$

$$\varphi ::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/\; P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi$$

Kripke structures

$$\begin{aligned} W &= \text{non-empty } \{\text{worlds}\} \\ I &= \textbf{PropVar} \to \mathcal{P}(W) \\ J &= \textbf{PName} \to \mathcal{P}(W \times W) \\ \mathcal{M} &= \langle W, I, J \rangle \end{aligned}$$

Semantics

$$\begin{aligned}
\mathcal{E}_\mathcal{M}[\![p]\!] &= I(p) \\
\mathcal{E}_\mathcal{M}[\![\neg\varphi]\!] &= W - \mathcal{E}_\mathcal{M}[\![\varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_\mathcal{M}[\![\varphi_1]\!]) \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_\mathcal{M}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_\mathcal{M}[\![P \text{ says } \varphi]\!] &= \{w \,|\, J(P)(w) \subseteq \mathcal{E}_\mathcal{M}[\![\varphi]\!]\} \\
\mathcal{E}_\mathcal{M}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$P ::= A \;/\; P \,\&\, Q \;/\; P \,|\, Q$$

$$\varphi ::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/\; P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi$$

Kripke structures

$$\begin{aligned} W &= \text{non-empty } \{\text{worlds}\} \\ I &= \textbf{PropVar} \to \mathcal{P}(W) \\ J &= \textbf{PName} \to \mathcal{P}(W \times W) \\ \mathcal{M} &= \langle W, I, J \rangle \end{aligned}$$

Semantics

$$\begin{aligned}
\mathcal{E}_\mathcal{M}[\![p]\!] &= I(p) \\
\mathcal{E}_\mathcal{M}[\![\neg\varphi]\!] &= W - \mathcal{E}_\mathcal{M}[\![\varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_\mathcal{M}[\![\varphi_1]\!]) \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_\mathcal{M}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_\mathcal{M}[\![P \text{ says } \varphi]\!] &= \{w \,|\, J(P)(w) \subseteq \mathcal{E}_\mathcal{M}[\![\varphi]\!]\} \\
\mathcal{E}_\mathcal{M}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lll}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lll}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_\mathcal{M}[\![p]\!] & = & I(p) \\
\mathcal{E}_\mathcal{M}[\![\neg\varphi]\!] & = & W - \mathcal{E}_\mathcal{M}[\![\varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_\mathcal{M}[\![\varphi_1]\!]) \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_\mathcal{M}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_\mathcal{M}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_\mathcal{M}[\![\varphi]\!]\} \\
\mathcal{E}_\mathcal{M}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_\mathcal{M}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_\mathcal{M}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$\begin{array}{lcl} P & ::= & A \;/\; P \,\&\, Q \;/\; P \mid Q \\ \varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\ & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi \end{array}$$

Kripke structures

$$\begin{array}{lcl} W & = & \text{non-empty } \{\text{worlds}\} \\ I & = & \textbf{PropVar} \to \mathcal{P}(W) \\ J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\ \mathcal{M} & = & \langle W, I, J \rangle \end{array}$$

Semantics

$$\begin{array}{rcl} \mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\ \mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\ \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\ \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\ \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\ \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\ \mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\ \mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\ \mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\ \mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!] \end{array}$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{rcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \,|\, Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{rcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \,|\, J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \,|\, Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \,|\, J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \,|\, Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \rightarrow \mathcal{P}(W) \\
J & = & \textbf{PName} \rightarrow \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \,|\, J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$P ::= A \;/\; P \,\&\, Q \;/\; P \mid Q$$

$$\varphi ::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/\; P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi$$

Kripke structures

$$\begin{aligned} W &= \text{non-empty } \{\text{worlds}\} \\ I &= \textbf{PropVar} \to \mathcal{P}(W) \\ J &= \textbf{PName} \to \mathcal{P}(W \times W) \\ \mathcal{M} &= \langle W, I, J \rangle \end{aligned}$$

Semantics

$$\begin{aligned}
\mathcal{E}_{\mathcal{M}}[\![p]\!] &= I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] &= W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] &= \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{aligned}
P &::= A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi &::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
&\quad\; P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{aligned}
$$

Kripke structures

$$
\begin{aligned}
W &= \text{non-empty } \{\text{worlds}\} \\
I &= \textbf{PropVar} \to \mathcal{P}(W) \\
J &= \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} &= \langle W, I, J \rangle
\end{aligned}
$$

Semantics

$$
\begin{aligned}
\mathcal{E}_{\mathcal{M}}[\![p]\!] &= I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] &= W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] &= \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{aligned}
P &::= A \;/\; P \,\&\, Q \;/\; P \,|\, Q \\
\varphi &::= p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
& \quad\;\; P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{aligned}
$$

Kripke structures

$$
\begin{aligned}
W &= \text{non-empty } \{\text{worlds}\} \\
I &= \textbf{PropVar} \to \mathcal{P}(W) \\
J &= \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} &= \langle W, I, J \rangle
\end{aligned}
$$

Semantics

$$
\begin{aligned}
\mathcal{E}_\mathcal{M}[\![p]\!] &= I(p) \\
\mathcal{E}_\mathcal{M}[\![\neg\varphi]\!] &= W - \mathcal{E}_\mathcal{M}[\![\varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_\mathcal{M}[\![\varphi_1]\!]) \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_\mathcal{M}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_\mathcal{M}[\![P \text{ says } \varphi]\!] &= \{w \,|\, J(P)(w) \subseteq \mathcal{E}_\mathcal{M}[\![\varphi]\!]\} \\
\mathcal{E}_\mathcal{M}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_\mathcal{M}[\![P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_\mathcal{M}[\![p]\!] & = & I(p) \\
\mathcal{E}_\mathcal{M}[\![\neg\varphi]\!] & = & W - \mathcal{E}_\mathcal{M}[\![\varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_\mathcal{M}[\![\varphi_1]\!] \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_\mathcal{M}[\![\varphi_1]\!]) \cup \mathcal{E}_\mathcal{M}[\![\varphi_2]\!] \\
\mathcal{E}_\mathcal{M}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_\mathcal{M}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_\mathcal{M}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_\mathcal{M}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_\mathcal{M}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_\mathcal{M}[\![\varphi]\!]\} \\
\mathcal{E}_\mathcal{M}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_\mathcal{M}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_\mathcal{M}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_\mathcal{M}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Access-Control Logic Syntax & Semantics

Syntax

- Principals (actors)
- Statements they make

BNF

$$
\begin{array}{lcl}
P & ::= & A \;/\; P \,\&\, Q \;/\; P \mid Q \\
\varphi & ::= & p \;/\; \neg\varphi \;/\; \varphi_1 \wedge \varphi_2 \;/\; \varphi_1 \vee \varphi_2 \;/\; \varphi_1 \supset \varphi_2 \;/\; \varphi_1 \equiv \varphi_2 \;/ \\
 & & P \Rightarrow Q \;/\; P \text{ says } \varphi \;/\; P \text{ controls } \varphi \;/\; P \text{ reps } Q \text{ on } \varphi
\end{array}
$$

Kripke structures

$$
\begin{array}{lcl}
W & = & \text{non-empty } \{\text{worlds}\} \\
I & = & \textbf{PropVar} \to \mathcal{P}(W) \\
J & = & \textbf{PName} \to \mathcal{P}(W \times W) \\
\mathcal{M} & = & \langle W, I, J \rangle
\end{array}
$$

Semantics

$$
\begin{array}{rcl}
\mathcal{E}_{\mathcal{M}}[\![p]\!] & = & I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] & = & W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] & = & (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] & = & \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] & = & \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] & = & \{w \mid J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] & = & \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{array}
$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_\mathcal{M}[\![H_i]\!] = W$

then $\mathcal{E}_\mathcal{M}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$
then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$

then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\quad \dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\quad \dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ $\qquad$ *Says* $\quad \dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\quad \dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\quad \dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\quad \dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\quad \dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow \quad \dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid \quad \dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid \quad \dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \overset{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \overset{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$
then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\quad \dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\quad \dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ $\qquad$ *Says* $\quad \dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\quad \dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\quad \dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\quad \dfrac{}{P \,|\, Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\quad \dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\quad \dfrac{}{P \Rightarrow P}$

Monotonicity of $|$ $\quad \dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \,|\, Q' \Rightarrow P \,|\, Q}$

Associativity of $|$ $\quad \dfrac{P \,|\, (Q \,|\, R) \text{ says } \varphi}{(P \,|\, Q) \,|\, R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \,|\, Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$

then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

$$\text{If all } \mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$$
$$\text{then } \mathcal{E}_{\mathcal{M}}[\![C]\!] = W$$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\quad \dfrac{}{\varphi}\quad$ if φ is an instance of a prop-logic tautology

Modus Ponens $\quad \dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ $\qquad$ *Says* $\quad \dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\quad \dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\quad \dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\quad \dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\quad \dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\quad \dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\quad \dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\quad \dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \;\stackrel{\text{def}}{=}\; (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \;\stackrel{\text{def}}{=}\; P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$

then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

$$\text{If all } \mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$$
$$\text{then } \mathcal{E}_{\mathcal{M}}[\![C]\!] = W$$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$

then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures $\mathcal{M}$* and each $i \in \{1, \ldots, n\}$:

If all $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$
then $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\dfrac{}{\varphi}$ if φ is an instance of a prop-logic tautology

Modus Ponens $\dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ *Says* $\dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

Inference Rules

Rules

- Inconvenient to use Kripke semantics
- Use inference rules $\dfrac{H_1 \cdots H_n}{C}$ instead

Soundness

$\dfrac{H_1 \cdots H_n}{C}$ is sound if *for all Kripke structures* $\mathcal{M}$ and each $i \in \{1, \ldots, n\}$:

$$\text{If all } \mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W \text{ then } \mathcal{E}_{\mathcal{M}}[\![C]\!] = W$$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Core inference rules

Taut $\quad \dfrac{}{\varphi}$ $\quad$ if φ is an instance of a prop-logic tautology

Modus Ponens $\quad \dfrac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$ $\qquad$ *Says* $\quad \dfrac{\varphi}{P \text{ says } \varphi}$

MP Says $\quad \dfrac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$

Speaks For $\quad \dfrac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$

Quoting $\quad \dfrac{}{P \mid Q \text{ says } \varphi \equiv P \text{ says } Q \text{ says } \varphi}$

& Says $\quad \dfrac{}{P \,\&\, Q \text{ says } \varphi \equiv P \text{ says } \varphi \wedge Q \text{ says } \varphi}$

Idempotency of $\Rightarrow$ $\quad \dfrac{}{P \Rightarrow P}$

Monotonicity of $\mid$ $\quad \dfrac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$

Associativity of $\mid$ $\quad \dfrac{P \mid (Q \mid R) \text{ says } \varphi}{(P \mid Q) \mid R \text{ says } \varphi}$

$$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$$

$$P \text{ reps } Q \text{ on } \varphi \stackrel{\text{def}}{=} P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi$$

A Simple Proof

1.	P controls φ	Assumption
2.	P says φ	Assumption
3.	P says $\varphi \supset \varphi$	1 def'n controls
4.	φ	2, 3 Modus Ponens

Derived inference rule

$$\textit{Controls} \quad \frac{P \text{ controls } \varphi \qquad P \text{ says } \varphi}{\varphi}$$

All derived rules are sound

A Simple Proof

1.	P controls φ	Assumption
2.	P says φ	Assumption
3.	P says $\varphi \supset \varphi$	1 def'n controls
4.	φ	2, 3 Modus Ponens

Derived inference rule

$$\textit{Controls} \quad \frac{P \text{ controls } \varphi \qquad P \text{ says } \varphi}{\varphi}$$

All derived rules are sound

A Simple Proof

1.	P controls φ	Assumption
2.	P says φ	Assumption
3.	P says $\varphi \supset \varphi$	1 def'n controls
4.	φ	2, 3 Modus Ponens

Derived inference rule

$$\textit{Controls} \quad \frac{P \text{ controls } \varphi \qquad P \text{ says } \varphi}{\varphi}$$

All derived rules are sound

A Simple Proof

1.	P controls φ	Assumption
2.	P says φ	Assumption
3.	P says $\varphi \supset \varphi$	1 def'n controls
4.	φ	2, 3 Modus Ponens

Derived inference rule

$$\textit{Controls} \quad \frac{P \text{ controls } \varphi \qquad P \text{ says } \varphi}{\varphi}$$

All derived rules are sound

A Simple Proof

1.	P controls φ	Assumption
2.	P says φ	Assumption
3.	P says $\varphi \supset \varphi$	1 def'n controls
4.	φ	2, 3 Modus Ponens

Derived inference rule

$$\textit{Controls} \quad \frac{P \text{ controls } \varphi \qquad P \text{ says } \varphi}{\varphi}$$

All derived rules are sound

A Simple Proof

1.	P controls φ	Assumption
2.	P says φ	Assumption
3.	P says $\varphi \supset \varphi$	1 def'n controls
4.	φ	2, 3 Modus Ponens

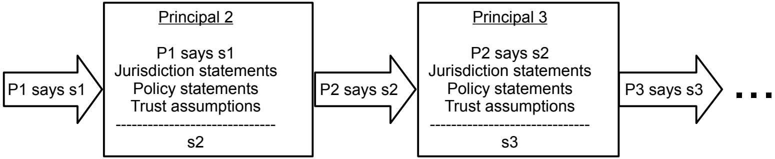
Derived inference rule

$$\textit{Controls} \quad \frac{P \text{ controls } \varphi \qquad P \text{ says } \varphi}{\varphi}$$

All derived rules are sound

General Form of CONOPS

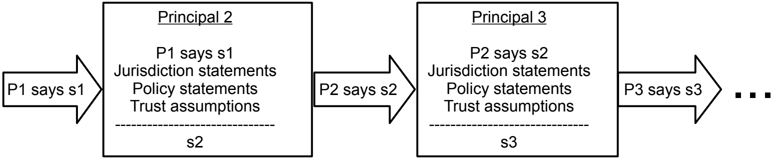
"The CONOPS ... describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule

General Form of CONOPS

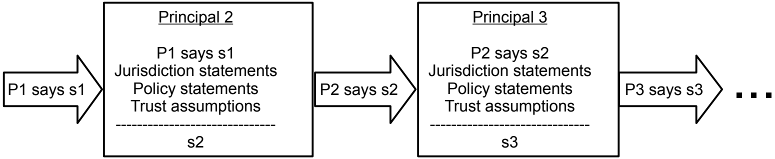
"The CONOPS ... describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule

General Form of CONOPS

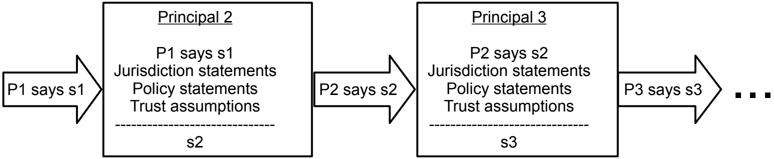
"The CONOPS ... describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule

General Form of CONOPS

"The CONOPS ... describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule

Example: A (Hypothetical) Kill Chain

Joint Terminal Air Controller

$\Longleftrightarrow$

Airborne Early Warning & Control

$\Updownarrow$ $\swarrow\nearrow$ $\Updownarrow$

Remotely Piloted Vehicle

$\Longleftrightarrow$

Air Operations Center

Example: A (Hypothetical) Kill Chain

$\Longleftrightarrow$

Joint Terminal Air Controller

Airborne Early Warning & Control

$\Updownarrow$ $\swarrow\nearrow$ $\Updownarrow$

$\Longleftrightarrow$

Remotely Piloted Vehicle

Air Operations Center

Example: A (Hypothetical) Kill Chain

Joint Terminal Air Controller

$\Longleftrightarrow$

Airborne Early Warning & Control

$\Updownarrow$ $\swarrow\nearrow$ $\Updownarrow$

Remotely Piloted Vehicle

$\Longleftrightarrow$

Air Operations Center

Example: A (Hypothetical) Kill Chain

Joint Terminal Air Controller

$\Longleftrightarrow$

Airborne Early Warning & Control

$\Updownarrow$ $\swarrow\nearrow$ $\Updownarrow$

Remotely Piloted Vehicle

$\Longleftrightarrow$

Air Operations Center

Example: A (Hypothetical) Kill Chain

Joint Terminal Air Controller

$\Longleftrightarrow$

Airborne Early Warning & Control

$\Updownarrow$ $\swarrow\nearrow$ $\Updownarrow$

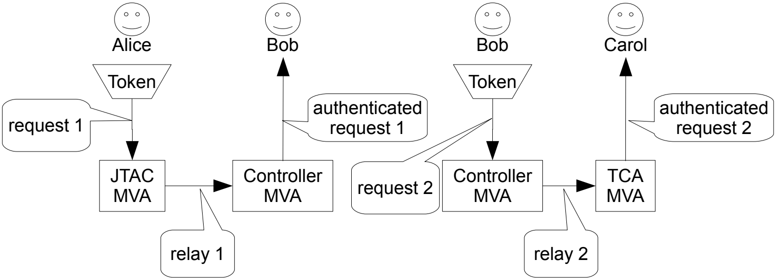
Remotely Piloted Vehicle

$\Longleftrightarrow$

Air Operations Center

Hypothetical CONOPS

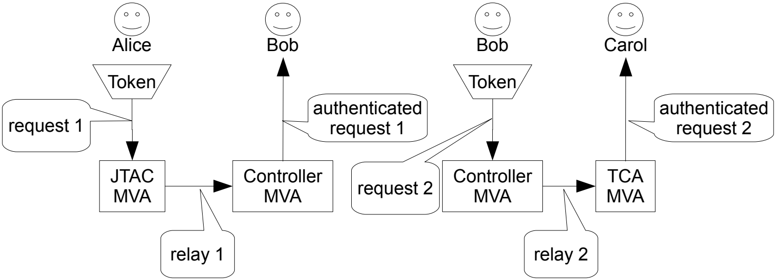
All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(Token_{Alice} \mid JTAC)$ says $\langle strike, target \rangle$
relay 1	$(K_{JTAC\text{-}MVA} \mid JTAC)$ says $\langle strike, target \rangle$
authenticated request 1	$JTAC$ says $\langle strike, target \rangle$
request 2	$(Token_{Bob} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
relay 2	$(K_{Controller\text{-}MVA} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
authenticated request 2	$Controller$ says $(JTAC$ says $\langle strike, target \rangle)$

Hypothetical CONOPS

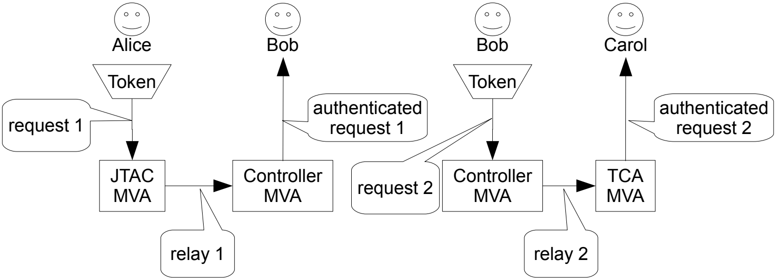
All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(Token_{Alice} \mid JTAC)$ says $\langle strike, target \rangle$
relay 1	$(K_{JTAC\text{-}MVA} \mid JTAC)$ says $\langle strike, target \rangle$
authenticated request 1	$JTAC$ says $\langle strike, target \rangle$
request 2	$(Token_{Bob} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
relay 2	$(K_{Controller\text{-}MVA} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
authenticated request 2	$Controller$ says $(JTAC$ says $\langle strike, target \rangle)$

Hypothetical CONOPS

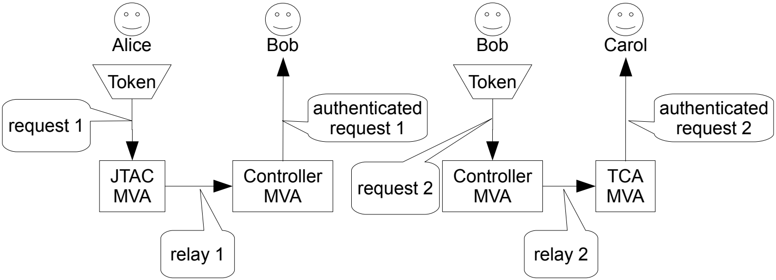
All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(\mathit{Token}_{\mathit{Alice}} \mid \mathit{JTAC})$ says $\langle \mathit{strike}, \mathit{target} \rangle$
relay 1	$(K_{\mathit{JTAC\text{-}MVA}} \mid \mathit{JTAC})$ says $\langle \mathit{strike}, \mathit{target} \rangle$
authenticated request 1	JTAC says $\langle \mathit{strike}, \mathit{target} \rangle$
request 2	$(\mathit{Token}_{\mathit{Bob}} \mid \mathit{Controller})$ says $(\mathit{JTAC}$ says $\langle \mathit{strike}, \mathit{target} \rangle)$
relay 2	$(K_{\mathit{Controller\text{-}MVA}} \mid \mathit{Controller})$ says $(\mathit{JTAC}$ says $\langle \mathit{strike}, \mathit{target} \rangle)$
authenticated request 2	$\mathit{Controller}$ says $(\mathit{JTAC}$ says $\langle \mathit{strike}, \mathit{target} \rangle)$

Hypothetical CONOPS

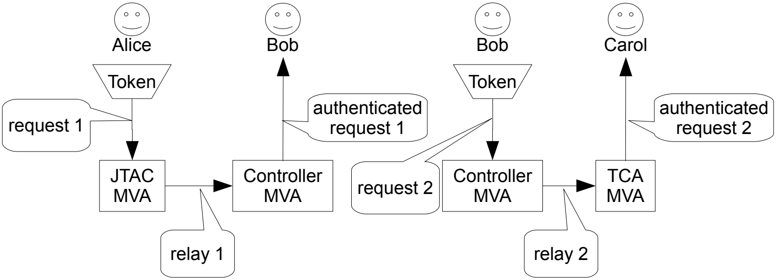
All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(Token_{Alice} \mid JTAC)$ says $\langle strike, target\rangle$
relay 1	$(K_{JTAC\text{-}MVA} \mid JTAC)$ says $\langle strike, target\rangle$
authenticated request 1	$JTAC$ says $\langle strike, target\rangle$
request 2	$(Token_{Bob} \mid Controller)$ says $(JTAC$ says $\langle strike, target\rangle)$
relay 2	$(K_{Controller\text{-}MVA} \mid Controller)$ says $(JTAC$ says $\langle strike, target\rangle)$
authenticated request 2	$Controller$ says $(JTAC$ says $\langle strike, target\rangle)$

Hypothetical CONOPS

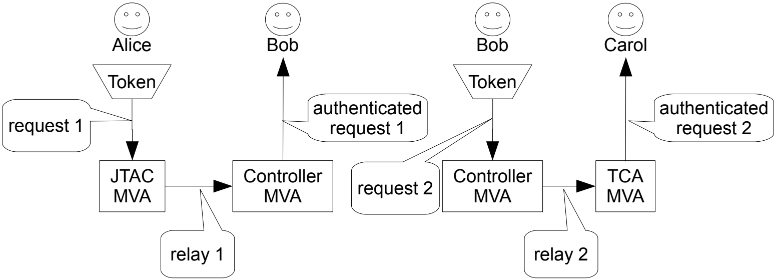
All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(Token_{Alice} \mid JTAC)$ says $\langle strike, target \rangle$
relay 1	$(K_{JTAC\text{-}MVA} \mid JTAC)$ says $\langle strike, target \rangle$
authenticated request 1	$JTAC$ says $\langle strike, target \rangle$
request 2	$(Token_{Bob} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
relay 2	$(K_{Controller\text{-}MVA} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
authenticated request 2	$Controller$ says $(JTAC$ says $\langle strike, target \rangle)$

Hypothetical CONOPS

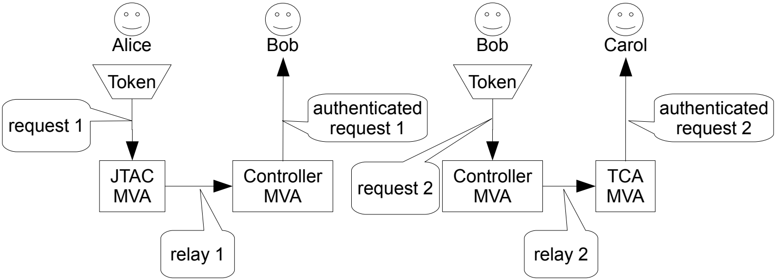
All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(Token_{Alice} \mid JTAC)$ says $\langle strike, target\rangle$
relay 1	$(K_{JTAC\text{-}MVA} \mid JTAC)$ says $\langle strike, target\rangle$
authenticated request 1	$JTAC$ says $\langle strike, target\rangle$
request 2	$(Token_{Bob} \mid Controller)$ says $(JTAC$ says $\langle strike, target\rangle)$
relay 2	$(K_{Controller\text{-}MVA} \mid Controller)$ says $(JTAC$ says $\langle strike, target\rangle)$
authenticated request 2	$Controller$ says $(JTAC$ says $\langle strike, target\rangle)$

Hypothetical CONOPS

All personnel, roles, requests, and commands authenticated by a *Mission Validation Appliance (MVA)*.



Statement	Formal Representation
request 1	$(Token_{Alice} \mid JTAC)$ says $\langle strike, target \rangle$
relay 1	$(K_{JTAC\text{-}MVA} \mid JTAC)$ says $\langle strike, target \rangle$
authenticated request 1	$JTAC$ says $\langle strike, target \rangle$
request 2	$(Token_{Bob} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
relay 2	$(K_{Controller\text{-}MVA} \mid Controller)$ says $(JTAC$ says $\langle strike, target \rangle)$
authenticated request 2	$Controller$ says $(JTAC$ says $\langle strike, target \rangle)$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(Token\ or\ Key \mid Role)$ says φ
Delegation Cert	K_{Auth} says $(Person\ or\ Object$ reps $Role$ on $\varphi)$
Key Certificate	K_{Auth} says $(Token\ or\ Key \Rightarrow Person\ or\ Object)$
Jurisdiction	$Auth$ controls $(Person\ or\ Object$ reps $Role$ on $\varphi)$
Jurisdiction	$Auth$ controls $(Token\ or\ Key \Rightarrow Person\ or\ Object)$
Trust Assumption	$K_{Auth} \Rightarrow Auth$

Transmitting MVA:

$$MVA\ 1\ \frac{\begin{array}{c}(Token \mid Role) \text{ says } \varphi \\ K_{Auth} \text{ says } (Person \text{ reps } Role \text{ on } \varphi) \\ K_{Auth} \text{ says } (Token \Rightarrow Person) \\ Auth \text{ controls } (Person \text{ reps } Role \text{ on } \varphi) \\ Auth \text{ controls } (Token \Rightarrow Person) \\ K_{Auth} \Rightarrow Auth\end{array}}{K_{MVA_1} \mid Role \text{ says } \varphi}$$

Receiving MVA
:

$$MVA\ 2\ \frac{\begin{array}{c}(K_{MVA_1} \mid Role) \text{ says } \varphi \\ K_{Auth} \text{ says } (MVA_1 \text{ reps } Role \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow MVA_1) \\ Auth \text{ controls } (MVA_1 \text{ reps } Role \text{ on } \varphi) \\ Auth \text{ controls } (K_{MVA_1} \Rightarrow MVA_1) \\ K_{Auth} \Rightarrow Auth\end{array}}{Role \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\textit{MVA 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\textit{MVA 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\textit{MVA 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\textit{MVA 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA
:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role})\ \text{says}\ \varphi \\ K_{Auth}\ \text{says}\ (\mathit{Person}\ \text{reps}\ \mathit{Role}\ \text{on}\ \varphi) \\ K_{Auth}\ \text{says}\ (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth}\ \text{controls}\ (\mathit{Person}\ \text{reps}\ \mathit{Role}\ \text{on}\ \varphi) \\ \mathit{Auth}\ \text{controls}\ (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role}\ \text{says}\ \varphi}$$

Receiving MVA

:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role})\ \text{says}\ \varphi \\ K_{Auth}\ \text{says}\ (\mathit{MVA}_1\ \text{reps}\ \mathit{Role}\ \text{on}\ \varphi) \\ K_{Auth}\ \text{says}\ (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth}\ \text{controls}\ (\mathit{MVA}_1\ \text{reps}\ \mathit{Role}\ \text{on}\ \varphi) \\ \mathit{Auth}\ \text{controls}\ (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role}\ \text{says}\ \varphi}$$

Repeated Pattern of Authentication and Trust

Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input	$(\mathit{Token\ or\ Key} \mid \mathit{Role})$ says φ
Delegation Cert	K_{Auth} says $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Key Certificate	K_{Auth} says $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Jurisdiction	Auth controls $(\mathit{Person\ or\ Object}$ reps Role on $\varphi)$
Jurisdiction	Auth controls $(\mathit{Token\ or\ Key} \Rightarrow \mathit{Person\ or\ Object})$
Trust Assumption	$K_{Auth} \Rightarrow \mathit{Auth}$

Transmitting MVA:

$$\mathit{MVA\ 1}\ \frac{\begin{array}{c}(\mathit{Token} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (\mathit{Token} \Rightarrow \mathit{Person}) \\ \mathit{Auth} \text{ controls } (\mathit{Person} \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (\mathit{Token} \Rightarrow \mathit{Person}) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{K_{MVA_1} \mid \mathit{Role} \text{ says } \varphi}$$

Receiving MVA

:

$$\mathit{MVA\ 2}\ \frac{\begin{array}{c}(K_{MVA_1} \mid \mathit{Role}) \text{ says } \varphi \\ K_{Auth} \text{ says } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ K_{Auth} \text{ says } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ \mathit{Auth} \text{ controls } (\mathit{MVA}_1 \text{ reps } \mathit{Role} \text{ on } \varphi) \\ \mathit{Auth} \text{ controls } (K_{MVA_1} \Rightarrow \mathit{MVA}_1) \\ K_{Auth} \Rightarrow \mathit{Auth}\end{array}}{\mathit{Role} \text{ says } \varphi}$$

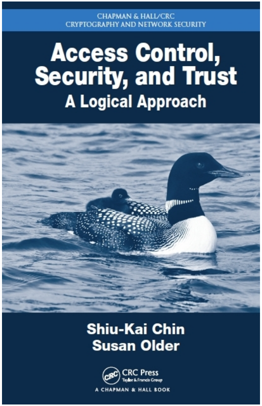
Findings & Conclusions

226+ ACE cadets, captains, & lieutenants from 40+ universities

Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on access-control logic taught in ACE



Increased their capabilities to design, specify, evaluate, and procure critical systems

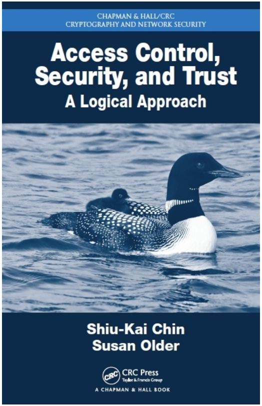
Findings & Conclusions

226+ ACE cadets, captains, & lieutenants from 40+ universities

Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on access-control logic taught in ACE



Increased their capabilities to design, specify, evaluate, and procure critical systems

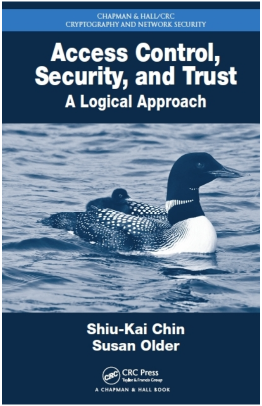
Findings & Conclusions

226+ ACE cadets, captains, & lieutenants from 40+ universities

Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on access-control logic taught in ACE



Increased their capabilities to design, specify, evaluate, and procure critical systems

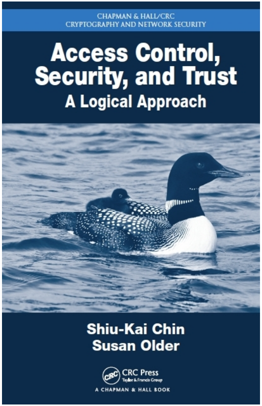
Findings & Conclusions

226+ ACE cadets, captains, & lieutenants from 40+ universities

Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on access-control logic taught in ACE



Increased their capabilities to design, specify, evaluate, and procure critical systems

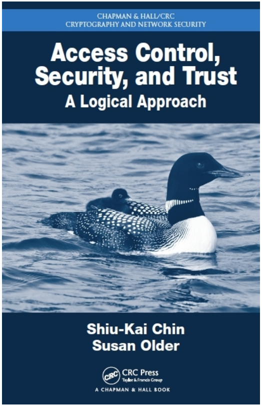
Findings & Conclusions

226+ ACE cadets, captains, & lieutenants from 40+ universities

Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on access-control logic taught in ACE



Increased their capabilities to design, specify, evaluate, and procure critical systems